

Massimiliano Sala  
Teo Mora  
Ludovic Perret  
Shojiro Sakata  
Carlo Traverso  
*Editors*

---

# Gröbner Bases, Coding, and Cryptography



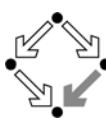
# Gröbner Bases, Coding, and Cryptography

Massimiliano Sala · Teo Mora · Ludovic Perret ·  
Shojiro Sakata · Carlo Traverso  
Editors

# Gröbner Bases, Coding, and Cryptography



Springer



RISC

Massimiliano Sala  
University College Cork  
Boole Centre for Research in Informatics  
Crosses Green  
17 South Bank  
Cork, Ireland  
[msala@bcri.ucc.ie](mailto:msala@bcri.ucc.ie)

Teo Mora  
Università Genova  
Dipto. Matematica  
Via Dodecaneso, 35  
16146 Genova, Italy  
[theomora@disi.unige.it](mailto:theomora@disi.unige.it)

Ludovic Perret  
University of Paris VI, LIP6  
104 avenue du Président Kennedy  
75016 Paris, France  
[ludovic.perret@lip6.fr](mailto:ludovic.perret@lip6.fr)

Shojiro Sakata  
Toyohashi University of Technology  
Faculty of Engineering  
Dept. Production Systems Engineering  
1-1 Hibarigaoka  
Toyohashi, Aichi  
Tempaku-cho 441-8580, Japan  
[sakata@ice.uec.ac.jp](mailto:sakata@ice.uec.ac.jp)

Carlo Traverso  
Università Pisa  
Dipto. Matematica  
Largo Bruno Pontecorvo, 5  
56127 Pisa, Italy  
[traverso@dm.unipi.it](mailto:traverso@dm.unipi.it)

ISBN 978-3-540-93805-7

e-ISBN 978-3-540-93806-4

DOI 10.1007/978-3-540-93806-4

Library of Congress Control Number: 2008944307

© 2009 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

[springer.com](http://springer.com)

# Preface

**Massimiliano Sala**

In the period February–July 2006 a major research event took place in Linz (Austria): the Special Semester in Gröbner Bases and Related Areas.<sup>1</sup> Organized by RICAM (in close cooperation with RISC) and funded by the Austrian Academy of Sciences, it saw the involvement of hundreds of people working in Gröbner bases theory and their applications. In particular, a workshop (D1<sup>2</sup>) was held, co-chaired by Mikhail Klin (algebraic combinatorics), Ludovic Perret (cryptography) and me (coding theory). The aim of the workshop was twofold: to present possible applications of the theory to experts in Gröbner bases (so that they could explore new research fields) and to present Gröbner bases as an attractive tool to people working in other areas. Therefore, the invited talks were mainly tutorials and surveys, while posters and contributed talks outlined specific research results.

Workshop D1 was a success, with a large audience coming from different backgrounds. It was suggested that some<sup>3</sup> of the best D1 presentations related to cryptography and codes would be collected in a book of the RISC Book Series. The invited talks would become book *chapters*. The posters and contributed talks would become short *notes* at the end of the book. I was appointed Managing Editor, with an Editorial Board composed of Teo Mora (Gröbner bases related papers), Ludovic Perret (cryptography), Shojiro Sakata (AG codes) and Carlo Traverso (Gröbner bases and coding). To cover some interesting aspects not presented at Workshop D1, we invited a few more papers and notes.

I would like to thank all of them for their great help and assistance in planning, shaping and editing this book. The Board and I would like to express our gratefulness for their supervision to Bruno Buchberger and the series editor Peter Paule.

---

<sup>1</sup><http://www.ricam.oeaw.ac.at/specsem/srs/groeb/index.htm>.

<sup>2</sup>“Gröbner Bases in Cryptography, Coding Theory, and Algebraic Combinatorics”.

<sup>3</sup>Other D1 presentations will appear in a special issue of Journal of Symbolic Computation, edited by D. Augot, J.-C. Faugère and L. Perret.

# Contents

## Gröbner Bases, Coding, and Cryptography: a Guide

<b>to the State-of-Art</b> . . . . .	1
Massimiliano Sala	
1 In the Beginning . . . . .	1
2 Until Now . . . . .	2
2.1 Classical Coding Theory . . . . .	3
2.2 AG Codes . . . . .	4
2.3 Coding Miscellanea . . . . .	4
2.4 Cryptography . . . . .	5
3 Final Comments . . . . .	6
References . . . . .	6

## Part 1 Invited Papers

<b>Gröbner Technology</b> . . . . .	11
Teo Mora	
1 Notation and Definitions . . . . .	11
2 Term-Orderings: Classification and Representation . . . . .	16
3 Buchberger's Theorem and Algorithm . . . . .	19
References . . . . .	24

## The FGLM Problem and Möller's Algorithm on Zero-dimensional

<b>Ideals</b> . . . . .	27
Teo Mora	
1 Duality . . . . .	27
2 Möller's Algorithm . . . . .	28
3 The FGLM Problem . . . . .	33
4 The FGLM Matrix . . . . .	33
5 Pointers . . . . .	35
6 Point Evaluation . . . . .	38
6.1 Möller's Algorithm . . . . .	38
6.2 Cerlienco–Mureddu Correspondence . . . . .	38
6.3 Farr–Gao Analysis . . . . .	39
6.4 Points with Multiplicities . . . . .	42
References . . . . .	43

<b>An Introduction to Linear and Cyclic Codes</b>	47
Daniel Augot, Emanuele Betti and Emmanuela Orsini	
1 An Overview on Error Correcting Codes	47
2 Linear Codes	48
2.1 Basic Definitions	48
2.2 Hamming Distance	49
2.3 Decoding Linear Codes	52
3 Some Bounds on Codes	54
4 Cyclic Codes	55
4.1 An Algebraic Correspondence	55
4.2 Encoding and Decoding with Cyclic Codes	56
4.3 Zeros of Cyclic Codes	57
5 Some Examples of Cyclic Codes	58
5.1 Hamming and Simplex Codes	58
5.2 Quadratic Residue Codes	60
6 BCH Codes	60
6.1 On the Optimality of BCH Codes	61
7 Decoding BCH Codes	62
8 On the Asymptotic Properties of Cyclic Codes	66
References	67
<b>Decoding Cyclic Codes: the Cooper Philosophy</b>	69
Teo Mora and Emmanuela Orsini	
1 Introduction	69
2 Decoding Binary BCH Codes	71
3 Gröbner Bases for Cyclic Codes	74
3.1 Decoding Binary Cyclic Codes	74
3.2 Decoding Cyclic Codes over $\mathbb{F}_q$	75
3.3 A New System with the Newton Identities	76
4 The CRHT Syndrome Variety	77
5 The Gianni–Kalkbrenner Shape Theorem	78
6 The General Error Locator Polynomial	85
7 A Newton-Based Decoder	88
References	90
<b>A Tutorial on AG Code Construction from a Gröbner Basis Perspective</b>	93
Douglas A. Leonard	
1 Introduction	93
2 Traditional AG Approach	95
3 Weighted Total-Degree Orders	97
4 Hermitian Codes and Affine-Variety Codes	97
5 Curve Definition	99
References	106

<b>Automorphisms and Encoding of AG and Order Domain Codes</b>	107
John B. Little	
1 Introduction	107
2 Other Encoding Methods for AG Goppa Codes	108
3 Automorphisms and Module Structures	109
4 A Systematic Encoding Algorithm	110
5 Complexity Comparisons	112
6 Automorphisms of Curves and AG Goppa Codes	112
7 Examples	114
References	119
<b>Algebraic Geometry Codes from Order Domains</b>	121
Olav Geil	
1 Introduction	121
2 Order Domains with Weight Functions	122
3 Codes from Order Domains	125
4 One-Point Geometric Goppa Codes	132
5 Gröbner Basis Theoretical Tools for the Construction of Order Domains	133
6 Gröbner Basis Theoretical Tools for the Code Construction	137
7 The Connection to Valuation Theory	140
References	140
<b>The BMS Algorithm</b>	143
Shojiro Sakata	
1 Introduction	143
2 Generating Arrays	146
3 BMS Algorithm	148
4 Variations	154
4.1 Multiarray BMS Algorithm	157
4.2 Vectorial BMS Algorithm	158
4.3 Non-Homogeneous BMS Algorithm	160
4.4 Submodule BMS Algorithm	160
4.5 Semigroup BMS Algorithm	161
5 Conclusion	161
Appendix A: Computation of BMS Algorithm	161
Example of Computation	161
References	163
<b>The BMS Algorithm and Decoding of AG Codes</b>	165
Shojiro Sakata	
1 Introduction	166
2 Syndrome Decoding of Dual Codes	168
3 Multivariate Polynomial Interpolation and List Decoding of Primal Codes	173
4 Other Relevant Decoding Methods of Primal/Dual Codes	179

5 Conclusion . . . . .	182
References . . . . .	183
<b>A Tutorial on AG Code Decoding from a Gröbner Basis</b>	
<b>Perspective</b> . . . . .	187
Douglas A. Leonard	
1 Introduction . . . . .	187
2 Functional Decoding of RS Codes and AG Codes Using Syndromes and Error-Locator Ideals . . . . .	187
3 Interpolation to Do List Decoding for RS Codes and AG Codes . . . . .	192
References . . . . .	195
<b>FGLM-Like Decoding: from Fitzpatrick's Approach to Recent Developments</b> . . . . .	
Eleonora Guerrini and Anna Rimoldi	
1 Introduction . . . . .	197
2 Iterative Computation of Gröbner Basis . . . . .	198
3 The Key Equation for Alternant Codes . . . . .	201
4 Variations . . . . .	202
5 Some Applications to AG Codes . . . . .	203
6 Errors and Erasures for Alternant Codes . . . . .	204
6.1 Errors and Erasures . . . . .	204
6.2 Solutions Using Gröbner Bases . . . . .	205
7 List Decoding Problem . . . . .	207
7.1 Sudan's Approach . . . . .	208
7.2 Improvements on the Interpolation Steps for the RS Codes . . . . .	210
7.3 Method in Sect. 12.2 Applied to List Decoding for AG Codes . . . . .	212
7.4 Hard-Decision List Decoding and List Decoding with Soft Information . . . . .	214
8 Conclusions . . . . .	216
References . . . . .	216
<b>An Introduction to Ring-Linear Coding Theory</b> . . . . .	
Marcus Greferath	
1 Introduction and History . . . . .	219
2 Rings and Modules . . . . .	221
2.1 Some Classes of Rings . . . . .	221
3 Weight Functions on Finite Rings and Modules . . . . .	223
4 Linear and Cyclic Codes . . . . .	224
4.1 Cyclic Linear Codes . . . . .	225
5 A Foundational Result: Code Equivalence . . . . .	225
6 Weight Enumerators and MacWilliams' Identity . . . . .	227
7 Code Optimality: Bounds on the Parameters of Codes . . . . .	230
8 Outlook: the Future of Ring-Linear Coding . . . . .	233

9 Addendum: the Non-commutative Case . . . . .	234
References . . . . .	236
<b>Gröbner Bases over Commutative Rings and Applications to Coding Theory . . . . .</b> 239	
Eimear Byrne and Teo Mora	
1 Introduction . . . . .	239
2 Gröbner Basis over Commutative Rings: the Lost Lore . . . . .	240
2.1 Notation . . . . .	240
2.2 Zacharias Rings . . . . .	244
2.3 Möller: Gröbner Basis over a Principal Ideal Ring . . . . .	245
2.4 Spear's Theorem . . . . .	247
2.5 Szekeres Ideals . . . . .	247
3 Finite Chain Rings . . . . .	248
4 Solving a Key Equation . . . . .	249
5 Alternant Codes . . . . .	252
5.1 Unique Decoding $C$ for the Hamming Distance . . . . .	253
5.2 Unique Decoding of $C$ for the Lee Distance . . . . .	255
5.3 List Decoding of $C$ for the Hamming Distance . . . . .	256
References . . . . .	258
<b>Overview of Cryptanalysis Techniques in Multivariate Public Key Cryptography . . . . .</b> 263	
Olivier Billet and Jintai Ding	
1 Introduction . . . . .	263
2 Inversion Attacks . . . . .	264
2.1 Matsumoto–Imai Scheme A and Its Variations . . . . .	265
2.2 Direct Inversion Attacks . . . . .	267
2.3 MinRank . . . . .	269
2.4 Unbalanced Oil and Vinegar . . . . .	272
2.5 Defense Mechanisms . . . . .	273
3 Structural Attacks . . . . .	274
3.1 Isomorphism of Polynomials . . . . .	275
3.2 Two Rounds . . . . .	277
4 Discussion . . . . .	279
References . . . . .	280
<b>A Survey on Polly Cracker Systems . . . . .</b> 285	
Françoise Levy-dit-Vehel, Maria Grazia Marinari, Ludovic Perret and Carlo Traverso	
1 Introduction . . . . .	285
2 The Seminal Paper . . . . .	287
2.1 Barkee's Cryptosystem . . . . .	287
2.2 The Fantomas Attack . . . . .	288
2.3 The Moriarty Attack . . . . .	288
2.4 Bulygin's Attack . . . . .	289

3	CA-Style Cryptosystems . . . . .	290
3.1	Generic Design . . . . .	290
3.2	Graph 3-Coloring . . . . .	291
3.3	Graph Perfect Code . . . . .	291
3.4	Intelligent Linear Algebra Attack . . . . .	292
3.5	EnRoot . . . . .	292
3.6	0-Evaluation Attack . . . . .	293
3.7	3-SAT . . . . .	294
4	Further Attacks . . . . .	295
4.1	Basic CCA (Steinwandt and Geiselmann 2002)	295
4.2	Differential Attack . . . . .	296
4.3	The 2-Nomial Attack . . . . .	297
4.4	Further Linear Algebra Attacks . . . . .	298
5	Polly-Two . . . . .	299
6	Non-commutative Gröbner Cryptosystems? No Thanks!	300
6.1	Non-commutative Polly Cracker . . . . .	300
6.2	Monoid Algebras . . . . .	301
6.3	Pritchard's Decryption Algorithm . . . . .	302
7	Conclusion . . . . .	303
	References . . . . .	303
	<b>Block Ciphers: Algebraic Cryptanalysis and Gröbner Bases</b> . . . . .	307
	Carlos Cid and Ralf-Philipp Weinmann	
1	Introduction . . . . .	307
2	Design of Block Ciphers . . . . .	308
3	Block Cipher Cryptanalysis . . . . .	310
4	Algebraic Cryptanalysis . . . . .	312
4.1	Polynomial Descriptions of Block Ciphers . . . . .	313
4.2	Field Equations . . . . .	314
4.3	Polynomial Systems over $\mathbb{F}_2$ . . . . .	315
4.4	Equations for Non-linear Components . . . . .	315
4.5	Equations for Inversion over $\mathbb{F}_{2^n}$ . . . . .	316
4.6	Block Cipher Embeddings . . . . .	316
4.7	Direct Construction of Gröbner Bases . . . . .	317
5	Small Scale and Experimental Ciphers . . . . .	318
5.1	Small Scale Variants of the AES . . . . .	318
5.2	Flurry and Curry . . . . .	319
5.3	Other Examples . . . . .	320
6	Experimental Results . . . . .	320
6.1	Small Versions of the AES . . . . .	320
6.2	Flurry and Curry . . . . .	321
6.3	Other Experiments . . . . .	322
7	Attack Strategies . . . . .	322
7.1	Meet-in-the-Middle and Incremental Techniques . . . . .	322
7.2	Differential-Algebraic Cryptanalysis . . . . .	323

8 Alternative Methods for Solving Polynomial Systems . . . . .	324
9 Conclusions . . . . .	325
References . . . . .	325

**Algebraic Attacks on Stream Ciphers with Gröbner Bases . . . . .** 329

Frederik Armknecht and Gwenolé Ars

1 Introduction . . . . .	329
2 Keystream Generators . . . . .	330
3 Algebraic Attacks . . . . .	333
4 Finding Equations . . . . .	336
4.1 Simple Combiners . . . . .	336
4.2 Combiners with Memory . . . . .	339
4.3 Considering Several Equations Simultaneously . . . . .	341
5 Computing Solutions . . . . .	343
5.1 Minimum Number of Outputs . . . . .	344
5.2 Time Effort . . . . .	345
6 Conclusions . . . . .	346
References . . . . .	347

**Part 2 Notes****Canonical Representation of Quasicyclic Codes Using Gröbner**

Bases Theory . . . . .	351
Kristine Lally	
1 Introduction . . . . .	351
2 Characterisation Using Gröbner Bases Theory . . . . .	352
3 Parity Check Matrix and Dual Code . . . . .	354
4 Recent Application to QC LDPC Codes . . . . .	354
References . . . . .	355

**About the  $n$ th-Root Codes: a Gröbner Basis Approach**

to the Weight Computation . . . . .	357
-------------------------------------	-----

Marta Giorgetti

1 General $n$ th-Root Codes . . . . .	357
1.1 Computing Distance and Weight Distribution for an $n$ th-Root Code . . . . .	358
2 Conclusions and Further Research . . . . .	360
References . . . . .	360

**Decoding Linear Error-Correcting Codes up to Half the Minimum**

Distance with Gröbner Bases . . . . .	361
---------------------------------------	-----

Stanislav Bulygin and Ruud Pellikaan

1 Introduction . . . . .	361
2 Matrix in MDS Form . . . . .	361
3 Decoding up to Half the Minimum Distance . . . . .	362
4 Conclusion and Future Work . . . . .	364
References . . . . .	364

<b>Gröbner Bases for the Distance Distribution of Systematic Codes . . . . .</b>	367
Eleonora Guerrini, Emmanuela Orsini and Ilaria Simonetti	
1 Preliminaries . . . . .	367
2 Theoretical Results . . . . .	368
3 Numerical Computations . . . . .	370
References . . . . .	371
<b>A Prize Problem in Coding Theory . . . . .</b>	373
Jon-Lark Kim	
1 Introduction . . . . .	373
2 Related Facts about a Putative Type II [72, 36, 16] Code . . . . .	374
3 Future Work . . . . .	375
4 Monetary Prizes . . . . .	376
References . . . . .	376
<b>An Application of Möller’s Algorithm to Coding Theory . . . . .</b>	379
M. Borges-Quintana, M.A. Borges-Trenard and E. Martínez-Moro	
1 Introduction . . . . .	379
2 An Ideal Associated with a Linear Code . . . . .	379
2.1 A Second Way of Getting the Data for I . . . . .	380
3 Examples . . . . .	381
3.1 Working out with a Gröbner Representation . . . . .	381
3.2 Combinatorial Properties of a Binary Code . . . . .	382
3.3 Example: the Golay Code . . . . .	383
3.4 GAP Computing Section . . . . .	383
References . . . . .	384
<b>Mattson–Solomon Transform and Algebra Codes . . . . .</b>	385
Edgar Martínez-Moro and Diego Ruano	
Introduction . . . . .	385
1 Mattson–Solomon Transform . . . . .	386
2 Generator Theory . . . . .	386
3 A Note on the Syndrome Variety . . . . .	388
References . . . . .	388
<b>Decoding Folded Reed–Solomon Codes Using Hensel-Lifting . . . . .</b>	389
Peter Beelen and Kristian Brander	
1 Introduction . . . . .	389
2 Folded Reed–Solomon Codes . . . . .	390
3 Decoding of Folded Reed–Solomon Codes . . . . .	390
References . . . . .	393
<b>A Note on the Generalisation of the Guruswami–Sudan List Decoding Algorithm to Reed–Muller Codes . . . . .</b>	395
Daniel Augot and Michael Stepanov	
1 Definitions and Notation . . . . .	395

2	The Algorithm . . . . .	396
3	The Analysis . . . . .	396
	References . . . . .	397
<b>Viewing Multipoint Codes as Subcodes of One-Point Codes . . . . .</b>		399
Gretchen L. Matthews		
1	Introduction . . . . .	399
2	Embedding a Multipoint Code in a One-Point Code . . . . .	400
3	Examples . . . . .	400
4	Conclusion . . . . .	402
	References . . . . .	402
<b>A Short Introduction to Cyclic Convolutional Codes . . . . .</b>		403
Heide Gluesing-Luerssen, Barbara Langfeld and Wiland Schmale		
1	Introduction and Preliminaries . . . . .	403
2	How to Define Cyclic Convolutional Codes? . . . . .	404
3	Analyzing Cyclic CC's with Gröbner-type Theory . . . . .	406
	References . . . . .	407
<b>On the Non-linearity of Boolean Functions . . . . .</b>		409
Ilaria Simonetti		
1	Introduction . . . . .	409
2	Preliminaries and Notation . . . . .	409
3	Computing the Non-linearity . . . . .	411
	References . . . . .	413
<b>Quasigroups as Boolean Functions, Their Equation Systems and Gröbner Bases . . . . .</b>		415
D. Gligoroski, V. Dimitrova and S. Markovski		
1	Introduction . . . . .	415
2	Quasigroups as Vector Valued Boolean Functions . . . . .	416
2.1	Lexicographic Ordering of Finite Quasigroups . . . . .	416
2.2	Vector Valued Boolean Functions . . . . .	416
2.3	Classification of Quasigroups . . . . .	417
3	Systems of Quasigroup Equations and Gröbner Bases . . . . .	418
	References . . . . .	420
<b>A New Measure to Estimate Pseudo-Randomness of Boolean Functions and Relations with Gröbner Bases . . . . .</b>		421
Danilo Gligoroski, Smile Markovski and Svein Johan Knapskog		
1	Introduction . . . . .	421
2	Normalized Average Number of Terms—NANT . . . . .	422
3	NANT and SHA-Family of Hash Functions . . . . .	423
	References . . . . .	425

<b>Radical Computation for Small Characteristics . . . . .</b>	427
Ryutaroh Matsumoto	
1 Introduction . . . . .	427
2 Another Radical Computation Method for Positive Characteristic . . . . .	428
3 Comparison of Computational Time and Discussion . . . . .	428
References . . . . .	430

# Gröbner Technology

Teo Mora

## 1 Notation and Definitions

$\mathbb{F}$  denotes an arbitrary field,  $\overline{\mathbb{F}}$  denotes its algebraic closure and  $\mathbb{F}_q$  denotes a finite field of size  $q$  (so  $q$  is implicitly understood to be a power of a prime) and  $\mathcal{P} := \mathbb{F}[X] := \mathbb{F}[x_1, \dots, x_n]$  the polynomial ring over the field  $\mathbb{F}$ .

For any ideal  $I \subset \mathcal{P}$  and any extension field  $\mathbb{E}$  of  $\mathbb{F}$ , let  $\mathcal{V}_{\mathbb{E}}(I)$  denote the set of the rational points of  $I$  over  $\mathbb{E}$ . We also write  $\mathcal{V}(I) = \mathcal{V}_{\overline{\mathbb{F}}}(I)$ .

Let  $\mathcal{T}$  be the set of terms in  $\mathcal{P}$ , *id est*

$$\mathcal{T} := \{x_1^{a_1} \cdots x_n^{a_n} : (a_1, \dots, a_n) \in \mathbb{N}^n\},$$

which is a multiplicative version of the additive semigroup  $\mathbb{N}^n$ , the relation between these notations being obvious: given

$$\alpha := (a_1, \dots, a_n), \quad \beta := (b_1, \dots, b_n), \quad \gamma := (c_1, \dots, c_n)$$

and the terms

$$\tau_a := X^\alpha = x_1^{a_1} \cdots x_n^{a_n}, \quad \tau_b := X^\beta = x_1^{b_1} \cdots x_n^{b_n}, \quad \tau_c := X^\gamma = x_1^{c_1} \cdots x_n^{c_n},$$

we have

$$\begin{aligned} \tau_a \cdot \tau_b = \tau_c &\iff a_i + b_i = c_i \quad \text{for each } i \iff \alpha + \beta = \gamma, \\ \tau_a \mid \tau_b &\iff a_i \leq b_i \quad \text{for each } i \iff \alpha \leq_P \beta, \end{aligned}$$

where  $\leq_P$  is the natural partial ordering over  $\mathbb{N}^n$ .

The assignment of a finite set of terms

$$G := \{\tau_1, \dots, \tau_v\} \subset \mathcal{T}, \quad \tau_i = x_1^{a_1^{(i)}} \cdots x_n^{a_n^{(i)}}$$

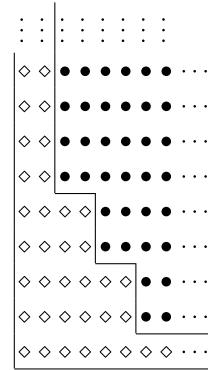
—or, equivalently of a finite set of integer vectors

$$\{a^{(1)}, \dots, a^{(v)}\} \subset \mathbb{N}^n, \quad a^{(i)} = (a_1^{(i)}, \dots, a_n^{(i)}) \in \mathbb{N}^n,$$

defines a partition of  $\mathcal{T}$  (resp.  $\mathbb{N}^n$ ) in two parts (see Fig. 1 where  $G := \{x_1^6 x_2, x_1^4 x_2^3, x_1^2 x_2^5\} \subset \mathcal{T}$ ):

---

T. Mora  
DIMA and DISI, Università di Genova, Genova, Italy  
e-mail: [theomora@dima.unige.it](mailto:theomora@dima.unige.it)

**Fig. 1** A Gröbner escalier

- $T := \{\tau \tau_i : \tau \in \mathcal{T}, 1 \leq i \leq v\} \cong \{\alpha + a^{(i)} : \alpha \in \mathbb{N}^n, 1 \leq i \leq v\} =: \Sigma$  which is a *semigroup ideal*, id est a subset  $T \subset \mathcal{T}$  (resp.  $\Sigma \subset \mathbb{N}^n$ ) such that

$$\tau \in \mathcal{T}, t \in T \implies \tau t \in T, \text{ resp. } a \in \Sigma, b \in \mathbb{N}^n, a \leq_P b \implies b \in \Sigma;$$

- ◊  $N := \mathcal{T} \setminus T \cong \mathbb{N}^n \setminus \Sigma =: \Delta$  which is an *order ideal*, id est a subset  $N \subset \mathcal{T}$  (resp.  $\Delta \subset \mathbb{N}^n$ ) such that

$$\tau \in \mathcal{T}, t \in N, \tau \mid t \implies \tau \in N, \text{ resp. } a \in \Delta, b \in \mathbb{N}^n, a \geq_P b \implies b \in \Delta.$$

Remark that the assignment of

- a finite monomial set  $G \subset \mathcal{T}$ ,
- a semigroup ideal  $T \subset \mathcal{T}$ ,
- an order ideal  $N \subset \mathcal{T}$

uniquely characterizes the other data: in fact

- $N$  and  $T$  are related by their being complementary in  $\mathcal{T}$ ,
- each semigroup ideal  $T \subset \mathcal{T}$  has a unique minimal basis  $G \subset T$  such that  $T := \{\tau \tau_i : \tau \in \mathcal{T}, \tau_i \in G\}$ ; the fact, whose proof is quite involved, that  $G$  is finite is known as Dickson's lemma but actually was already proved by Gordan (1900).

We recall that the well-orderings on  $\mathcal{T}$  which are a *semigroup ordering*, id est satisfy

$$\tau_1 < \tau_2 \implies \tau \tau_1 < \tau \tau_2 \quad \text{for each } \tau, \tau_1, \tau_2 \in \mathcal{T}$$

are called *term orderings*, even if the old-fashioned notion of *admissible ordering* can still be found somewhere.

For a free-module  $\mathcal{P}^m$ ,  $m \in \mathbb{N}$ , we denote by  $\{\mathbf{e}_1, \dots, \mathbf{e}_m\}$  its canonical basis,

$$\begin{aligned} \mathcal{T}^{(m)} &= \{t \mathbf{e}_i, t \in \mathcal{T}, 1 \leq i \leq m\} \\ &= \{x_1^{a_1} \cdots x_n^{a_n} \mathbf{e}_i, (a_1, \dots, a_n) \in \mathbb{N}^n, 1 \leq i \leq m\} \end{aligned}$$

denotes its monomial  $\mathbb{F}$ -basis and  $\prec$  denotes a well-ordering on  $\mathcal{T}^{(m)}$  which is compatible with the term-ordering  $<$  on  $\mathcal{T}$ , that is, satisfying

$$\tau_1 \leq \tau_2, \quad t_1 \preceq t_2, \quad \implies \quad \tau_1 t_1 \preceq \tau_2 t_2$$

for each  $\tau_1, \tau_2 \in \mathcal{T}, t_1, t_2 \in \mathcal{T}^{(m)}$ .

Note that  $\mathcal{T}^{(1)} = \mathcal{T}$ .

For each  $f = \sum_{\tau \in \mathcal{T}^{(m)}} c(f, \tau) \tau \in \mathcal{P}^m$ , its *support* is

$$\text{supp}(f) := \{\tau \in \mathcal{T}^{(m)} : c(f, \tau) \neq 0\},$$

its *leading term* is the term  $\mathbf{T}_<(f) := \max_{\prec}(\text{supp}(f))$ , its *leading coefficient* is  $\text{lc}_{<}(f) := c(f, \mathbf{T}_<(f))$  and its *leading monomial* is  $\mathbf{M}_<(f) := \text{lc}_{<}(f) \mathbf{T}_<(f)$ .

When  $\prec$  is understood we will drop the subscript, as in  $\mathbf{T}(f) = \mathbf{T}_<(f)$ .

For any set  $F \subset \mathcal{P}^m$ , write

- $\mathbf{T}\{F\} := \mathbf{T}_<\{F\} := \{\mathbf{T}(f) : f \in F\};$
- $\mathbf{M}\{F\} := \mathbf{M}_<\{F\} := \{\mathbf{M}(f) : f \in F\};$
- $\mathbf{T}(F) := \mathbf{T}_<(F) := \{\tau \mathbf{T}(f) : \tau \in \mathcal{T}, f \in F\}$ , a *monomial module*<sup>1</sup>;
- $\mathbf{N}(F) := \mathbf{N}_<(F) := \mathcal{T}^{(m)} \setminus \mathbf{T}_<(F)$ , an *order module*<sup>2</sup>;
- $\mathbb{I}(F) = \langle F \rangle$  the module generated by  $F$ .

Remark that, if  $m = 1$ , the assignment of  $\mathbf{T}\{F\}$  gives the partition  $\mathcal{T} = \mathbf{T}(F) \sqcup \mathbf{N}(F)$  discussed above, that the related semigroup ideal  $\mathbf{T}(F)$  is also denoted  $\Sigma(F)$  while the related order ideal  $\mathbf{N}(F)$  is also denoted  $\Delta(F)$  and labelled  $\Delta$ -set or *footprint*. When  $F$  is the Gröbner basis of the module  $\mathbb{I}(F)$  it generates,  $\mathbf{N}(F)$  is called the *Gröbner escalier* (Galligo 1974) of  $\mathbb{I}(F)$ .

We can now induce a finer partition of  $\mathcal{T}^{(m)}$  in terms of a module  $M \subset \mathcal{P}^m$  and a term-ordering  $\prec$ , by defining (see Fig. 2 where this time we have set  $M := \mathbb{I}(x_1^6, x_1^4 x_2^3, x_2^5) \subset \mathcal{P}$ )

- ◊  $\mathbf{N}_<(M) = \mathcal{T}^{(m)} \setminus \mathbf{T}_<(M)$  its *Gröbner escalier*;
- $\mathbf{B}_<(M) := \{x_h \tau : 1 \leq h \leq n, \tau \in \mathbf{N}_<(M) \setminus \mathbf{N}_<(M)\}$ , its *border set*;
- $\mathbf{J}_<(M) := \mathbf{T}_<(M) \setminus \mathbf{B}_<(M)$ ,
- \*  $\mathbf{G}_<(M) \subset \mathbf{B}_<(M)$  the unique minimal basis of  $\mathbf{T}_<(M)$ ,
- $\mathbf{C}_<(M) := \{\tau \in \mathbf{N}_<(M) : x_h \tau \in \mathbf{T}_<(M), \forall h\}$  its *corner set*.

Under this notation, the following properties are trivially satisfied:

**Lemma 1** *It holds*

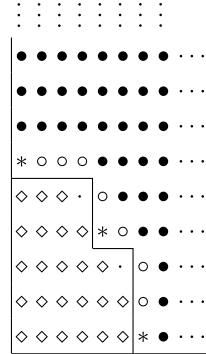
1.  $\mathbf{T}_<(M) = \{\tau \in \mathcal{T} : \exists g \in M : \mathbf{T}_<(g) = \tau\};$
2.  $\mathbf{J}_<(M) = \{\tau \in \mathbf{T}_<(M) : x_i \mid \tau \implies \frac{\tau}{x_i} \in \mathbf{T}_<(M)\};$
3.  $\mathbf{B}_<(M) = \{\tau \in \mathbf{T}_<(M) : \exists x_i \mid \tau, \frac{\tau}{x_i} \in \mathbf{N}_<(M)\};$

---

<sup>1</sup>Id est a subset  $T \subset \mathcal{T}^{(m)}$  such that  $\tau \in \mathcal{T}, t \in T \implies \tau t \in T$ .

<sup>2</sup>Id est a subset  $N \subset \mathcal{T}^{(m)}$  such that  $\tau \in \mathcal{T}, \tau t \in N \implies t \in N$ .

**Fig. 2** A refined Gröbner escalier



4.  $\mathbf{G}_\prec(\mathbf{M}) = \{\tau \in \mathbf{T}_\prec(\mathbf{M}) : \forall x_i \mid \tau, \frac{\tau}{x_i} \in \mathbf{N}_\prec(\mathbf{M})\};$
5.  $\mathbf{C}_\prec(\mathbf{M}) = \{\tau \in \mathbf{N}_\prec(\mathbf{M}) : \forall i, x_i \tau \in \mathbf{B}_\prec(\mathbf{M})\};$
6.  $\mathbf{N}_\prec(\mathbf{M}) = \{\tau \in \mathcal{T} : \exists g \in \mathbf{M} : \mathbf{T}_\prec(g) = \tau\};$
7.  $\mathbf{C}_\prec(\mathbf{M}) \cup \mathbf{T}_\prec(\mathbf{M})$  is a monomial module;
8.  $\mathbf{N}_\prec(\mathbf{M}) \cup \mathbf{G}_\prec(\mathbf{M})$  and  $\mathbf{N}_\prec(\mathbf{M}) \cup \mathbf{B}_\prec(\mathbf{M})$  are order modules.
9.  $\tau \in \mathbf{J}_\prec(\mathbf{M}) \iff \forall x_i \mid \tau, \frac{\tau}{x_i} \in \mathbf{T}_\prec(\mathbf{M});$
10.  $\tau \in \mathbf{B}_\prec(\mathbf{M}) \setminus \mathbf{G}_\prec(\mathbf{M}) \iff \exists h, H : \frac{\tau}{x_h} \in \mathbf{N}_\prec(\mathbf{M}), \frac{\tau}{x_H} \in \mathbf{B}_\prec(\mathbf{M}) \subset \mathbf{T}_\prec(\mathbf{M});$
11.  $\tau \in \mathbf{B}_\prec(\mathbf{M}) \setminus \mathbf{G}_\prec(\mathbf{M}) \implies \forall x_i \mid \tau, \frac{\tau}{x_i} \in \mathbf{N}_\prec(\mathbf{M}) \cup \mathbf{B}_\prec(\mathbf{M});$
12.  $\tau \in \mathbf{N}_\prec(\mathbf{M}) \cup \mathbf{G}_\prec(\mathbf{M}) \iff \forall x_i \mid \tau, \frac{\tau}{x_i} \in \mathbf{N}_\prec(\mathbf{M});$
13.  $\tau \in \mathbf{T}_\prec(\mathbf{M}) \cup \mathbf{C}_\prec(\mathbf{M}) \iff \forall i, x_i \tau \in \mathbf{T}_\prec(\mathbf{M});$
14.  $\tau \in \mathbf{N}_\prec(\mathbf{M}) \setminus \mathbf{C}_\prec(\mathbf{M}) \iff \exists h : x_h \tau \in \mathbf{N}_\prec(\mathbf{M}).$

**Lemma 2** Let  $\mathbf{N}$  be a finitely generated  $\mathcal{P}$ -module,  $\Phi : \mathcal{P}^m \mapsto \mathbf{N}$  be any surjective morphism and set  $\mathbf{M} := \ker(\Phi)$ . Then

1.  $\mathcal{P}^m \cong \mathbf{M} \oplus \text{Span}_{\mathbb{F}}(\mathbf{N}(\mathbf{M}));$
2.  $\mathbf{N} \cong \text{Span}_{\mathbb{F}}(\mathbf{N}(\mathbf{M}));$
3. for each  $f \in \mathcal{P}^m$ , there is a unique  $g := \text{Can}(f, \mathbf{M}, \prec) \in \text{Span}_{\mathbb{F}}(\mathbf{N}(\mathbf{M}))$  such that  $f - g \in \mathbf{M}$ .

Such  $g$  is called the canonical form of  $f$  w.r.t.  $\mathbf{M}$  and satisfies also:

- (a)  $\text{Can}(f_1, \mathbf{M}, \prec) = \text{Can}(f_2, \mathbf{M}, \prec) \iff f_1 - f_2 \in \mathbf{M};$
- (b)  $\text{Can}(f, \mathbf{M}, \prec) = 0 \iff f \in \mathbf{M}.$

**Definition 3** Let  $\mathbf{N}$  be a finitely generated  $\mathcal{P}$ -module,  $\Phi : \mathcal{P}^m \mapsto \mathbf{N}$  be any surjective morphism and set  $\mathbf{M} := \ker(\Phi)$ .

Let  $G \subset \mathbf{M}$ ,  $f, h, f_1, f_2 \in \mathcal{P}^m$ . Then

1.  $G$  will be called a Gröbner basis of  $\mathbf{M}$  if

$$\mathbf{T}(G) = \mathbf{T}(\mathbf{M}),$$

that is,  $\mathbf{T}\{G\} := \{\mathbf{T}(g) : g \in G\}$  generates  $\mathbf{T}(\mathbf{M}) = \mathbf{T}\{\mathbf{M}\}$ .

2. For each  $f_1, f_2 \in \mathcal{P}^m$  such that

$$\mathbf{T}(f_1) = t_1 \mathbf{e}_{l_1}, \quad \mathbf{T}(f_2) = t_2 \mathbf{e}_{l_2},$$

the *S-polynomial of  $f_1$  and  $f_2$*  exists only if  $\mathbf{e}_{l_1} = \mathbf{e}_{l_2} := \epsilon$ , in which case it is

$$S(f_1, f_2) := \text{lc}(f_2)^{-1} \frac{\delta(f_1, f_2)}{t_2} f_2 - \text{lc}(f_1)^{-1} \frac{\delta(f_1, f_2)}{t_1} f_1,$$

where  $\delta := \delta(f_1, f_2) := \text{lcm}(t_1, t_2)$ ;  $\delta\epsilon$  is called the *formal term* of  $S(f_1, f_2)$ .

3.  $f$  has a *Gröbner representation*  $\sum_{i=1}^{\mu} p_i g_i$  in terms of  $G$  if<sup>3</sup>

$$f = \sum_{i=1}^{\mu} p_i g_i, \quad p_i \in \mathcal{P}, \quad g_i \in G, \quad \mathbf{T}(p_i) \mathbf{T}(g_i) \preceq \mathbf{T}(f), \quad \text{for each } i.$$

4.  $f$  has the (*strong*) *Gröbner representation*  $\sum_{i=1}^{\mu} c_i t_i g_i$  in terms of  $G$  if

$$f = \sum_{i=1}^{\mu} c_i t_i g_i, \quad c_i \in \mathbb{F} \setminus \{0\}, \quad t_i \in \mathcal{T}, \quad g_i \in G,$$

with  $\mathbf{T}(f) = t_1 \mathbf{T}(g_1) \succ \dots \succ t_i \mathbf{T}(g_i) \succ \dots$ .

5.  $f$  has the *weak Gröbner representation*  $\sum_{i=1}^{\mu} c_i t_i g_i$  in terms of  $G$  if

$$f = \sum_{i=1}^{\mu} c_i t_i g_i, \quad c_i \in \mathbb{F} \setminus \{0\}, \quad t_i \in \mathcal{T}, \quad g_i \in G,$$

with  $\mathbf{T}(f) = t_1 \mathbf{T}(g_1) \succeq \dots \succeq t_i \mathbf{T}(g_i) \succeq \dots$ .

6. For any  $f_1, f_2 \in \mathcal{P}^m$ , whose S-polynomial exists and has  $\delta\epsilon$  as formal term, we say that  $S(f_1, f_2)$  has a *quasi-Gröbner representation* in terms of  $G$  if it can be written as  $S(g, f) = \sum_{k=1}^{\mu} p_k g_k$ , with  $p_k \in \mathcal{P}, g_k \in G$  and  $\mathbf{T}(p_k) \mathbf{T}(g_k) \prec \delta\epsilon$  for each  $k$ .

7.  $h := \text{NF}_{\prec}(f, G)$  is called a *normal form* of  $f$  w.r.t.  $G$ , if

- $f - h \in \mathbb{I}(G)$  has a strong Gröbner representation in terms of  $G$  and
- $h \neq 0 \implies \mathbf{T}(h) \notin \mathbf{T}(G)$ .

8. The *reduced Gröbner basis* of  $M$  wrt  $\prec$  is the set

$$\{\tau - \text{Can}(\tau, M, \prec) : \tau \in \mathbf{G}_{\prec}(M)\}.$$

9. The *border basis* of  $M$  w.r.t.  $\prec$  is the set

$$\{\tau - \text{Can}(\tau, M, \prec) : \tau \in \mathbf{B}_{\prec}(M)\}.$$

---

<sup>3</sup>Note that here, unlike in (4), we are not assuming  $i \neq j \implies \mathbf{T}(p_i) \mathbf{T}(g_i) \neq \mathbf{T}(p_j) \mathbf{T}(g_j)$ ; moreover both here, in (4) and in (5) a same element of  $G$  can repeatedly appear.

10. A *Gröbner representation* of  $\mathbf{M}$  is the assignment of

- a linearly independent set  $\mathbf{q} = \{q_1, \dots, q_s\}$  ( $q_1 = 1$ ), where  $s = \#(\mathbf{N}(\mathbf{M}))$ , such that  $\mathcal{P}^m/\mathbf{M} = \text{Span}_{\mathbb{F}}(\mathbf{q})$ ,
- the set

$$\mathcal{M} = \mathcal{M}(\mathbf{q}) := \{(a_{lj}^{(h)}) \in \mathbb{F}^{s^2}, 1 \leq h \leq n\}$$

of the  $s \times s$  square matrices  $(a_{lj}^{(h)})$  defined by the equalities

$$x_h q_l = \sum_j a_{lj}^{(h)} q_j, \quad \forall l, j, h, 1 \leq l, j \leq s, 1 \leq h \leq n$$

in  $\mathcal{P}^m/\mathbf{M} = \text{Span}_{\mathbb{F}}(\mathbf{q})$ .

11. For each  $f \in \mathcal{P}$  the *Gröbner description* of  $f$  in terms of a Gröbner representation  $(\mathbf{q}, \mathcal{M})$  is the unique vector

$$\mathbf{Rep}(f, \mathbf{q}) := (\gamma(f, q_1, \mathbf{q}), \dots, \gamma(f, q_s, \mathbf{q})) \in \mathbb{F}^s$$

such that  $f - \sum_j \gamma(f, q_j, \mathbf{q}) q_j \in \mathbf{M}$ .

12. The *linear representation* of  $\mathbf{M}$  w.r.t.  $\prec$  is the Gröbner representation  $(\mathbf{N}_\prec(\mathbf{M}), \mathcal{M}(\mathbf{N}_\prec(\mathbf{M})))$  where  $\mathbf{q} = \mathbf{N}_\prec(\mathbf{M})$ .

With these definitions, if  $\mathbf{N}_\prec(\mathbf{M}) = \{\tau_1, \dots, \tau_s\}$ , the *Gröbner description*

$$\mathbf{Rep}(f, \mathbf{N}_\prec(\mathbf{M})) := (\gamma(f, \tau_1, \mathbf{N}_\prec(\mathbf{M})), \dots, \gamma(f, \tau_s, \mathbf{N}_\prec(\mathbf{M})))$$

of  $f$  in terms of the linear representation of  $\mathbf{M}$  w.r.t.  $\prec$  is a convoluted synonym of the notion of canonical form

$$\text{Can}(f, \mathbf{M}, \prec) = \sum_{j=1}^s \gamma(f, \tau_j, \prec) \tau_j = \sum_{j=1}^s \gamma(f, \tau_j, \mathbf{N}_\prec(\mathbf{M})) \tau_j$$

of  $f$  in terms of  $\prec$ .

## 2 Term-Orderings: Classification and Representation

**Definition 4** A *weight function*  $v_w : \mathcal{T} \mapsto \mathbb{R}$  on  $\mathcal{T}$  and  $\mathcal{P}$  is the assignment of a vector  $w := (w_1, \dots, w_n) \in \mathbb{R}^n$ ,  $w_i \geq 0$ , so that  $v_w(X^a) = w \cdot a = \sum_i w_i a_i$ .

**Theorem 5** (Erdős 1956) *Each semigroup ordering  $<$  on  $\mathcal{T}$  is characterized by assigning  $r \leq n$  linearly independent vectors*

$$w_1, \dots, w_j := (w_{j1}, \dots, w_{jn}), \dots, w_r \in \mathbb{R}^n$$

—or equivalently an  $r \times n$  matrix  $(w_{ji}) \in \mathbb{R}^{rn}$  of maximal rank—so that for each  $\tau_a := X^a$ ,  $\tau_b := X^b$  in  $\mathcal{T}$ , we have

$$\tau_a < \tau_b \iff \exists j: w_j \cdot a < w_j \cdot b \quad \text{and} \quad w_i \cdot a = w_i \cdot b \quad \text{for } i < j.$$

Moreover, such an ordering is a well-ordering iff, for each  $i$ ,  $X_i > 1$ , that is iff, for each  $i$ ,  $w_{ji} > 0$ , where  $j$  denotes the minimal value for which  $w_{ji} \neq 0$ .

Finally, if  $M_1, M_2$  are two  $r \times n$  matrices, then they characterize the same ordering  $<$  iff there is an invertible  $r$ -square matrix  $A = (a_{ij})$  such that

$$M_1 = AM_2 \quad \text{and} \quad a_{ij} = \begin{cases} 0 & \text{if } i < j, \\ 1 & \text{if } i = j. \end{cases}$$

Among the term-orderings we will quote those which have common and practical use, also for applications.

- The **lexicographical** (lex) ordering induced by  $X_1 < X_2 < \dots < X_n$  is defined by

$$X_1^{a_1} \cdots X_n^{a_n} < X_1^{b_1} \cdots X_n^{b_n} \iff \exists j: a_j < b_j \quad \text{and} \quad a_i = b_i \quad \text{for } i > j;$$

it has good elimination properties since it allows to compute all the elimination ideals  $I \cap \mathbb{F}[X_1, \dots, X_i]$ :

**Fact 6** If  $G$  is the Gröbner basis of  $I \subset \mathbb{F}[X_1, \dots, X_n]$  w.r.t. lex then, for each  $i \leq n$ ,  $G \cap \mathbb{F}[X_1, \dots, X_i]$  is the Gröbner basis of  $I \cap \mathbb{F}[X_1, \dots, X_i]$  w.r.t. lex.

- Note that the lexicographical ordering depends on a chosen ordering imposed on the variables; recently many authors prefer using the lexicographical ordering induced by  $X_1 > X_2 > \dots > X_n$  which is defined by

$$X_1^{a_1} \cdots X_n^{a_n} < X_1^{b_1} \cdots X_n^{b_n} \iff \exists j: a_j < b_j \quad \text{and} \quad a_i = b_i \quad \text{for } i < j.$$

- The **reverse lexicographical** (rev-lex) ordering induced by  $X_1 < X_2 < \dots < X_n$  is defined by

$$X_1^{a_1} \cdots X_n^{a_n} < X_1^{b_1} \cdots X_n^{b_n} \iff \exists j: a_j > b_j \quad \text{and} \quad a_i = b_i \quad \text{for } i < j;$$

it is not a well-ordering since  $\dots < X_i^{d+1} < X_i^d < \dots < X_1 < 1$ .

- The **deg-rev-lex**<sup>4</sup> (degree reverse lexicographical) ordering induced by  $X_1 < X_2 < \dots < X_n$  is the one where terms are first compared by their degree and the ties are solved using rev-lex: it is defined by

$$X^a < X^b \iff \exists j: a_j > b_j \quad \text{and} \quad a_i = b_i \quad \text{for } 0 \leq i < j,$$

where we set  $a_0 := -\sum_i a_i$ ,  $b_0 := -\sum_i b_i$  and has the following property

---

<sup>4</sup>Often shorthanded as *drl*.

**Fact 7** Denoting, for each  $i \leq n$ ,  $\pi_i : \mathcal{T} \mapsto \mathcal{T} \cap \mathbb{F}[X_1, \dots, X_i]$  the projection<sup>5</sup> defined by

$$\pi_i(X_j) := \begin{cases} X_j & \text{if } j > i \\ 1 & \text{if } j \leq i \end{cases}$$

then any two terms  $t_1, t_2 \in \mathcal{T}$  satisfy

$$t_1 < t_2 \iff \exists j: d_{j1} < d_{j2}, \quad \text{and} \quad d_{i1} = d_{i2} \quad \text{for each } i < j$$

where we have set  $d_{ji} := \deg(\pi_j(t_i))$ .

- Naturally, also the definitions of the rev-lex and deg-rev-lex orderings depend on a chosen ordering imposed on the variables; thus, the deg-rev-lex ordering induced by  $X_1 > X_2 > \dots > X_n$  is defined as

$$X^a < X^b \iff \exists j: a_j > b_j \quad \text{and} \quad a_i = b_i \quad \text{for } n+1 \geq i > j,$$

where we set  $a_{n+1} := -\sum_i a_i$ ,  $b_{n+1} := -\sum_i b_i$ .

- More in general, given an ordering  $<$  on  $\mathcal{T}$  its **degree extension** is the ordering  $\prec$  defined as

$$t_1 \prec t_2 \iff \deg(t_1) < \deg(t_2) \quad \text{or} \quad \deg(t_1) = \deg(t_2), \quad t_1 < t_2.$$

- If we have a weight vector  $w := (w_1, \dots, w_n) \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  and a term ordering  $<$ , the construction leading to the degree extension of  $<$  can be performed to lead to the **weight extension**  $\prec$  of  $<$  (or the *refinement* of  $v_w$  with  $<$ ) defined as

$$t \prec T \iff v_w(t) < v_w(T) \quad \text{or} \quad v_w(t) = v_w(T), \quad t < T.$$

Bayer and Stillman (1987) proved that the rev-lex ordering is the ‘most efficient’ refinement of a weight function  $v_w$ .

Given a term-ordering  $<$  on  $\mathcal{T}$ , a  $<$ -compatible well-ordering  $\prec$  on  $\mathcal{T}^{(m)}$  can be defined in different ways; we limit ourselves to quote the more standard constructions referring to Carrà Ferro and Sit (1994), Caboara and Silvestri (1999) for a more general treatment: setting an ordering  $\ll$  on the canonical basis  $\{\mathbf{e}_1, \dots, \mathbf{e}_m\}$ ,

- the **TOP** (term over position) ordering is defined as

$$t_1 \mathbf{e}_{l_1} \prec t_2 \mathbf{e}_{l_2} \iff t_1 < t_2 \quad \text{or} \quad t_1 = t_2, \mathbf{e}_{l_1} \ll \mathbf{e}_{l_2};$$

- the **POT** (position over term) ordering is defined as

$$\mathbf{e}_{l_1} \prec \mathbf{e}_{l_2} \iff \mathbf{e}_{l_1} \ll \mathbf{e}_{l_2} \quad \text{or} \quad \mathbf{e}_{l_1} = \mathbf{e}_{l_2}, t_1 < t_2.$$

---

<sup>5</sup>Obviously  $\pi_0$  is just the identity.

---

```

 $(g, \sum_{i=1}^{\mu} c_i t_i g_i) := \text{NormalForm}(f, G)$ 
 $g := f, i := 0,$ 
While  $\mathbf{T}(g) \in \mathbf{T}(G)$  do
  Let  $t \in \mathcal{T}, \gamma \in G : t\mathbf{T}(\gamma) = \mathbf{T}(g),$ 
   $i := i + 1, c_i := \frac{\text{lc}(g)}{\text{lc}(\gamma)}, t_i := t, g_i := \gamma, g := g - c_i t_i g_i.$ 
   $\mu := i$ 

```

---

**Fig. 3** Buchberger normal form algorithm

### 3 Buchberger's Theorem and Algorithm

The *Buchberger Normal Form Algorithm* (Buchberger 1965, 1970, 1998, 2006) (see Fig. 3) is a Gaussian-like linear algebra reduction which, given a finite set  $F \subset \mathcal{P}^m$  and an element  $f \in \mathcal{P}^m$ , returns a normal form  $g$  of  $f$  w.r.t.  $F$  and a strong Gröbner representation<sup>6</sup>  $\sum_{i=1}^{\mu} c_i t_i g_i$  of  $f - g$  in terms of  $F$ ; extending it we obtain the *Buchberger Canonical Form Algorithm* (Buchberger 1965, 1970, 1998, 2006) (see Fig. 4) which, if  $F$  is assumed to be a Gröbner basis, returns the canonical form  $g := \text{Can}(f, M, \prec) \in \text{Span}_{\mathbb{F}}(\mathbf{N}(l))$  and strong Gröbner representation  $\sum_{i=1}^{\mu} c_i t_i g_i$  of  $f - g$  in terms of  $F$ .

**Corollary 8** Let  $N$  be a finitely generated  $\mathcal{P}$ -module,  $\Phi : \mathcal{P}^m \rightarrow N$  be any surjective morphism and set  $M := \ker(\Phi)$ . Let  $G$  be a Gröbner basis of  $M$  w.r.t.  $\prec$ . Then

1. For each  $f \in \mathcal{P}^m$ ,  $f - \text{Can}(f, M)$  has a strong Gröbner representation in terms of  $G$ ;
2. The reduced Gröbner basis of  $M$  w.r.t.  $\prec$  is the unique set  $G \subset M$  such that<sup>7</sup>
  - (a)  $\mathbf{T}_{\prec}\{G\}$  is an irredundant basis of  $\mathbf{T}_{\prec}(M)$ ;
  - (b) for each  $g \in G$ ,  $\text{lc}(g) = 1$ ;
  - (c) for each  $g \in G$ ,  $g - \mathbf{T}(g) \in \text{Span}_{\mathbb{F}}(\mathbf{N}(M))$ .

On each free module  $\mathcal{P}^s$  ( $\{\mathbf{e}_1, \dots, \mathbf{e}_s\}$  denotes its canonical basis) one can impose a valuation  $v : \mathcal{P}^s \rightarrow \mathcal{T}$  by fixing  $s$  terms  $\tau_1, \dots, \tau_s$  and defining for each  $i$   $v(\mathbf{e}_i) := \tau_i$ , so that, for each  $f := (h_1, \dots, h_s) = \sum_i h_i \mathbf{e}_i$  we have  $v(f) := \max_{\prec} \{\mathbf{T}_{\prec}(h_i) \tau_i\}$ ; by definition, its *leading form*  $\mathcal{L}(f)$  is the homogeneous component (of degree  $v(f)$ )  $(v_1, \dots, v_m)$  where

$$v_i = \begin{cases} \mathbf{M}(h_i) & \text{iff } \mathbf{T}(t_i) \tau_i = v(f) \\ 0 & \text{otherwise.} \end{cases}$$

---

<sup>6</sup>The reason why *strong* Gröbner representations are pinned up among Gröbner representations of a polynomial is that the output of Buchberger Form Algorithms is necessarily *strong*.

The notion of weak Gröbner representation (Definition 3.5) has a similar rôle in the more esoteric theory of Gröbner bases for polynomials over a ring, for which the reader is directed to Byrne and Mora (2009).

<sup>7</sup>A basis which satisfies only conditions (a) and (b), but not necessarily (c), is called a *minimal Gröbner basis*.

---

$(g, \sum_{i=1}^{\mu} c_i t_i g_i) := \text{CanonicalForm}(f, G)$   
 $h := f, i := 0, g := 0,$   
**While**  $h \neq 0$  **do**  
 %%  $f = g + \sum_{i=1}^{\mu} c_i t_i g_i + h,$   
 %%  $\mathbf{T}(f - g) \geq \mathbf{T}(h);$   
 %%  $i > 0 \implies \mathbf{T}(f - g) = t_1 \mathbf{T}(g_1) > t_2 \mathbf{T}(g_2) > \dots > t_i \mathbf{T}(g_i) > \mathbf{T}(h);$   
**If**  $\mathbf{T}(h) \in \mathbf{T}(G)$  **do**  
     **Let**  $t \in \mathcal{T}, \gamma \in G : t\mathbf{T}(\gamma) = \mathbf{T}(h)$   
      $i := i + 1, c_i := \frac{\text{lcm}(h)}{\text{lcm}(\gamma)}, t_i := t, g_i := \gamma, h := h - c_i t_i g_i.$   
**Else**  
     %%  $\mathbf{T}(h) \in \mathbf{N}(M)$   
      $h := h - M(h), g := g + M(h)$   
 $\mu := i$

---

**Fig. 4** Buchberger canonical form algorithm

Such valuation is of course compatible with the natural valuation of  $\mathcal{P}$  and an element  $\sum_i h_i \mathbf{e}_i$  is homogeneous of degree  $\tau$  iff, for each  $i$

$$h_i \neq 0 \implies h_i = M(h_i) \quad \text{and} \quad \mathbf{T}(h_i)\tau_i = \tau.$$

**Definition 9** Denoting, for each set  $F \subset \mathcal{P}^s$ ,

$$\mathcal{L}\{F\} := \{\mathcal{L}(f) : f \in F\}, \quad \mathcal{L}(F) := \mathbb{I}(\mathcal{L}\{F\}),$$

given a module  $E \subset \mathcal{P}^s$ , the homogeneous module  $\mathcal{L}(E)$  is called the *leitmodul* of  $E$ . Any set  $B \subset E$  such that  $\mathcal{L}(B) = \mathcal{L}(E)$  is called a *standard basis* of  $E$  and is a basis of it.

Fixed a set  $\{g_1, \dots, g_s\} := G \subset M$ , with  $M(g_j) := c_j \tau_j \mathbf{e}_{l_j}$ , for each  $j$ , I will freely use the shorthand

$$\mathbf{T}(l_1, l_2, \dots, l_r) := \text{lcm}(\tau_i : i \in \{l_1, l_2, \dots, l_r\})\varepsilon$$

for each set  $\{l_1, l_2, \dots, l_r\} \subseteq \{1, \dots, s\}$  satisfying  $\mathbf{e}_{l_1} = \dots = \mathbf{e}_{l_r} =: \varepsilon$ ; in particular for  $i, j, k, 1 \leq i, j, k \leq s$ ,  $\mathbf{e}_{l_i} = \mathbf{e}_{l_j} = \mathbf{e}_{l_k} =: \varepsilon$ , we have

$$\mathbf{T}(i) = \mathbf{T}(g_i), \quad \mathbf{T}(i, j) := \text{lcm}(\tau_i, \tau_j)\varepsilon, \quad \mathbf{T}(i, j, k) := \text{lcm}(\tau_i, \tau_j, \tau_k)\varepsilon;$$

for each pair  $\{i, j\}$ ,  $1 \leq i < j \leq s$  for which  $\mathbf{e}_{l_i} = \mathbf{e}_{l_j} =: \varepsilon$ , I will also use the shorthand  $S(i, j)$  to denotes  $S(g_i, g_j)$ ,  $\omega(i, j) := \mathbf{T}(i, j)\varepsilon$  to denote its formal term and

$$s(i, j) := c_j^{-1} \frac{\mathbf{T}(i, j)}{\tau_j} \mathbf{e}_j - c_i^{-1} \frac{\mathbf{T}(i, j)}{\tau_i} \mathbf{e}_i.$$

If, with the current notation, we impose on the module  $\mathcal{P}^s$  the valuation  $v$  defined by  $v(\mathbf{e}_j) := \tau_j$ , we have that if  $f := \sum_j h_j \mathbf{e}_j \in \mathcal{P}^s$  satisfies  $\sum_j h_j g_j = 0$

necessarily, denoting

$$\tau\varepsilon := \max_{\prec} \{\mathbf{T}_{\prec}(h_i) \mathbf{T}_{\prec}(g_i)\} \quad \text{and} \quad I := \{i, 1 \leq i \leq s : \mathbf{T}(h_i) \mathbf{T}(g_i) = \tau\varepsilon\}$$

the homogeneous element  $\mathcal{L}(f) := \sum_j \nu_j \mathbf{e}_j \in \mathcal{P}^s$  of degree  $\tau$  satisfies

- $0 \neq \nu_j \implies j \in I$  and  $\nu_j = \mathbf{M}(h_j) =: d_j \omega_j$ ,
- $\sum_{j=1}^s \nu_j \mathbf{M}_{\prec}(g_j) = \sum_{j \in I} (d_j \omega_j) \cdot (c_j \tau_j \mathbf{e}_{l_j}) = (\sum_{j \in I} (d_j c_j) \cdot (\tau_j \omega_j)) \varepsilon = 0$ ,
- $\sum_{j \in I} d_j \operatorname{lc}(g_j) = 0$  and  $\omega_j \mathbf{T}_{\prec}(g_j) = \tau\varepsilon$  for each  $j \in I$ .

**Definition 10** Given a finite set  $G := \{g_1, \dots, g_s\} \subset \mathcal{P}^m$ ,  $\mathbf{M}(g_j) := c_j \tau_j \mathbf{e}_{l_j}$ , and denoting  $\mathfrak{S} : \mathcal{P}^s \rightarrow \mathcal{P}$  the map defined by  $\sum_{i=1}^s p_i \mathbf{e}_i \mapsto \sum_{i=1}^s p_i g_i$ :

1. each element of  $\ker(\mathfrak{S}) \subset \mathcal{P}^s$  is called a *syzygy* of  $G$ ;
2. the *syzygy module* of  $G$  is the module

$$\ker(\mathfrak{S}) := \{(p_1, \dots, p_s) : \sum_{i=1}^s p_i g_i = 0\} \subset \mathcal{P}^s;$$

3. the *natural valuation*  $v$  on  $\mathcal{P}^s$  is the one defined by  $v(\mathbf{e}_j) := \tau_j$ .

*Remark 11*

1. if  $f := (p_1, \dots, p_s)$  is a syzygy of  $G$ , then  $\mathcal{L}(f) := \sum_j \nu_j \mathbf{e}_j \in \mathcal{P}^s$  is a homogeneous syzygy of  $\mathbf{M}\{G\}$ ;
2. for each homogeneous syzygy  $\phi := \sum_j d_j \omega_j \mathbf{e}_j \in \mathcal{P}^s$  of  $\mathbf{M}\{G\}$  the element  $h := \mathfrak{S}(\phi) = \sum_j d_j \omega_j g_j \in \mathcal{P}^m$ , if is not zero, satisfies

$$\mathbf{T}(h) < v(\phi) = \omega_j \tau_j \quad \text{for each } j;$$

therefore if  $h = \sum_j p_j g_j$  is a Gröbner representation in terms of  $G$ , then

$$f := \phi - h = \phi - \sum_j p_j \mathbf{e}_j = \sum_j (d_j \omega_j - p_j) \mathbf{e}_j \in \ker(\mathfrak{S})$$

is a syzygy and satisfies  $v(f) = v(\phi)$  and  $\mathcal{L}(f) = \phi$ ;

3. for each  $i, j, 1 \leq i < j \leq s$ , for which  $S(i, j)$  exists and has  $\omega(i, j) := \mathbf{T}(i, j)\varepsilon$  as formal term, it holds  $\mathcal{L}(S(i, j)) = s(i, j)$  which is a homogeneous element of degree  $\mathbf{T}(i, j)$ ;
4. conversely  $S(i, j) = \mathfrak{S}(s(i, j))$ ;
5. denoting  $\mathfrak{B} := \{\{i, j\} : 1 \leq i < j \leq s, S(i, j) \text{ exists}\}$ ,  $\{s(i, j) : \{i, j\} \in \mathfrak{B}\}$  is a homogeneous basis of the syzygy module of  $\mathbf{M}\{G\}$ .

**Lemma 12** (Buchberger's First Criterion 1979) *With the present notation, under the assumption that  $\mathbf{M}$  is an ideal, it holds*

$$\mathbf{T}(i) \mathbf{T}(j) = \mathbf{T}(i, j) \implies \operatorname{NF}(S(i, j), G) = 0.$$

**Lemma 13** (Buchberger's Second Criterion 1979) For  $i, j, 1 \leq i < j \leq s$ ,  $\mathbf{e}_{l_i} = \mathbf{e}_{l_j} =: \varepsilon$ , if there is  $k, 1 \leq k \leq s$ :  $\mathbf{T}(k) \mid \mathbf{T}(i, j)$ ,—so that in particular  $\mathbf{e}_{l_k} = \varepsilon$ ,—and  $S(i, k)$  and  $S(k, j)$  have a quasi-Gröbner representation in terms of  $G$ , then also  $S(i, j)$  has a quasi-Gröbner representation.

**Definition 14** (Gebauer and Möller 1985, 1988) Denoting  $\mathfrak{B} := \{\{i, j\} : 1 \leq i < j \leq s, S(i, j) \text{ exists}\}$  and

$$\mathfrak{B}_1 := \begin{cases} \{\{i, j\} : \mathbf{T}(i)\mathbf{T}(j) = \mathbf{T}(i, j)\} & \text{iff } M \text{ is an ideal,} \\ \emptyset & \text{otherwise} \end{cases}$$

a subset  $\mathfrak{GM} \subset \mathfrak{B} \setminus \mathfrak{B}_1$  is called a *Gebauer–Möller set* for  $G$  iff the set  $\{s(i, j) : \{i, j\} \in \mathfrak{GM} \cup \mathfrak{B}_1\}$  is a homogeneous basis of the syzygy module of  $\mathbf{M}\{G\}$ .

**Theorem 15** (Buchberger) Let  $M \subset \mathcal{P}^m$  be a sub-module, and  $\{g_1, \dots, g_s\} =: G \subset M$ , with  $\mathbf{T}(g_j) := \tau_j \mathbf{e}_{l_j}$  and wlog  $\text{lc}(g_j) = 1$  for each  $j$ ; denoting  $\mathfrak{B}$  and  $\mathfrak{B}_1$  as in Definition 14 and  $\mathfrak{GM} \subset \mathfrak{B} \setminus \mathfrak{B}_1$  any Gebauer–Möller set for  $G$ , the following conditions are equivalent:

1.  $G$  is a Gröbner basis of  $M$ ;
2.  $f \in M \iff$  it has a Gröbner representation in terms of  $G$ ;
3.  $f \in M \iff$  it has a strong Gröbner representation in terms of  $G$ ;
4. for each  $f \in \mathcal{P}^m \setminus \{0\}$  and any normal form  $h := \text{NF}(f, G)$  of  $f$  w.r.t.  $G$ ,  $f \in M \iff h = 0$ ;
5. for each  $f \in \mathcal{P} \setminus \{0\}$ ,  $f - \text{Can}(f, M)$  has a strong Gröbner representation in terms of  $G$ ;
6. for each  $i, j, 1 \leq i < j \leq s$ , the  $S$ -polynomial  $S(i, j)$  (if it exists) has a quasi-Gröbner representation in terms of  $G$ .
7. for each homogeneous basis  $B$  of the syzygy module of  $\mathbf{M}\{G\}$  and for each element  $\phi \in B$ , there is a syzygy  $f_\phi \in \ker(\mathfrak{S})$  of  $G$ , such that  $\mathcal{L}(f_\phi) = \phi$ ;
8. for each  $\{i, j\} \in \mathfrak{GM}$ , the  $S$ -polynomial  $S(i, j)$  has a quasi-Gröbner representation in terms of  $G$ .
9. for each  $\{i, j\} \in \mathfrak{GM}$  the  $S$ -polynomial  $S(i, j)$  has a Gröbner representation in terms of  $G$ .

**Corollary 16** With the present notation and under the equivalent conditions of Theorem 15, the set

$$\{f_\phi : \phi \in \mathfrak{GM}\} \cup \left\{ c_j^{-1} \frac{\mathbf{T}(i, j)}{\tau_j} \mathbf{e}_j - c_i^{-1} \frac{\mathbf{T}(i, j)}{\tau_i} \mathbf{e}_i : (i, j) \in \mathfrak{B}_1 \right\}$$

is a standard basis of  $\ker(\mathfrak{S})$ .

**Lemma 17** (Möller 1988) For each  $i, j, k : 1 \leq i, j, k \leq s$ ,  $\mathbf{e}_{l_i} = \mathbf{e}_{l_j} = \mathbf{e}_{l_k}$ , it holds

$$\frac{\text{lcm}(\tau_i, \tau_j, \tau_k)}{\text{lcm}(\tau_i, \tau_k)} S(i, k) - \frac{\text{lcm}(\tau_i, \tau_j, \tau_k)}{\text{lcm}(\tau_i, \tau_j)} S(i, j) + \frac{\text{lcm}(\tau_i, \tau_j, \tau_k)}{\text{lcm}(\tau_k, \tau_j)} S(k, j) = 0.$$

**Corollary 18** (Gebauer and Möller 1985, 1988) (Compare Lemma 13)

Under the assumption of Lemma 17 if the equivalent conditions  $\mathbf{T}(i, j, k) = \mathbf{T}(i, j)$  and  $\mathbf{T}(k) \mid \mathbf{T}(i, j)$  are satisfied and both  $S(i, k)$  and  $S(k, j)$  have a quasi-Gröbner representation in terms of  $G$ , then also  $S(i, j)$  has a quasi-Gröbner representation.

**Proposition 19** (Gebauer and Möller 1985, 1988) With the present notation, denote

$$\mathfrak{GM}_* \subset \{\{i, j\}, 1 \leq i < j < s\} \text{ a Gebauer–Möller set for } \{g_1, \dots, g_{s-1}\}$$

$$\mathfrak{B}_2 := \{\{i, j\} \in \mathfrak{GM}_* : \mathbf{T}(i, j, s) = \mathbf{T}(i, j), \mathbf{T}(i, s) \neq \mathbf{T}(i, j) \neq \mathbf{T}(j, s)\}.$$

Let  $\mathsf{T} := \{\mathbf{T}(j, s) : 1 \leq j < s\}$  and  $\mathsf{T}' \subset \mathsf{T}$  be the set of the elements  $\tau \in \mathsf{T}$  such that either

- exists  $\tau' \in \mathsf{T} : \tau' \mid \tau \neq \tau'$  or
- (in case  $\mathsf{M}$  is an ideal) exists  $i_\tau : 1 \leq i_\tau < s, \mathbf{T}(i_\tau)\mathbf{T}(s) = \mathbf{T}(i_\tau, s) = \tau$ ;

for each  $\tau \in \mathsf{T} \setminus \mathsf{T}'$  choose  $i_\tau, 1 \leq i_\tau < s$ , such that  $\mathbf{T}(i_\tau, s) = \tau$  and define

$$\mathfrak{B}_3(G) := \{\{i_\tau, s\} : \tau \in \mathsf{T} \setminus \mathsf{T}'\}.$$

Then  $(\mathfrak{GM}_* \setminus \mathfrak{B}_2) \cup \mathfrak{B}_3(G)$  is a Gebauer–Möller set for  $G$

Thus, given a finite basis  $F := \{g_1, \dots, g_s\} \subset \mathsf{M}$ , the Buchberger Algorithm (Fig. 5) returns a Gröbner basis  $G$  of  $\mathsf{M}$  by iteratively forcing condition (9) of Theorem 15 and applying Proposition 19 in order to efficiently remove the so called *useless pairs*, *id est* those which are known, for theoretical reasons (Lemmas 12 and 13, Corollary 18), having 0 as normal form.

---

```

(G) := GröbnerBasis(F)
where
  F := {g1, ..., gs} ⊂ P \ {0},
  G is a Gröbner basis of the ideal I(F);
  G := {g1, g2}, B := ∅
  If T(1)T(2) ≠ T(1, 2) then B := B ∪ {{1, 2}}
  For each r, 3 ≤ r ≤ s do
    G := G ∪ {gr}
    B2 := {{i, j} ∈ B : T(i, j), T(i, r) ≠ T(i, j) ≠ T(j, r)}
    B := (B \ B2) ∪ B3(G)
  While B ≠ ∅ do
    Choose {i, j} ∈ B, B := B \ {{i, j}}, h := S(i, j)
    (h, ∑i=1^μ ci ti gi) := NormalForm(h, G)
    If h ≠ 0 then
      s := s + 1, gs := h, G := G ∪ {gs}
      B2 := {{i, j} ∈ B : T(s), T(i, j), T(i, s) ≠ T(i, j) ≠ T(j, s)}
      B := (B \ B2) ∪ B3(G)

```

---

**Fig. 5** Buchberger's algorithm (sketch)

Figure 5 is a poor sketch of the standard implementation, whose description can be found in Giovini et al. (1991) and which is mainly based on Traverso's analysis (Traverso and Donato 1989); the reader is suggested to consider the recently proposed new implementation (Brickenstein 2005) of Buchberger's algorithm, Faugère's algorithms F<sub>4</sub> (Faugére 1999) and F<sub>5</sub> (Faugére 2002) which compute Gröbner basis by a strongly improved version of Macaulay's algorithms (Macaulay 1913, 1916) and Gerdt–Blinkov (Zarkov 1996; Gerdt and Blinkov 1998a, 1998b) algorithm which computes Gröbner basis via an adaptation of Janet's notion of *complete bases* and his corresponding algorithm to compute them (Janet 1920).

Since often a lex Gröbner basis computation is either infeasible or time-consuming, it is efficient to deduce the required lex Gröbner basis from the feasible degrevlex one via elementary linear algebra (see Mora 2009).

**Acknowledgements** For their comments and suggestions, the author thanks all the authors of this book and especially M. Sala.

## References

- D. Bayer and M. Stillman, *A theorem on refining division orders by the reverse lexicographic order*, Duke Math. J. **55** (1987), nos. 2, 321–328.
- M. Brickenstein, *Gröbner bases with slim polynomials*, Reports in Comp. Alg. 35, Univ. Kaiserslautern, Kaiserslautern, 2005, <http://www.mathematik.uni-kl.de/>.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *A criterion for detecting unnecessary reductions in the construction of Gröbner bases*, Symbolic and algebraic computation (EUROSAM 1979), LNCS, vol. **72**, Springer, Berlin, 1979, pp. 3–21.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- E. Byrne and T. Mora, *Gröbner bases over commutative rings and applications to coding theory*, this volume, 2009, pp. 239–261.
- M. Caboara and M. Silvestri, *Classification of compatible module orderings*, J. Pure Appl. Algebra **142** (1999), nos. 1, 13–24.
- G. Carrà Ferro and W. Y. Sit, *On term-orderings and rankings*, Computational algebra (Fairfax, VA, 1993), Lecture Notes in Pure and Appl. Math., vol. **151**, Dekker, New York, 1994, pp. 31–77.
- J. Erdős, *On the structure of ordered real vector spaces*, Publ. Math. Debrecen **4** (1956), 334–343.
- J. C. Faugére, *A new efficient algorithm for computing Gröbner bases (F<sub>4</sub>)*, J. Pure Appl. Algebra **139** (1999), nos. 1–3, 61–88.
- J. C. Faugére, *A new efficient algorithm for computing Gröbner bases without reduction to zero (F<sub>5</sub>)*, Proc. of ISSAC 2002, ACM, New York, 2002, pp. 75–83.
- A. Galligo, *À propos du théorème de-préparation de Weierstrass*, Fonctions de plusieurs variables complexes, Springer, Berlin, 1974, pp. 543–579. LNM. 409.
- R. Gebauer and H. M. Möller, *A fast variant of Buchberger's algorithm*, preprint, 1985.
- R. Gebauer and H. M. Möller, *On an installation of Buchberger's algorithm*, J. Symb. Comput. **6** (1988), nos. 2–3, 275–286.

- V. P. Gerdt and Y. A. Blinkov, *Involutive bases of polynomial ideals*, Math. Comput. Simulation **45** (1998a), nos. 5–6, 519–541.
- V. P. Gerdt and Y. A. Blinkov, *Minimal involutive bases*, Math. Comput. Simulation **45** (1998b), nos. 5–6, 543–560.
- A. Giovini, T. Mora, G. Niesi, L. Robbiano and C. Traverso, “*One Sugar cube, please*” or selection strategies in the Buchberger algorithm, Proceedings of ISSAC 1991, ACM, New York, 1991, pp. 49–54.
- P. Gordan, *Les invariants des formes binaires*, Journal de Mathématiques Pure et Appliquées **6** (1900), 141–156.
- M. Janet, *Sur les systèmes d'équations aux dérivées partielles*, Journal de Mathématiques Pure et Appliquées **3** (1920), 65–151.
- F. S. Macaulay, *On the resolution of a given modular system into primary systems including some properties of Hilbert numbers*, Math. Ann. **74** (1913), no. 1, 66–121.
- F. S. Macaulay, *The Algebraic Theory of Modular Systems*, Cambridge University Press, Cambridge, 1916.
- H. M. Möller, *On the construction of Gröbner bases using syzygies*, J. Symb. Comput. **6** (1988), nos. 2–3, 345–359.
- T. Mora, *The FGFM problem and Moeller's algorithm on zero-dimensional ideals*, this volume, 2009, pp. 1–100.
- C. Traverso and L. Donato, *Experimenting the Gröbner basis algorithm with Alpi system*, Proc. of ISSAC 1989, ACM, New York 1989, pp. 192–198.
- A. Y. Zarkov, *Solving zero-dimensional involutive systems*, Proc. of MEGA 1994, Birkhäuser, Basel, 1996, pp. 389–399.

# The FGLM Problem and Möller's Algorithm on Zero-dimensional Ideals

Teo Mora

Our notation on Gröbner bases (Buchberger 1965, 2006) is from Mora (2009).

## 1 Duality

Denote  $\mathcal{P}^* := \text{Hom}_{\mathbb{F}}(\mathcal{P}, \mathbb{F})$  the  $\mathbb{F}$ -vector space of all  $\mathbb{F}$ -linear functionals  $\ell : \mathcal{P} \mapsto \mathbb{F}$  and remark that  $f \in \mathcal{P}, \ell \in \mathcal{P}^* \implies \ell(f) = \sum_{\tau \in T} c(f, \tau) \ell(\tau)$  and that  $\mathcal{P}^*$  is made a  $\mathcal{P}$ -module by defining  $\ell \cdot f \in \mathcal{P}^*$ , for each  $\ell \in \mathcal{P}^*, f \in \mathcal{P}$ , as

$$(\ell \cdot f)(g) := \ell(fg) \quad \text{for each } g \in \mathcal{P}.$$

Two sets  $\mathbb{L} = \{\ell_1, \dots, \ell_r\} \subset \mathcal{P}^*$  and  $\mathbf{q} = \{q_1, \dots, q_s\} \subset \mathcal{P}$  are said to be

- *triangular* if  $r = s$ ,  $\ell_i(q_j) = 0$ , for each  $i < j$  and  $\ell_j(q_j) \neq 0$ , for each  $j$ ;
- *biorthogonal* if

$$r = s \quad \text{and} \quad \ell_i(q_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$

For each  $\mathbb{F}$ -vector subspace  $L \subset \mathcal{P}^*$ , let

$$\mathfrak{P}(L) := \{g \in \mathcal{P} : \ell(g) = 0, \forall \ell \in L\}$$

and, for each  $\mathbb{F}$ -vector subspace  $P \subset \mathcal{P}$ , let

$$\mathfrak{L}(P) := \{\ell \in \mathcal{P}^* : \ell(g) = 0, \forall g \in P\}.$$

**Lemma 1** For each  $\mathbb{F}$ -vector subspaces  $P, P_1, P_2 \subset \mathcal{P}$  and each  $\mathbb{F}$ -vector subspaces  $L, L_1, L_2 \subset \mathcal{P}^*$  it holds

1. if  $P$  is an ideal then  $\mathfrak{L}(P)$  is a  $\mathcal{P}$ -module;
2. if  $L$  is a  $\mathcal{P}$ -module then  $\mathfrak{P}(L)$  is an ideal;
3.  $P_1 \subset P_2 \implies \mathfrak{L}(P_1) \supset \mathfrak{L}(P_2)$ ;

---

T. Mora

DIMA and DISI, Università di Genova, Genova, Italy  
e-mail: [theomora@dima.unige.it](mailto:theomora@dima.unige.it)

4.  $L_1 \subset L_2 \implies \mathfrak{P}(L_1) \supset \mathfrak{P}(L_2)$ ;
5.  $\mathfrak{L}(P_1 \cap P_2) \supset \mathfrak{L}(P_1) + \mathfrak{L}(P_2)$ ;
6.  $\mathfrak{P}(L_1 \cap L_2) \supset \mathfrak{P}(L_1) + \mathfrak{P}(L_2)$ ;
7.  $\mathfrak{L}(P_1 + P_2) = \mathfrak{L}(P_1) \cap \mathfrak{L}(P_2)$ ;
8.  $\mathfrak{P}(L_1 + L_2) = \mathfrak{P}(L_1) \cap \mathfrak{P}(L_2)$ ;
9.  $P = \mathfrak{P}\mathfrak{L}(P)$ ;
10.  $L \subset \mathfrak{L}\mathfrak{P}(L)$ ;
11.  $\dim_{\mathbb{F}}(L) < \infty \implies L = \mathfrak{L}\mathfrak{P}(L)$ .

*Id est*  $\mathfrak{P}$  and  $\mathfrak{L}$  define a duality between finite dimensional  $\mathcal{P}$ -modules of functionals and zero-dimensional ideals.

## 2 Möller's Algorithm

Let  $\mathbb{L} = \{\ell_1, \dots, \ell_s\} \subset \mathcal{P}^*$  be a (not necessarily linearly independent) set of  $\mathbb{F}$ -linear functionals such that  $L := \text{Span}_{\mathbb{F}}(\mathbb{L})$  is a  $\mathcal{P}$ -module, and let us denote, for each  $f \in \mathcal{P}$ ,  $v(f, \mathbb{L}) := (\ell_1(f), \dots, \ell_s(f)) \in \mathbb{F}^s$ . Since  $\dim_{\mathbb{F}}(L) < \infty$  then  $\mathfrak{l} := \mathfrak{P}(L)$  is a zero-dimensional ideal and

$$\#(\mathbf{N}(\mathfrak{l})) = \deg(\mathfrak{l}) = \dim_{\mathbb{F}}(L) =: r \leq s;$$

therefore, denoting

$$\mathbf{N}(\mathfrak{l}) = \{t_1, \dots, t_r\}, \quad 1 = t_1 < \dots < t_i < t_{i+1} < \dots < t_r,$$

we can consider the  $s \times r$  matrix  $\ell_i(t_j)$  whose columns are the vectors  $v(t_j, \mathbb{L})$  and are linearly independent, since any relation  $\sum_j c_j v(t_j, \mathbb{L}) = 0$  would imply

$$\ell_i \left( \sum_j c_j t_j \right) = \sum_j c_j \ell_i(t_j) = 0 \quad \text{and} \quad \sum_j c_j t_j \in \mathfrak{P}(L) = \mathfrak{l}$$

contradicting the definition of  $\mathbf{N}(\mathfrak{l})$ .

The matrix  $\ell_i(t_j)$  has rank  $r \leq s$  and it is possible to extract an ordered subset  $\Lambda := \{\lambda_1, \dots, \lambda_r\} \subset \mathbb{L}$ , satisfying  $\text{Span}_{\mathbb{F}}\{\Lambda\} = \text{Span}_{\mathbb{F}}\{\mathbb{L}\}$  and to renumber the terms in  $\mathbf{N}(\mathfrak{l})$  in such a way that each principal minor  $\lambda_i(t_j)$ ,  $1 \leq i, j \leq \sigma \leq r$  is invertible. Therefore, if we consider a set

$$\mathbf{q} := \{q_1, \dots, q_r\} \subset \mathcal{P}$$

which is triangular w.r.t.  $\Lambda$ , and  $(a_{ij})$  denotes the invertible matrix such that  $q_i = \sum_{j=1}^r a_{ij} t_j$ ,  $\forall i \leq r$ , then for each  $\sigma \leq r$

- $\{q_1, \dots, q_\sigma\}$  and  $\{\lambda_1, \dots, \lambda_\sigma\}$  are triangular;
- $\text{Span}_{\mathbb{F}}\{t_1, \dots, t_\sigma\} = \text{Span}_{\mathbb{F}}\{q_1, \dots, q_\sigma\}$ ;
- $(a_{ij})$  is lower triangular.

If we now further assume that

1.  $\dim_{\mathbb{F}}(L) = r = s$  and
2. each subvectorspace  $L_\sigma := \text{Span}_{\mathbb{F}}(\{\ell_1, \dots, \ell_\sigma\})$  is a  $\mathcal{P}$ -module

so that each  $\mathfrak{l}_\sigma = \mathfrak{P}(L_\sigma)$  is a zero-dimensional ideal and there is a chain

$$\mathfrak{l}_1 \supset \mathfrak{l}_2 \supset \cdots \supset \mathfrak{l}_s = \mathfrak{l},$$

then we have, for each  $\sigma$

- $\lambda_\sigma = \ell_\sigma$ ,
- $\mathbf{N}(\mathfrak{l}_\sigma) = \{t_1, \dots, t_\sigma\}$  is an order ideal,
- $\mathfrak{l}_\sigma \oplus \text{Span}_{\mathbb{F}}\{q_1, \dots, q_\sigma\} = \mathcal{P}$ ,
- $\mathbf{T}(q_\sigma) = t_\sigma$ .

In conclusion we can prove the following theorems.

**Theorem 2** (Möller) *Let  $\mathcal{P} := \mathbb{F}[x_1, \dots, x_n]$ , and  $<$  be any term-ordering. Let  $\mathbb{L} = \{\ell_1, \dots, \ell_s\} \subset \mathcal{P}^*$  be a set of  $\mathbb{F}$ -linear functionals such that  $\mathfrak{P}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$  is a zero-dimensional ideal.*

*Then there are*

- an integer  $r \in \mathbb{N}$ ,
- an order ideal  $\mathbf{N} := \{t_1, \dots, t_r\} \subset \mathcal{T}$ ,
- an ordered subset  $\Lambda := \{\lambda_1, \dots, \lambda_r\} \subset \mathbb{L}$ ,
- an ordered set  $\mathbf{q} := \{q_1, \dots, q_r\} \subset \mathcal{P}$ ,

*such that, denoting  $L := \text{Span}_{\mathbb{F}}(\mathbb{L})$  and  $\mathfrak{l} := \mathfrak{P}(L)$ , it holds:*

1.  $r = \deg(\mathfrak{l}) = \dim_{\mathbb{F}}(\mathbb{L})$ ,
2.  $\mathbf{N}(\mathfrak{l}) = \mathbf{N}$ ,
3.  $\text{Span}_{\mathbb{F}}(\Lambda) = \text{Span}_{\mathbb{F}}(\mathbb{L})$ ,
4.  $\text{Span}_{\mathbb{F}}\{t_1, \dots, t_\sigma\} = \text{Span}_{\mathbb{F}}\{q_1, \dots, q_\sigma\}, \forall \sigma \leq r$ ,
5.  $\{q_1, \dots, q_\sigma\}, \{\lambda_1, \dots, \lambda_\sigma\}$  are triangular,  $\forall \sigma \leq r$ .

*If, moreover, we have*

- $\dim_{\mathbb{F}}(L) = r = s$  and
- $L_\sigma := \text{Span}_{\mathbb{F}}(\{\ell_1, \dots, \ell_\sigma\})$  is a  $\mathcal{P}$ -module,  $\forall \sigma$ ,

*then it further holds*

6.  $\lambda_\sigma = \ell_\sigma$ ,
7.  $\mathbf{N}(\mathfrak{l}_\sigma) = \{t_1, \dots, t_\sigma\}$  is an order ideal,
8.  $\mathfrak{l}_\sigma \oplus \text{Span}_{\mathbb{F}}\{q_1, \dots, q_\sigma\} = \mathcal{P}$ ,
9.  $\mathbf{T}(q_\sigma) = t_\sigma$

*for each  $\sigma \leq r$ , where  $\mathfrak{l}_\sigma = \mathfrak{P}(L_\sigma)$ .*

**Corollary 3** (Lagrange Interpolation Formula) *Let  $\mathcal{P} := \mathbb{F}[x_1, \dots, x_n]$ ,  $<$  be any term-ordering.  $\mathbb{L} = \{\ell_1, \dots, \ell_s\} \subset \mathcal{P}^*$  be a set of linearly independent  $\mathbb{F}$ -linear functionals such that  $\mathfrak{l} := \mathfrak{P}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$  is a 0-dim. ideal.*

There exists a set  $\mathbf{q} = \{q_1, \dots, q_s\} \subset \mathcal{P}$  such that

1.  $q_i = \text{Can}(q_i, \mathbf{l}) \in \text{Span}_{\mathbb{F}}(\mathbf{N}(\mathbf{l}))$ ;
2.  $\mathbb{L}$  and  $\mathbf{q}$  are triangular;
3.  $\mathcal{P}/\mathbf{l} \cong \text{Span}_{\mathbb{F}}(\mathbf{q})$ .

There exists a set  $\mathbf{q}' = \{q'_1, \dots, q'_s\} \subset \mathcal{P}$  such that

1.  $q'_i = \text{Can}(q'_i, \mathbf{l}) \in \text{Span}_{\mathbb{F}}(\mathbf{N}(\mathbf{l}))$ ;
2.  $\mathbb{L}$  and  $\mathbf{q}'$  are biorthogonal;
3.  $\mathcal{P}/\mathbf{l} \cong \text{Span}_{\mathbb{F}}(\mathbf{q}')$ .

Let  $c_1, \dots, c_s \in \mathbb{F}$  and let  $q := \sum_i c_i q'_i \in \mathcal{P}$ . Then, if  $\{g_1, \dots, g_t\}$  denotes a Gröbner basis of  $\mathbf{l}$ , one has

1.  $q$  is the unique polynomial in  $\text{Span}_{\mathbb{F}}(\mathbf{N}(\mathbf{l}))$  such that  $\ell_i(q) = c_i$ , for each  $i$ ;
2. for each  $p \in \mathcal{P}$  the following statements are equivalent:
  - (a)  $\ell_i(p) = c_i$ , for each  $i$ ,
  - (b)  $q = \text{Can}(p, \mathbf{l})$ ,
  - (c) exist  $h_j \in \mathcal{P}$  such that

$$p = q + \sum_{j=1}^t h_j g_j, \mathbf{T}(h_j) \mathbf{T}(g_j) \leq \mathbf{T}(p - q).$$

Möller's Algorithm (Möller and Buchberger 1982; Faugère et al. 1993; Mariari and Möller 1993; Alonso and Marinari 2003) is a procedure which, given a set of  $\mathbb{F}$ -linear functionals  $\mathbb{L} = \{\ell_1, \dots, \ell_s\} \subset \mathcal{P}^*$  such that  $\mathfrak{P}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$  is a zero-dimensional ideal, allows to compute the data whose existence is stated in Theorem 2. The stronger version of the algorithm (Fig. 1), which assumes that for each  $\sigma \leq s$   $L_\sigma := \text{Span}_{\mathbb{F}}(\{\ell_1, \dots, \ell_\sigma\})$  is a  $\mathcal{P}$ -module, is performed by induction on  $\sigma$  and gives the complete structure of each ideal  $\mathbf{l}_\sigma = \mathfrak{P}(L_\sigma)$ .

Its correctness is based on the following

**Lemma 4** Let  $\mathcal{P} := \mathbb{F}[x_1, \dots, x_n]$ ,  $<$  be any term-ordering;  $\mathbb{L} = \{\ell_1, \dots, \ell_r\} \subset \mathcal{P}^*$  be a set of linearly independent  $\mathbb{F}$ -linear functionals such that  $\mathbf{l} := \mathfrak{P}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$  is a zero-dimensional ideal and let

$$\mathbf{N} := \{t_1, \dots, t_r\} \subset \mathcal{T},$$

$$\mathbf{q} := \{q_1, \dots, q_r\} \subset \mathcal{P},$$

$$G := \{g_1, \dots, g_t\} \subset \mathcal{P},$$

be such that

- $\mathbf{N}$  is an order ideal,
- $\text{Span}_{\mathbb{F}}\{t_1, \dots, t_r\} = \text{Span}_{\mathbb{F}}\{q_1, \dots, q_r\}$ ,
- $\{q_1, \dots, q_r\}$  and  $\{\ell_1, \dots, \ell_r\}$  are triangular,
- $\ell(g) = 0$  for each  $g \in G$  and each  $\ell \in \mathbb{L}$ ,

---

$(G_1, \dots, G_s, \mathbf{N}, \mathbf{q}) := \mathbf{G\text{-basis}}(\mathbb{L}, <)$   
**where**

$\mathbb{L} = \{\ell_1, \dots, \ell_s\} \subset \mathcal{P}^*$  is s.t.

$$L_\sigma := \text{Span}_{\mathbb{F}}(\{\ell_1, \dots, \ell_\sigma\})$$

is a  $\mathcal{P}$ -module, for each  $\sigma \leq s$ ,

$\mathbf{l}_\sigma = \mathfrak{P}(L_\sigma)$ , for each  $\sigma \leq s$ ,

$G_\sigma \subset \mathbf{l}_\sigma$  is the reduced Gröbner basis of  $\mathbf{l}_\sigma$ ,  $\forall \sigma \leq s$ ,

$\mathbf{N} := \{t_1, \dots, t_s\}$  is an order ideal,

$\mathbf{q} := \{q_1, \dots, q_s\} \subset \mathcal{P}$  is a set triangular to  $\mathbb{L}$ ,

$\mathbf{N}_\sigma := \{t_1, \dots, t_\sigma\} = \mathbf{N}(\mathbf{l}_\sigma)$ ,  $\forall \sigma \leq s$ ,

$q_\sigma \in \text{Span}_{\mathbb{F}}\{\mathbf{N}_\sigma\}$ , and  $\mathbf{T}(q_\sigma) = t_\sigma$ ,  $\forall \sigma \leq s$ ,

$\text{Span}_{\mathbb{F}}\{t_1, \dots, t_\sigma\} = \text{Span}_{\mathbb{F}}\{q_1, \dots, q_\sigma\}$ ,  $\forall \sigma \leq s$ ,

$\{q_1, \dots, q_\sigma\}$  and  $\{\ell_1, \dots, \ell_\sigma\}$  are triangular  $\forall \sigma$ .

$\sigma := 1, t_1 := 1, \mathbf{N} := \{t_1\}, q_1 := \ell_1(1)^{-1}t_1$ ,

$\mathbf{q} := \{q_1\}, G_1 := \{x_h - \ell_1(x_h), 1 \leq h \leq n\}$ ,

$\% \mathbf{N}_\sigma \sqcup \mathbf{T}(G_\sigma) = \mathcal{T}$ .

$\% \ell_j(f) = 0$  for all  $f \in G_\sigma, 1 \leq j \leq \sigma$ .

**For**  $\sigma := 2..s$  **do**

- $t := \min\{\mathbf{T}(f) : f \in G_\sigma, \ell_\sigma(f) \neq 0\}$ ,
- Let**  $\mathbf{f} \in G_\sigma : \mathbf{T}(\mathbf{f}) = t$ ,
- $t_\sigma := t, q_\sigma := \ell_\sigma(\mathbf{f})^{-1}\mathbf{f}, \mathbf{N} := \mathbf{N} \cup \{t_\sigma\}$ ,
- $\mathbf{q} := \mathbf{q} \cup \{q_\sigma\}$ ,
- \*  $G_\sigma := \{f - \ell_\sigma(f)q_\sigma : f \in G_{\sigma-1}\}$ .
- For each**  $h = 1..n : x_h t \notin \mathbf{T}(G_\sigma)$  **do**
- $p := x_h t$ ,
- \* **For**  $i = 1..\sigma$  **do**  $p := p - \ell_i(p)q_i$ ,
- $G_\sigma := G_\sigma \cup \{p\}$ ;
- $\% \mathbf{N}_\sigma \sqcup \mathbf{T}(G_\sigma) = \mathcal{T}$ ,
- $\% \ell_j(f) = 0$  for all  $f \in G_\sigma, 1 \leq j \leq \sigma$ .

---

**Fig. 1** Möller's Algorithm (1)

- $\mathbf{N} \sqcup \mathbf{T}_{<}(G) = \mathcal{T}$ ,
- for each  $g \in G, g - \text{lc}(g)\mathbf{T}_{<}(g) \in \text{Span}_{\mathbb{F}}(\mathbf{N})$ ,

then  $G$  is a reduced Gröbner basis of  $\mathfrak{P}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$  w.r.t.  $<$ .

The assumption that for each  $\sigma \leq s$ ,  $L_\sigma := \text{Span}_{\mathbb{F}}(\{\ell_1, \dots, \ell_\sigma\})$  can be satisfied if for instance the 0-dimensional ideal  $\mathbf{l} = \mathfrak{P}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$  is described in terms of a *Macaulay representation* (cf. Alonso and Marinari 2006), but often<sup>1</sup> it is not satisfied, thus requiring an alternative version (Fig. 2) performed by induction on the terms and not on the functionals and which returns also a basis of  $\text{Span}_{\mathbb{F}}(\mathbb{L})$ .

---

<sup>1</sup>Mainly in the solution of the FGLM Problem, where in any case the functionals are properly reordered so they satisfy such property.

---

 $(G, r, \mathbf{N}, \Lambda, \mathbf{q}) := \mathbf{G\text{-basis}}(\mathbb{L}, <)$ 
**where**
 $\mathbb{L} = \{\ell_1, \dots, \ell_s\} \subset \mathcal{P}^*$  is s.t.  $\mathfrak{l} := \mathfrak{P}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$  is a zero-dimensional ideal;
 $G \subset \mathfrak{l}$  is the reduced Gröbner basis of  $\mathfrak{l}$  w.r.t.  $<$ ;
 $r = \deg(\mathfrak{l}) = \dim_{\mathbb{F}}(\text{Span}_{\mathbb{F}}(\mathbb{L}))$ ;
 $\mathbf{N} := \{t_1, \dots, t_r\} = \mathbf{N}(\mathfrak{l})$ ;
 $1 = t_1 < t_2 < \dots < t_i < t_{i+1} < \dots < t_r$ ,
 $\Lambda := \{\lambda_1, \dots, \lambda_r\} \subset \mathbb{L}$ , is a linearly independent basis of  $\text{Span}_{\mathbb{F}}(\mathbb{L})$ ;
 $\mathbf{q} := \{q_1, \dots, q_r\} \subset \mathcal{P}$  is a set triangular to  $\Lambda$ ;
 $q_i \in \text{Span}_{\mathbb{F}}\{t_1, \dots, t_i\}, \mathbf{T}(q_i) = t_i$ , for each  $i \leq r$ ;
 $\text{Span}_{\mathbb{F}}\{t_1, \dots, t_i\} = \text{Span}_{\mathbb{F}}\{q_1, \dots, q_i\}$ , for each  $i \leq r$ ;
 $\{q_1, \dots, q_i\}$  and  $\{\lambda_1, \dots, \lambda_i\}$  are triangular, for each  $i \leq r$ .
 $G := \emptyset, r := 1, t_1 := 1, \mathbf{N} := \{t_1\}$ ,
 $v := (\ell_1(t_1), \dots, \ell_s(t_1))$ ,
 $\mu := \min\{j : \ell_j(1) \neq 0\}$ ,
 $\lambda_1 := \ell_\mu, \Lambda := \{\lambda_1\}$ ,
 $q_1 := \lambda_1(1)^{-1}t_1, \mathbf{q} := \{q_1\}, \text{vect}(1) := \lambda_1(1)^{-1}v$ ,
 $\% \% \text{vect}(1) = (\ell_1(q_1), \dots, \ell_s(q_1))$ ,
**While**  $\mathbf{N} \sqcup \mathbf{T}(G) \neq \mathcal{T}$  **do**
 $t := \min_{<} \{\tau \in \mathcal{T}, \tau \notin \mathbf{N} \sqcup \mathbf{T}(G)\}$ ,
 $q := t, v := (\ell_1(q), \dots, \ell_s(q))$ 
**For**  $j = 1..r$  **do**
 $v := v - \lambda_j(q) \text{vect}(j), q := q - \lambda_j(q)q_j$ ,
 $\% \% v = (\ell_1(q), \dots, \ell_s(q))$ .
**If**  $v = 0$  **then**
 $G := G \cup \{q\}$ ,
**else**
 $r := r + 1$ 
 $t_r := t, \mathbf{N} := \mathbf{N} \cup \{t_r\}$ ,
 $\mu := \min\{j : \ell_j(q) \neq 0\}$ ,
 $\lambda_r := \ell_\mu, \Lambda := \Lambda \cup \{\lambda_r\}$ ,
 $q_r := \lambda_r(q)^{-1}q, \mathbf{q} := \mathbf{q} \cup \{q_r\}, \text{vect}(r) := \lambda_r(q)^{-1}v$ 
 $\% \% \text{vect}(i) = (\ell_1(q_i), \dots, \ell_s(q_i))$  for each  $i, 1 \leq i \leq r$ 
 $G, r, \mathbf{N}, \Lambda, \mathbf{q}$ 

**Fig. 2** Möller's Algorithm (2)

*Remark 5* If, in the algorithm of Fig. 1, we define  $p$  in instruction  $\diamond$  as  $p := x_h \mathbf{f}$  instead of  $p := x_h t$ , we have two counterbalancing effects:

- the final output, while still a Gröbner basis, is not, in principle, reduced;
- since  $\mathbf{f} \in \mathfrak{l}_\sigma$ , we have  $x_h \mathbf{f} \in \mathfrak{l}_\sigma$  and  $\ell_i(p) = 0$  for each  $i \leq \sigma$  so that one can perform the instruction  $*$  for the single value  $i := \sigma$ .

Equivalently, defining, in the algorithm of Fig. 1,  $p$  in instruction  $\diamond$  as

$$p := x_h \mathbf{f} - \ell_\sigma(x_h \mathbf{f})q_\sigma = (x_h - \ell_\sigma(x_h \mathbf{f})\ell_\sigma(\mathbf{f})^{-1})\mathbf{f} \quad (1)$$

we can simply remove the instruction  $*$ .

Finally note that the algorithm discussed in Guerrini and Rimoldi (2009) is the generalization to modules of the version of the algorithm of Fig. 1 where, in instruction  $\diamond$ ,  $p$  is defined as in (1) and the instructions  $*$  and  $\bullet$  are removed.

### 3 The FGLM Problem

For its elimination property, the *lex* ordering is a good tool for solving [Gianni–Kalkbrenner Algorithm (Gianni 1989b; Kalkbrenner 1989; Mora and Orsini 2009), Lazard’s triangular sets (Lazard 1991, 1992; Aubry et al. 1999; Aubry and Moreno Maza 1999)] or for applications [see the CRHT-like algorithms in BCH codes (Mora and Orsini 2009)] but both practical experience and theoretical argument show that, in general, *lex* is a very bad choice for applying the Buchberger (1965, 2006) Algorithm. On the other side the *degrevlex ordering* is the *optimal* choice for applying it (Bayer and Stillman 1987). This suggests (Faugère et al. 1993) the

**Problem 6** (FGLM Problem) *Given*

- a term-ordering  $<$  on the polynomial ring  $\mathcal{P} := \mathbb{F}[x_1, \dots, x_n]$ ,
- a zero-dimensional ideal  $\mathfrak{l} \subset \mathcal{P}$  and
- its reduced Gröbner basis  $G_<$  w.r.t. the term-ordering  $<$ ,

to deduce the Gröbner basis  $G_<$  of  $\mathfrak{l}$  w.r.t.  $<$ .

### 4 The FGLM Matrix

Let  $<$  be a term-ordering and  $\mathbf{N}_<(\mathfrak{l}) = \{\tau_1, \dots, \tau_s\}$ ; in order to apply Möller’s Algorithm to the FGLM Problem, we just need to choose as functionals  $\mathbb{L} := \{\ell_1, \dots, \ell_s\}$  the coefficients of the canonical forms  $\ell_i(\cdot) := \gamma(\cdot, \tau_i, \mathbf{N}_<(\mathfrak{l}))$  so that we need to compute

$$\mathbf{Rep}(f, \mathbf{N}_<(\mathfrak{l})) := (\gamma(f, \tau_1, \mathbf{N}_<(\mathfrak{l})), \dots, \gamma(f, \tau_s, \mathbf{N}_<(\mathfrak{l})))$$

for each  $f \in \mathbb{B} := \{x_i \tau_j, 1 \leq i \leq n, 1 \leq j \leq s\}$ .

The key idea of FGLM is to treat such elements by  $<$ -increasing ordering, so that, when the loop is treating a term  $x_h \tau_l$ , we have previously managed the term  $\tau_l$  and thus previously computed  $\mathbf{Rep}(\tau_l, \mathbf{N}_<(\mathfrak{l}))$  which satisfies the relation

$$\tau_l - \sum_{j=1}^s \gamma(\tau_l, \tau_j, \mathbf{N}_<(\mathfrak{l})) \tau_j = \tau_l - \text{Can}(\tau_l, \mathfrak{l}, <) \in \mathfrak{l},$$

so that  $x_h \tau_l - \sum_{j=1}^s \gamma(\tau_l, \tau_j, \mathbf{N}_<(\mathfrak{l})) x_h \tau_j \in \mathfrak{l}$ , and

$$\begin{aligned} \text{Can}(x_h \tau_l, \mathbf{l}, \prec) &= \sum_{j=1}^s \gamma(\tau_l, \tau_j, \mathbf{N}_\prec(\mathbf{l})) \text{Can}(x_h \tau_j, \mathbf{l}, \prec) \\ &= \sum_{i=1}^s \left( \sum_{j=1}^s \gamma(\tau_l, \tau_j, \mathbf{N}_\prec(\mathbf{l})) \gamma(x_h \tau_j, \tau_i, \mathbf{N}_\prec(\mathbf{l})) \right) \tau_i. \end{aligned}$$

For the  $\prec$ -minimal  $\omega := x_h \tau_l \in \mathbf{B}$  under consideration we have the following three cases:

- if  $\omega \notin \mathbf{T}_\prec(\mathbf{l})$  then  $\omega \in \mathbf{N}_\prec(\mathbf{l})$ , so that we add  $\omega$  to  $\mathbf{N}$  and  $\{\omega x_h : 1 \leq h \leq n\}$  to  $\mathbf{B}$ ;
- if there is  $g \in G_\prec$  such that

$$\mathbf{T}_\prec(g) = \omega \quad \text{and} \quad g = \omega - \sum_{\tau \in \mathbf{N}_\prec(\mathbf{l})} \gamma(\omega, \tau, \mathbf{N}_\prec(\mathbf{l})) \tau,$$

since the procedure iterates on  $\prec$ -increasing values of  $\omega$ , we have

$$\gamma(\omega, \tau, \mathbf{N}_\prec(\mathbf{l})) \neq 0 \implies \tau \prec \omega \implies \tau \in \mathbf{N};$$

- if there is  $H, 1 \leq H \leq n, \tau \in \mathbf{T}_\prec(\mathbf{l})$  such that  $\omega = x_H \tau$ ; thus  $\tau \prec \omega$  has been already treated so that we have obtained a representation

$$\text{Can}(\tau, \mathbf{l}, \prec) = \sum_{j=1}^s \gamma(\tau, \tau_j, \mathbf{N}_\prec(\mathbf{l})) \tau_j;$$

since in such representation we have

$$\gamma(\tau, \tau_j, \mathbf{N}_\prec(\mathbf{l})) \neq 0 \implies \tau_j \prec \tau \implies \tau_j \in \mathbf{N}, x_H \tau_j \prec x_H \tau = \omega = x_h \tau_l$$

and  $\tau = x_h \tau_l$  for  $\tau_l := \frac{\tau_j}{x_H}$ , we also have the representation

$$\text{Can}(x_H \tau, \mathbf{l}, \prec) = \sum_{j=1}^s \gamma(\tau, \tau_j, \mathbf{N}_\prec(\mathbf{l})) \text{Can}(x_H \tau_j, \mathbf{l}, \prec)$$

and we can use the same formula as above to derive

$$\begin{aligned} \gamma(x_h \tau_l, \tau_i, \mathbf{N}_\prec(\mathbf{l})) &= \gamma(x_H \tau, \tau_i, \mathbf{N}_\prec(\mathbf{l})) \\ &= \sum_{j=1}^s \gamma(\tau, \tau_j, \mathbf{N}_\prec(\mathbf{l})) \gamma(x_H \tau_j, \tau_i, \mathbf{N}_\prec(\mathbf{l})) \\ &= \sum_{j=1}^s \gamma(x_h \tau_l, \tau_j, \mathbf{N}_\prec(\mathbf{l})) \gamma(x_H \tau_j, \tau_i, \mathbf{N}_\prec(\mathbf{l})). \end{aligned}$$

These remarks can be formalized in the algorithm described in Fig. 3; Fig. 4 proposes the instantiation of Möller's Algorithm (Fig. 2) to the setting of the FGLM Problem.

---

$(\mathbf{N}_\prec, \mathcal{M}) := \text{FGLM-Matrix}(G_\prec)$

**where**

$G_\prec \subset \mathfrak{l}$  is the reduced Gröbner basis of  $\mathfrak{l}$  w.r.t.  $\prec$ ;

$s = \deg(\mathfrak{l})$ ,

$\mathbf{N}_\prec := \{\tau_1, \dots, \tau_s\} = \mathbf{N}_\prec(\mathfrak{l})$ ,

$\mathbf{1} = \tau_1 \prec \tau_2 \prec \dots \prec \tau_j \prec \tau_{j+1} \prec \dots \prec \tau_s$ ,

$\mathcal{M} = \mathcal{M}(\mathbf{N}_\prec) = \left\{ \left( a_{ij}^{(h)} \right) \in \mathbb{F}^{s^2}, 1 \leq h \leq n \right\}$  is the set of the square matrices defined by the equalities  $x_h \tau_l = \sum_j a_{lj}^{(h)} \tau_j$  in  $\mathcal{P}/\mathfrak{l} = \text{Span}_{\mathbb{F}}(\mathbf{N}_\prec)$ ;

$r := 1$ ,  $\tau_1 := 1$ ,  $\mathbf{N}_\prec := \{\tau_1\}$ ,  $\mathbf{B} := \{x_h : 1 \leq h \leq n\}$ ,

**While**  $\mathbf{B} \neq \emptyset$  **do**

$\omega := \min_\prec(\mathbf{B})$ ,  $\mathbf{B} := \mathbf{B} \setminus \{\omega\}$ ,

$h, l : \omega := x_h \tau_l$

**If**  $\omega \notin \mathbf{T}_\prec(\mathfrak{l})$  **then**

$r := r + 1$

$\tau_r := \omega$ ,  $\mathbf{N}_\prec := \mathbf{N}_\prec \cup \{\tau_r\}$ ,  $\mathbf{B} := \mathbf{B} \cup \{x_h \tau_r : 1 \leq h \leq n\}$ ,

$a_{lr}^{(k)} := 1$ ;

**else**

**if**  $\exists g := \mathbf{T}_\prec(g) - \sum_{j=1}^r \gamma(\omega, \tau_j, \mathbf{N}_\prec) \tau_j \in G_\prec : \mathbf{T}_\prec(g) = \omega = x_h \tau_l$  **then**

**For**  $j = 1..r$  **do**  $a_{lj}^{(h)} := \gamma(\omega, \tau_j, \mathbf{N}_\prec)$

**else**

**Let**  $H, \iota : 1 \leq H \leq n, 1 \leq \iota \leq r : x_h \tau_\iota \in \mathbf{T}_\prec(G_\prec)$ ,  $\tau_\iota = x_H \tau_\iota$ ;

**For**  $i = 1..r$  **do**  $a_{li}^{(h)} := \sum_{j=1}^r a_{lj}^{(h)} a_{ji}^{(H)}$

**For each**  $H, i : x_H \tau_i = \omega$  **do**

**For**  $j = 1..r$  **do**  $a_{ij}^{(H)} := a_{lj}^{(h)}$ ;

$\mathbf{N}_\prec, \mathcal{M}$

---

**Fig. 3** The FGLM Matrix

## 5 Pointers

Remark (Compare Guerrini and Rimoldi 2009) that the Berlekamp–Massey Algorithm can be interpreted as a sort of FGLM Algorithm on modules with functionals depending on the state of the computation.<sup>2</sup>

---

<sup>2</sup>in fact, with Berlekamp's (1968) notation we assume to have found the basis  $\{(\sigma^{(k)}, \omega^{(k)}), (\tau^{(k)}, \gamma^{(k)})\}$  of the module

$$M_k := \left\{ (a(z), b(z)) \in \mathbb{F}_2[z]^2 : (1+S)a(z) \equiv b(z) \pmod{z^{k+1}} \right\} \subset \mathbb{F}_2[z]^2$$

and we consider the new functional  $\lambda_{k+1} : \mathbb{F}_2[z]^2 \rightarrow \mathbb{F}_2$  defined by

$$\lambda_{k+1}(a(z), b(z)) := \Delta_1^{(k)}$$

where  $\Delta_1^{(k)} \in \mathbb{F}_2$  is the value for which

$$(1+S)a(z) - b(z) \equiv \Delta_1^{(k)} z^{k+1} \pmod{z^{k+2}}.$$

---

 $(G, \mathbf{N}, \mathbf{q}) := \mathbf{FGLM}(G_{\prec}, <)$ 
**where**

$\prec$  and  $\prec$  are term-orderings on  $\mathcal{P}$ ,  
 $I \subset \mathcal{P}$  is a zero-dimensional ideal,  
 $G_{\prec} \subset I$  is the reduced Gröbner basis of  $I$  w.r.t.  $\prec$ ;  
 $s = \deg(I)$ ,  
 $\mathbf{N}_{\prec} := \{\tau_1, \dots, \tau_s\} = \mathbf{N}_{\prec}(I)$ ,  
 $1 = \tau_1 \prec \tau_2 \prec \dots \prec \tau_j \prec \tau_{j+1} \prec \dots \prec \tau_s$ ,  
 $\mathcal{M} = \mathcal{M}(\mathbf{N}_{\prec}) = \left\{ \left( a_{ij}^{(h)} \right) \in \mathbb{F}^{s^2}, 1 \leq h \leq n \right\}$  is the set of the square matrices defined by the equalities  $x_h \tau_i = \sum_j a_{ij}^{(h)} \tau_j$  in  $\mathcal{P}/I = \text{Span}_{\mathbb{F}}(\mathbf{N}_{\prec})$ ;  
 $G \subset I$  is the reduced Gröbner basis of  $I$  w.r.t.  $<$ ,  
 $\mathbf{N} := \{t_1, \dots, t_s\} = \mathbf{N}_{\prec}(I)$ ,  
 $1 = t_1 < t_2 < \dots < t_j < t_{j+1} < \dots < t_s$ ,  
 $\mu : \{1, \dots, s\} \mapsto \{1, \dots, s\}$  is a permutation,  
 $\mathbf{q} := \{q_1, \dots, q_s\} \subset \mathcal{P}$  is a set triangular to

$$\{\gamma(\cdot, \tau_{\mu(1)}, \mathbf{N}_{\prec}), \dots, \gamma(\cdot, \tau_{\mu(s)}, \mathbf{N}_{\prec})\}$$

$q_i \in \text{Span}_{\mathbb{F}}\{t_1, \dots, t_i\}$ ,  $\mathbf{T}_{\prec}(q_i) = t_i$ , for each  $i \leq s$ ,  
 $\{q_1, \dots, q_i\}$  and  $\{\gamma(\cdot, \tau_{\mu(1)}, \mathbf{N}_{\prec}), \dots, \gamma(\cdot, \tau_{\mu(i)}, \mathbf{N}_{\prec})\}$  are triangular for all  $i \leq s$ .

$(\mathbf{N}_{\prec}, \mathcal{M}) := \mathbf{FGLM-Matrix}(G_{\prec})$

$G := \emptyset, r := 1, t_1 := 1, \mathbf{N} := \{t_1\}, q_1 := 1, \mathbf{q} := \{q_1\}$ ,  
 $B := \{x_h, 1 \leq h \leq n\}$   
 $\text{vect}(1) := (1, 0, \dots, 0), \mu(1) := 1$ ,  
 $\% \text{ vect}(1) = \mathbf{Rep}(q_1, \mathbf{N}_{\prec}), \mu(1) = \min\{j : \gamma(q_1, \tau_j, \mathbf{N}_{\prec}) \neq 0\}$

**While**  $B \neq \emptyset$  **do**

$t := \min_{\prec}(B), B := B \setminus \{t\}$ ,

$l, h : t = x_h t_l = x_h \mathbf{T}_{\prec}(q_l)$

**If**  $t \notin \mathbf{T}_{\prec}(G)$  **then**

$q := x_h t_l$

**For**  $i = 1..s$  **do**  $v_i := \sum_{j=1}^s \gamma(q_l, \tau_j, \mathbf{N}_{\prec}) a_{ji}^{(h)}$ ;

$v := (v_1, \dots, v_s)$

$\% v = \mathbf{Rep}(q, \mathbf{N}_{\prec})$

**For**  $j = 1..r$  **do**

$v := v - \gamma(q, \tau_{\mu(j)}, \mathbf{N}_{\prec}) \text{vect}(j), q := q - \gamma(q, \tau_{\mu(j)}, \mathbf{N}_{\prec}) q_j$ ,

$\% v = \mathbf{Rep}(q, \mathbf{N}_{\prec})$

**If**  $v = 0$  **then**

$G := G \cup \{q\}$ ,

**else**

$r := r + 1$

$t_r := t, \mathbf{N} := \mathbf{N} \cup \{t_r\}$ ,

$\mu(r) := \min\{j : \gamma(q, \tau_j, \mathbf{N}_{\prec}) \neq 0\}$ ,

$q_r := \gamma(q, \tau_{\mu(r)}, \mathbf{N}_{\prec})^{-1} q, \text{vect}(r) := \gamma(q, \tau_{\mu(r)}, \mathbf{N}_{\prec})^{-1} v$

$\% \text{ vect}(i) = \mathbf{Rep}(q_i, \mathbf{N}_{\prec}), \forall i, 1 \leq i \leq r$

$\mathbf{q} := \mathbf{q} \cup \{q_r\}$ ,

$B := B \cup \{x_h t_r, 1 \leq h \leq n\}$ ,

 $G, \mathbf{N}, \mathbf{q}$ 


---

**Fig. 4** The FGLM Algorithm

---

In other words, we can consider the functionals  $\lambda_k : \mathbb{F}_2[z]^2 \rightarrow \mathbb{F}_2$ ,  $0 \leq k \leq 2t$  defined by  $\lambda_{k+1}(a(z), b(z)) := c_k$  where

$$\sum_k c_k z^k = (1 + S)a(z) - b(z) \in \mathbb{F}_2[[z]]$$

and each module  $M_k$  satisfies

$$M_k := \left\{ (a(z), b(z)) \in \mathbb{F}_2[z]^2 : \lambda_i(a(z), b(z)) = 0, 0 \leq i \leq k \right\} \subset \mathbb{F}_2[z]^2.$$

However, the earliest instance of the FGLM Algorithm goes back to 1936: in fact, the Todd–Coxeter Algorithm (Todd and Coxeter 1936) can be easily read (Reinert and Madlener 1998) as a re-formulation of **FGLM-Matrix** (Fig. 3) over groups viewed as quotients of a non-commutative polynomial rings modulo a binomial ideal.

The FGLM Problem was already solved essentially by means of the FGLM Algorithm in Buchberger (1970, 1998).

Möller's Algorithm was introduced for the first time in Möller and Buchberger (1982): in that setting the considered functionals were point evaluations, the aim being multivariate interpolation; the same procedure was proposed in Gianni (1989a) as a tool to efficiently perform a change of coordinate in a 0-dimensional ideal.

Faugère et al. (1993) introduced the FGLM Problem and solved it with the algorithm presented in Fig. 4; the paper gives also a precise complexity analysis and introduced both the FGLM Matrix and the efficient algorithm (Fig. 3) computing it.

Marinari and Möller (1993) reconsidered Möller's and the FGLM Algorithms, merging them and interpreting them in the setting of functionals; Alonso and Marinari (2003) is a survey which discusses also Macaulay's Algorithm to describe the structure of the canonical module  $\mathcal{L}(I)$ .

The FGLM Algorithm *proper* solves the FGLM Problem only for a 0-dim. ideal; Licciardi (1994) explains how to extend it to a multi-dimensional ideal; the corresponding algorithm is however far from being fast. The same weakness is shared by the Gröbner Walk Algorithm (Collard 1993).

The most efficient algorithm for the solution of the FGLM Problem, at least in the multidimensional case, is the Hilbert Driven Algorithm (Traverso 1996): assuming wlog that  $I$  is homogeneous, the knowledge of the basis  $G_<$  allows to compute the Hilbert function of  $I$  and thus, at each step, to predict how many new generators of a fixed degree are needed in the basis  $G_<$ ; when such generators are produced, all other S-pairs of same degree are discarded and the Hilbert function of the monomial ideal  $(T_<(g) : g \in G_<)$  is re-evaluated and the computation is performed in higher degree.

Recently new ideas have been proposed which, in my opinion, promise to be more efficient than the FGLM and the Hilbert Driven Algorithms (Basiri and Faugère 2003; Sala and Zanoni 2004).

Möller's Algorithm has been generalized to projective spaces (Abbott et al. 2000) and to non-commutative setting (Borges-Trenard et al. 2000).

Borges-Quintana et al. (2006a, 2006b, 2007) use an improved version of the FGLM algorithm for binomial ideals in order to correct binary linear codes (see Borges-Quintana et al. 2009).

---

For this interpretation I am strongly indebted to Fitzpatrick and Jennings (1998), Gianni and Trager (2002).

## 6 Point Evaluation

### 6.1 Möller's Algorithm

As we have already remarked the functionals considered in Möller and Buchberger (1982) were evaluations at a set of points

$$\mathbf{X} := \{\mathbf{a}_1, \dots, \mathbf{a}_s\} \subset \mathbb{F}^n, \quad \mathbf{a}_i := (a_{i1}, \dots, a_{in}),$$

id est  $\ell_i(p) := p(\mathbf{a}_i)$ ,  $1 \leq i \leq s$ , for each  $p \in \mathcal{P}$ . Using the notation of Corollary 3 and denoting

$$v(\mathbf{X}, p) := (p(\mathbf{a}_1), \dots, p(\mathbf{a}_s)) \in \mathbb{F}^s \quad \text{for each } p \in \mathcal{P},$$

we can say that the aim of Möller and Buchberger (1982) was to produce the *Newton interpolators*  $\mathbf{q}$  and, by further linear algebra,—which of course requires the explicit computation of the vectors  $\text{vect}(i) = v(\mathbf{X}, q_i)$ —the *Lagrange interpolators*  $\mathbf{q}'$ ; as a byproduct the Algorithm returns also some irrelevant (for Möller and Buchberger 1982) data, namely the Gröbner basis  $G$  and the Gröbner *escalier*  $\mathbf{N}$  of the *defining ideal*  $\mathbf{l}(\mathbf{X}) := \mathfrak{P}(\mathbb{L})$  of  $\mathbf{X}$ , which, on the other side, are the data whose computation is the aim of Faugère et al. (1993).

In merging the two algorithms, Marinari and Möller (1993) proposed four variations of the algorithm: Alg. 1 (p. 115) which iterates on the terms and Alg. 2 (p. 117) which iterates on the functionals are essentially respectively Figs. 2 and 1; Alg. 1v and Alg. 2v (p. 127) are an adaptation aimed to take, in the case of point evaluation, explicitly advantage of the fact that each  $\mathbf{l}_\sigma$  is an ideal so that

$$p \in \mathbf{l}_\sigma \implies \ell_i(x_h p) = 0 \quad \text{for each } i, h, 1 \leq h \leq n, 1 \leq i \leq \sigma;$$

in particular, Alg. 1v is obtained from Fig. 1 by defining  $p$  as  $p := x_h \mathbf{f}$  in the instruction  $\diamond$ , as in Remark 5, and substituting the instruction  $*$  with  $p := \text{Can}(p, \mathbf{l}) - a_{\sigma h} \mathbf{f}$ .

### 6.2 Cerlienco–Mureddu Correspondence

In the case in which  $<$  is lex, Cerlienco and Mureddu (1990, 1995, 2002) proposed an efficient combinatorial algorithm which to each *ordered* finite set of points  $\mathbf{X}$  associates an order ideal  $\mathbf{N}(\mathbf{X})$  and a bijection, the *Cerlienco–Mureddu Correspondence*,  $\Phi(\mathbf{X}) : \mathbf{X} \mapsto \mathbf{N}(\mathbf{X})$  satisfying

**Theorem 7** (Cerlienco and Mureddu 1990)  $\mathbf{N}_{<}(\mathbf{X}) = \mathbf{N}(\mathbf{l}(\mathbf{X}))$  holds for each finite set of points  $\mathbf{X} \subset \mathbb{F}^n$ .

*Remark 8* (Cerlienco and Mureddu 1990) Once, the set  $\mathbf{N}(I(X)) := \{t_1, \dots, t_s\}$  is obtained via the Cerlienco–Mureddu Algorithm and Theorem 7, one deduces

$$\mathbf{G}_{<}(I(X)) := \{\tau_1, \dots, \tau_r\}, \tau_1 < \tau_2 < \dots < \tau_r, \tau_i := X_1^{d_1^{(i)}} \cdots X_n^{d_n^{(i)}}$$

and can obtain the lex Gröbner basis of  $I(X)$  by interpolation: for each  $\tau_j \in \mathbf{G}(I(X))$  we have just to find the unknowns  $a_{ij} \in \mathbb{F}$  which satisfy the linear equalities  $v(X, \tau_j) = \sum_{i=1}^s a_{ij} v(X, t_i)$ .

The Cerlienco–Mureddu Algorithm is iterative, in the sense that its input consists of a set of points  $X$ , the related Cerlienco–Mureddu Correspondence  $\Phi(X) : X \mapsto \mathbf{N}(X)$  and a point  $b \in \mathbb{F}^n \setminus X$  and its output is a single term  $\tau \in \mathcal{T} \setminus \mathbf{N}(X)$  such that, setting  $Y := X \cup \{b\}$ , we have

$$\mathbf{N}(Y) := \mathbf{N}(X) \cup \{\tau\}, \quad \text{and} \quad \Phi(Y)(a) := \begin{cases} \Phi(X)(a) & a \in X, \\ \tau & a = b. \end{cases}$$

The Cerlienco–Mureddu Correspondence and Lazard's Structural Theorem (Lazard 1985) are merged in

**Theorem 9** (Marinari 2006) *With the present notation, there is a combinatorial algorithm which, given  $X$ , returns sets of points  $X_{m\delta i} \subset \mathbb{F}^m$ ,  $\forall m, \delta, i : 1 \leq i \leq r, 1 \leq m \leq n, 1 \leq \delta \leq d_m^{(i)}$ , thus allowing to compute, by means of the Cerlienco–Mureddu Algorithm the corresponding order ideal*

$$F_{m\delta i} := \mathbf{N}(X_{m\delta i}) \subset \mathcal{T} \cap k[X_1, \dots, X_{m-1}]$$

and, by interpolation, unique polynomials  $\gamma_{m\delta i} := X_m - \sum_{\omega \in F_{m\delta i}} c_{\omega} \omega$  such that, setting  $f_i := \prod_m \prod_{\delta} \gamma_{m\delta i}, \{f_1, \dots, f_r\}$  is a minimal (non reduced) Gröbner basis of  $I(X)$ .

An alternative combinatorial algorithm returning at least the set  $\mathbf{N}(X)$  satisfying Theorem 7, has been independently proposed in Gao et al. (1993) and Felszeghy et al. (2006); apparently, unlike the Cerlienco–Mureddu Algorithm, it is not iterative.

A recent (Lederer 2008) Cerlienco–Mureddu-like proposal, very similar to those of Gao et al. (1993) and Felszeghy et al. (2006), while still not iterative, suggests a clever interpolation formula which successfully strengthens the weak proposal of Remark 8.

### 6.3 Farr–Gao Analysis

In Farr and Gao (2000) the authors specialize the Fitzpatrick Algorithm (Guerrini and Rimoldi 2009) in the case of ideal and for point evaluation as functionals, thus

essentially returning (see Remark 5) Fig. 1, and propose to improve it by computing reduced Gröbner basis at any iteration, thus giving a nearly *verbatim* version of Alg. 2v in Marinari and Möller (1993); next they compare on a series of random points the performance (the algorithms were implemented in MAGMA version 2.8) of Fitzpatrick’s Algorithm a.k.a. Alg. 2 in Marinari and Möller (1993) (Fig. 1), their version of Alg. 2v (Marinari and Möller 1993) and “the algorithm of Marinari and Möller (1993)” —from the context it is clear that among the four algorithms of Marinari and Möller (1993), they are speaking of the one reported in Abbott et al. (2000) *id est* Alg. 1 (Fig. 2).

Notwithstanding that the chosen tests<sup>3</sup> are strongly biased, they are illuminating and give a good perspective both on the performance of Figs. 2 and 1 and on the structure of the Gröbner *escalier*  $\mathbf{N}(\mathbf{X})$  of the defining ideal of a set  $\mathbf{X}$  of simple points.

### Remark 10

- As I said, the experiment is biased against Fig. 2, which according to the original Möller’s proposal, explicitly computes *all* vectors  $\text{vect}(i)$  thus requiring  $s^2$  evaluation of a polynomial at a functional; remark that in the worst case Fig. 1 and Alg. 2v (Marinari and Möller 1993) require at most *half* of such evaluation: remark that for the *drl* case and  $n \geq 31$  in fact the timing of Fig. 2 is roughly the double of the one of Fig. 1; the worst behaviour for *lex* will be explained below.
- For *lex*, Cerlienco–Mureddu Correspondence implies
  - $\bullet$   $\mathbf{N}(\mathbf{l}) = \{X_1^i, 0 \leq i < \#\mathbf{X}\}$  if and only if for  $i, j, 1 \leq i, j \leq s, i \neq j \implies a_{i1} \neq a_{j1}$  and
  - $\bullet$   $X_h \in \mathbf{N}(\mathbf{l})$  iff there are  $i, j, 1 \leq i, j \leq s$  for which  $a_{il} = a_{jl}$  for each  $l < h$ .

**Table 1** Average running times for 250 random points from  $(\mathbb{F}_q)^r$  (100 experiments)

$q$	$r$	Figure 2		Alg. 2v (Marinari and Möller 1993)		Figure 1	
		<i>drl</i>	<i>lex</i>	<i>drl</i>	<i>lex</i>	<i>drl</i>	<i>lex</i>
2	10	11.56	4.37	9.38	3.18	13.08	24.45
2	15	39.85	9.28	42.49	19.18	41.32	61.37
2	20	110.08	13.72	152.06	44.18	106.99	93.52
11	3	9.60	5.21	4.25	1.08	3.83	2.40
31	3	11.20	5.64	5.10	0.988	4.57	1.44
101	3	11.57	5.53	5.31	0.747	4.75	0.833
1009	3	12.51	6.41	5.70	0.477	5.03	0.464

<sup>3</sup>Compare Table 1 (Table 1 of Farr and Gao 2000); the other tests (Table 2–3) are similar; Table 4–6 report the application of the algorithm to the same points after the reordering induced by Gao et al. (1993); Table 4 of Farr and Gao (2000) is reported here in Table 2.

**Table 2** Average running times for 250 random points (sorted) from  $(\mathbb{F}_q)^r$  (100 experiments)

q	r	Figure 2		Alg. 2v (Marinari and Möller 1993)		Figure 1	
		drl	lex	drl	lex	drl	lex
2	10	7.78	2.72	6.60	1.84	7.17	2.22
2	15	32.71	6.24	35.65	9.17	31.58	7.14
2	20	98.97	9.26	139.66	24.05	92.93	11.28
11	3	8.11	2.80	3.96	0.944	3.43	1.01
31	3	11.15	3.75	5.10	0.932	4.47	0.950
101	3	11.63	4.18	5.31	0.721	4.71	0.704
1009	3	12.52	6.11	5.70	0.469	5.02	0.462

Thus, for lex and random points, Fig. 1 and Alg. 2v (Marinari and Möller 1993) need to evaluate in general *one* polynomial for each functional; thus in this experiment they have to perform  $s$  evaluations against  $s^2$ ; in such an handicap setting, Fig. 2 performs quite fairly.

3. The better behaviour of their algorithm (Alg. 2v Marinari and Möller 1993) w.r.t. Fitzpatrick's (Fig. 1) is correctly justified by Farr and Gao remarking that Fitzpatrick's Algorithm computes polynomials whose length (i.e. the size of their support) may grow exponentially in the number of variables; as a consequence, most of the computing time in the Fitzpatrick Algorithm is taken up with dealing with dense polynomials, and most of the time in the Farr–Gao algorithm involves the reduction step; this is true but does not hold for the algorithms presented here and in Marinari and Möller (1993) where dense polynomials are *never* used since all polynomials in  $G$ , being a combination of polynomials in  $\mathbf{q}$  are always canonical forms. The advantage of the algorithms proposed here against Fitzpatrick's and the improvement suggested in Farr and Gao (2000) is that, since the Newton interpolators are necessarily stored, they can be next efficiently applied to maintain the data as combination of elements in the Gröbner *escalier*  $\mathbf{N}$ .

This can be easily seen in the data for  $q = 2$  where Fig. 2 competes fairly well with the other algorithms notwithstanding its handicap *provided that r is small relative for the number n of points. If r is larger, then the advantage swings to Fig. 2* (Farr and Gao 2000). This suggests that Fig. 1 performs better than the data reported here, since the criticisms moved by Farr and Gao (2000) to their version of Fitzpatrick's Algorithm does not apply to Fig. 1.

4. Other indirect consequences of the structural properties pointed by Cerlienco–Mureddu Correspondence are of course also the behaviour of the algorithms in relation with the ratio  $r/n$  as well as the other remark of Farr and Gao (2000): *The reduced Gröbner bases for the defining ideal of a random set of 500 points from  $(\mathbb{F}_2)^{10}$  under lex order usually contains around 100 polynomials, while the border basis typically contains over 200!*

Which means, that the algorithm of Farr and Gao (2000) has *really* need to compute canonical forms, while Möller gets them for free.

5. A direct consequence of Cerlienco–Mureddu Correspondence is also the other behaviour pointed by Farr and Gao (2000), namely that *if the field size is allowed to grow, then the running time for Fig. 1 and Alg. 2v* (Marinari and Möller 1993) *under lex order actually decreases. The reason is that the Gröbner basis polynomials actually become simpler.*
6. In connection with Cerlienco–Mureddu Correspondence remark that for *degrevlex*, if  $n^i < s < n^{i+1}$  for  $s$  random points there is high probability of having  $\{\tau \in \mathcal{T}, \deg(\tau) \leq i\} \subset \mathbf{N} \subset \{\tau \in \mathcal{T}, \deg(\tau) \leq i + 1\}$  thus allowing to properly adapt the remarks given for *lex* also for *degrevlex*.
7. Quite interesting are also the data of Table 2:
  - (a) For  $r = 3$ ,  $q > s$  and both ordering, the data of the two tables are really similar, due to the fact that, with good probability, the Gröbner *escalier* is nearly the same for each subset of  $s' < s$  points of  $\mathbf{X}$ .
  - (b) For  $q = 2$  and *lex* the significant speed up in Fig. 1 and Alg. 2v (Marinari and Möller 1993) is an indirect consequence of the point ordering given by Gao et al. (1993), Felszeghy et al. (2006) which gives a better chance of maximizing the value of  $t$  in instruction  $\diamond$  thus minimizing the computation of instruction  $\star$ .<sup>4</sup>

*Remark 11* It is probably necessary to justify why Faugère et al. (1993) defined  $p$  in instruction  $\diamond$  of Fig. 1 as  $p := x_h f$ , thus being followed by Fitzpatrick and Farr and Gao (2000), instead of  $p := x_h t$ : the point is that, since the algorithms return a Gröbner description of  $p$  in terms of  $\mathbf{q}$ , if it is applied to elements of  $\mathbf{q}$  instead of  $\mathbf{N}(I)$  one freely obtains the corresponding sparser Gröbner description.

Also, for points evaluations, even multiple, computing  $\ell(x_h f)$  does not require explicit evaluation but can be immediately deduced, at cost of 1 product and sum, from that of  $\ell(f)$  via Leibniz formula (compare Möller 1993).

## 6.4 Points with Multiplicities

Points with multiplicity can be described in terms of functionals by means of *Macaulay representations* introduced by Macaulay (1913, 1916) and studied in Möller (1993), Marinari and Möller (1996), Marinari (2003), Marinari (2006) which describe their properties and algorithms to deal with them; a recent survey is Alonso

---

<sup>4</sup>Preliminary hand computations suggest that Gao et al. (1993), Felszeghy et al. (2006) do not obtain such advantage on Cerlienco–Mureddu Correspondence in the context of Theorem 9.

Oddly, Farr and Gao (2000) does not compare, in the *lex* case, the versions of Möller’s Algorithm with the interpolation scheme suggested by Remark 8.

The theoretical arguments of Marinari (2003) seem to suggest that Cerlienco–Mureddu interpolation scheme is a potential competitor of Möller’s Algorithm, probably the more so if we interpolate with the procedure sketched in Lederer (2008).

and Marinari (2006). For projective points compare Cioffi (1999), Cioffi and Orecchia (2001), Abbott et al. (2000). For characteristic 0, efficient techniques for solving zero-dimensional ideal via the FGLM Matrix are discussed in Auzinger and Stetter (1988), Möller and Stetter (1995), Mourrain (2005).

## References

- J. Abbott, A. Bigatti, M. Kreuzer, and L. Robbiano, *Computing ideals of points*, J. Symbolic Comput. **30** (2000), no. 4, 341–356.
- M. E. Alonso and M. G. Marinari, *The big mother of all dualities: Möller algorithm*, Comm. Algebra **31** (2003), no. 2, 783–818.
- M. E. Alonso and M. G. Marinari, *The big mother of all dualities. II. Macaulay bases*, AAECC **17** (2006), no. 6, 409–451.
- P. Aubry and M. Moreno Maza, *Triangular sets for solving polynomial systems: a comparative implementation of four methods*, J. Symbolic Comput. **28** (1999), nos. 1–2, 125–154.
- P. Aubry, D. Lazard, and M. Moreno Maza, *On the theories of triangular sets*, J. Symbolic Comput. **28** (1999), nos. 1–2, 105–124.
- W. Auzinger and H. J. Stetter, *An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations*, Internat. Schriftenreihe Numer. Math. **86** (1988), 11–30.
- E. R. Berlekamp, *Algebraic coding theory*, McGraw–Hill, New York, 1968.
- A. Basiri and J. C. Faugère, *Changing the ordering of Gröbner bases with LLL: case of two variables*, Proc. of ISSAC 2003, ACM, New York, 2003, pp. 23–29.
- D. Bayer and M. Stillman, *A theorem on refining division orders by the reverse lexicographic order*, Duke Math. J. **55** (1987), no. 2, 321–328.
- M. Borges-Quintana, M. A. Borges-Trenard, P. Fitzpatrick, and E. Martínez-Moro, *Gröbner bases and combinatorics for binary codes*, to appear in AAECC, <http://arxiv.org/abs/math.CO/0509164>, 2006a.
- M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro, *A general framework for applying FGLM techniques to linear codes*, LNCS, vol. **3857**, Springer, Berlin, 2006b, pp. 76–86.
- M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro, *On a Gröbner bases structure associated to linear codes*, J. Discrete Math. Sci. Cryptogr. **10** (2007), no. 2, 151–191.
- M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro, *An application of Möller's algorithm to coding theory*, this volume, 2009, pp. 379–384.
- M. A. Borges-Trenard, M. Borges-Quintana, and T. Mora, *Computing Gröbner bases by FGLM techniques in a non-commutative setting*, J. Symbolic Comput. **30** (2000), no. 4, 429–449.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- L. Cerlienco and M. Mureddu, *L'interpolazione polinomiale in dimensione  $\geq 2$* , preprint, 1990.
- L. Cerlienco and M. Mureddu, *From algebraic sets to monomial linear bases by means of combinatorial algorithms*, Disc. Math. **139** (1995), nos. 1–3, 73–87.
- L. Cerlienco and M. Mureddu, *Multivariate interpolation and standard bases for Macaulay modules*, J. Algebra **251** (2002), no. 2, 686–726.

- F. Cioffi, *Minimally generating ideals of fat points in polynomial time using linear algebra*, Ricerche di Mat. **XLVII** (1999), 55–63.
- F. Cioffi and F. Orecchia, *Computation of minimal generators of ideals of fat points*, Proc. of ISSAC 2001, ACM, New York, 2001, pp. 72–76.
- S. Collard, D. Mall, and M. Kalkbrener, *The Gröbner walk*, preprint, 1993.
- J. B. Farr and S. Gao, *Computing Gröbner bases for vanishing ideals of finite sets of points*, Tech. report, preprint, 2005.
- J. C. Faugère, P. Gianni, D. Lazard, and T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, J. Symbolic Comput. **16** (1993), no. 4, 329–344.
- B. Felszeghy, B. Ráth, and L. Rónyai, *The lex game and some applications*, J. Symbolic Comput. **41** (2006), no. 6, 663–681.
- P. Fitzpatrick and S. M. Jennings, *Comparison of two algorithms for decoding alternant codes*, AAECC **9** (1998), no. 3, 211–220.
- S. Gao, V. M. Rodrigues, and J. Stroomer, *Gröbner basis structure of finite sets of points*, preprint, 1993.
- P. Gianni, *Algebraic solution of systems of polynomial equations using Gröbner bases*, Proc. of AAECC1987, LNCS, vol. **356**, Springer, Berlin, 1989a, pp. 247–257.
- P. Gianni, *Properties of Gröbner bases under specializations*, Proc. of EUROCAL1987, LNCS, vol. **378**, Springer, Berlin, 1989b, pp. 293–297.
- P. Gianni and B. Trager, *Incremental decoding*, Proc. of LMCS2002, vols. **02–60**, 2002, pp. 138–146.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick's approach to recent developments*, this volume, 2009, pp. 1–100.
- M. Kalkbrener, *Solving systems of algebraic equations by using Gröbner bases*, Proc. of EUROCAL1987, LNCS, vol. **378**, Springer, Berlin, 1989, pp. 282–292.
- D. Lazard, *Ideal bases and primary decomposition: case of two variables*, J. Symbolic Comput. **1** (1985), no. 3, 261–270.
- D. Lazard, *A new method for solving algebraic systems of positive dimension*, Discrete Appl. Math. **33** (1991), nos. 1–3, 147–160.
- D. Lazard, *Solving zero-dimensional algebraic systems*, J. Symbolic Comput. **13** (1992), no. 2, 117–131.
- M. Lederer, *The vanishing ideal of a finite set of closed points in affine space*, J. Pure Appl. Algebra **212** (2008), no. 5, 1116–1133.
- S. Licciardi, *Implicitization of hypersurfaces and curves by the Primbasissatz and basis conversion*, Proc. of ISSAC1994, ACM, New York, 1994, pp. 191–196.
- F. S. Macaulay, *On the resolution of a given modular system into primary systems including some properties of Hilbert numbers*, Math. Ann. **74** (1913), no. 1, 66–121.
- F. S. Macaulay, *The Algebraic Theory of Modular Systems*, Cambridge University Press, Cambridge, 1916.
- M. G. Marinari, *A remark on a remark by Macaulay or enhancing Lazard structural theorem*, Bull. Iranian Math. Soc. **29** (2003), no. 1, 1–45.
- M. G. Marinari, *Cerlienco–Mureddu correspondence and Lazard structural theorem*, Investigaciones Mat. **27** (2006), 155–178.
- M. G. Marinari and H. M. Möller, *Gröbner bases of ideals defined by functionals with an application to ideals of projective points*, AAECC **4** (1993), no. 2, 103–145.
- M. G. Marinari and H. M. Möller, *On multiplicities in polynomial system solving*, Trans. AMS **348** (1996), no. 8, 3283–3321.
- H. M. Möller, *Systems of algebraic equations solved by means of endomorphisms*, LNCS, vol. **673**, Springer, Berlin, 1993, pp. 43–56.
- H. M. Möller and B. Buchberger, *The construction of multivariate polynomials with preassigned zeros*, LNCS, vol. **144**, Springer, Berlin, 1982, pp. 24–31.
- H. M. Möller and H. J. Stetter, *Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems*, Numer. Math. **70** (1995), no. 3, 311–329.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.

- T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, this volume, 2009, pp. 69–91.
- B. Mourrain, *Bezoutian and quotient ring structure*, J. Symbolic Comput. **39** (2005), nos. 3–4, 397–415.
- B. Reinert and K. Madlener, *A note on Nielsen reduction and coset enumeration*, Proc. of ISSAC1998, ACM, New York, 1998, pp. 171–178.
- M. Sala and A. Zanoni, *personal communication*, 2004.
- J. A. Todd and H. S. M. Coxeter, *A practical method for enumerating cosets of a finite abstract group*, Proc. Edinb. Math. Soc., II. Ser. **5** (1936), 26–34.
- C. Traverso, *Hilbert functions and the Buchberger algorithm*, J. Symbolic Comput. **22** (1996), no. 4, 355–376.

# An Introduction to Linear and Cyclic Codes

Daniel Augot, Emanuele Betti and  
Emmanuela Orsini

**Abstract** Our purpose is to recall some basic aspects about linear and cyclic codes. We first briefly describe the role of error-correcting codes in communication. To do this we introduce, with examples, the concept of linear codes and their parameters, in particular the Hamming distance.

A fundamental subclass of linear codes is given by cyclic codes, that enjoy a very interesting algebraic structure. In fact, cyclic codes can be viewed as ideals in a residue classes ring of univariate polynomials. BCH codes are the most studied family of cyclic codes, for which some efficient decoding algorithms are known, as the method of Sugiyama.

## 1 An Overview on Error Correcting Codes

We give a brief description of a communication scheme, following the classical paper by Shannon (1948). Suppose that an *information source*  $A$  wants to say something to a *destination*  $B$ . In our scheme the information is sent through a *channel*. If, for example,  $A$  and  $B$  are mobile phones, then the channel is the space where electromagnetic waves propagate. The real experience suggests to consider the case in which some interference (*noise*) is present in the channel where the information passes through.

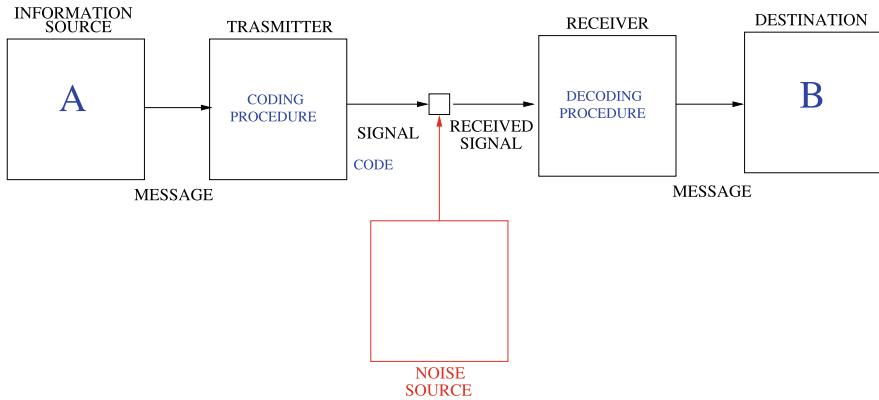
The basic idea of coding theory consists of adding some kind of redundancy to the message  $m$  that  $A$  wants to send to  $B$ . Following Fig. 1,  $A$  hands the message  $m$  to a device called a *transmitter* that uses a *coding procedure* to obtain a longer message  $m'$  that contains redundancy. The transmitter sends  $m'$  through the channel to another device called a *receiver*. Because of the noise in the channel, it may be that the message  $m''$  obtained after the transmission is different from  $m'$ . If the occurred errors are not too many (in a sense that will be clear later), the receiver is able to recover the original message  $m$ , using a *decoding procedure*.

---

D. Augot  
INRIA Paris-Rocquencourt, Paris, France  
e-mail: [Daniel.Augot@inria.fr](mailto:Daniel.Augot@inria.fr)

E. Betti  
Department of Mathematics, University of Florence, Florence, Italy  
e-mail: [betti@math.unifi.it](mailto:betti@math.unifi.it)

E. Orsini  
Department of Mathematics, University of Milan, Milan, Italy  
e-mail: [orsini@posso.dm.unipi.it](mailto:orsini@posso.dm.unipi.it)



**Fig. 1** A communication schema

To be more precise, the coding procedure is an injective map from the space of the admissible messages to a larger space. The code is the image of this map. A common assumption is that this map is a linear function between vector spaces. In the next section we will describe some basic concepts about coding theory using this restriction. The material of this tutorial can be found in Berlekamp (1968), Blahut (1983), Lin (1970), MacWilliams and Sloane (1977), Peterson and Weldon (1972), Pless (1982), Pless et al. (1998) and van Lint (1999).

## 2 Linear Codes

### 2.1 Basic Definitions

Linear codes are widely studied because of their algebraic structure, which makes them easier to describe than non-linear codes.

Let  $\mathbb{F}_q = GF(q)$  be the finite field with  $q$  elements and  $(\mathbb{F}_q)^n$  be the linear space of all  $n$ -tuples over  $\mathbb{F}_q$  (its elements are row vectors).

**Definition 1** Let  $k, n \in \mathbb{N}$  such that  $1 \leq k \leq n$ . A *linear code*  $C$  is a  $k$ -dimensional vector subspace of  $(\mathbb{F}_q)^n$ . We say that  $C$  is a linear code over  $\mathbb{F}_q$  with length  $n$  and dimension  $k$ . An element of  $C$  is called a *word* of  $C$ .

From now on we shorten “linear code on  $\mathbb{F}_q$  with length  $n$  and dimension  $k$ ” to “[ $n, k$ ] <sub>$q$</sub>  code”.

Denoting by “.” the usual scalar product, given a vector subspace  $S$  of  $(\mathbb{F}_q)^n$ , we can consider the dual space  $S^\perp$ .

**Definition 2** If  $C$  is an  $[n, k]_q$  code, its *dual code*  $C^\perp$  is the set of vectors orthogonal to all words of  $C$ :

$$C^\perp = \{c' \mid c' \cdot c = 0, \forall c \in C\}.$$

Thus  $C^\perp$  is an  $[n, n-k]_q$  code.

**Definition 3** If  $C$  is an  $[n, k]_q$  code, then any matrix  $G$  whose rows form a basis for  $C$  as a  $k$ -dimensional vector space is called a *generator matrix* for  $C$ . If  $G$  has the form  $G = [I_k \mid A]$ , where  $I_k$  is the  $k \times k$  identity matrix,  $G$  is called a generator matrix in *standard form*.

Thanks to this algebraic description, linear codes allow very easy encoding. Given a generator matrix  $G$ , the encoding procedure of a message  $m \in (\mathbb{F}_q)^k$  into the word  $c \in (\mathbb{F}_q)^n$  is just the matrix multiplication  $mG = c$ . When the generator matrix is in standard form  $[I_k \mid A]$ ,  $m$  is encoded in  $mG = (m, mA)$ . In this case the message  $m$  is formed by the first  $k$  components of the associated word. Such an encoding is called *systematic*.

We conclude this section with another simple characterization of linear codes.

**Definition 4** A *parity-check matrix* for an  $[n, k]_q$  code  $C$  is a generator matrix  $H \in \mathbb{F}_q^{(n-k) \times n}$  for  $C^\perp$ .

It is easy to see that  $C$  may be expressed as the null space of a parity-check matrix  $H$ :

$$\forall x \in (\mathbb{F}_q)^n, \quad Hx^T = 0 \quad \Leftrightarrow \quad x \in C.$$

## 2.2 Hamming Distance

To motivate the next definitions we describe what could happen during a transmission process.

*Example 1* We suppose that the space of messages is  $(\mathbb{F}_2)^2$ :

$$(0, 0) = v_1, \quad (0, 1) = v_2, \quad (1, 0) = v_3, \quad (1, 1) = v_4.$$

Let  $C$  be the  $[6, 2]_2$  code generated by

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

Then:

$$C = \{(0, 0, 0, 0, 0, 0), (1, 1, 1, 1, 1, 1), (0, 0, 0, 1, 1, 1), (1, 1, 1, 0, 0, 0)\}.$$

To send  $v_2 = (0, 1)$  we transmit the word  $v_2G = (0, 0, 0, 1, 1, 1)$ ; typically, during the transmission the message gets distorted by noise and the receiver has to perform some operations to obtain the transmitted word. Let  $w$  be the received vector. Several different situations could come up:

1.  $w = (0, 0, 0, 1, 1, 1)$ , then  $w \in C$ , so the receiver deduces correctly that no errors have occurred and no correction is needed. It concludes that the message was  $v_2$ .
2.  $w = (0, 0, 0, 1, 0, 1) \notin C$ , then the receiver concludes that some errors have occurred. In this case it may “correct” and “detect” the error as follows. It may suppose that the word transmitted was  $(0, 0, 0, 1, 1, 1)$ , since that is the word that differs in the least number of positions from the received word  $w$ .
3.  $w = (0, 0, 0, 1, 0, 0) \notin C$ . The receiver correctly reaches the conclusion that there were some errors during the transmission, but if it tries to correct as in the previous case, it concludes that the word “nearest” to  $w$  is  $(0, 0, 0, 0, 0, 0)$ . In this case it corrects in a wrong way.
4.  $w = (0, 0, 0, 0, 0, 0) \in C$ . The receiver deduces incorrectly that no errors have occurred.

From the previous example we understand that, when the decoder gets a received vector which is not a word, it has to find the word in  $C$  which has been sent by the encoder, i.e., among all words, it has to find the one which has the “highest probability” of being sent. To do this it needs a priori knowledge on the channel, more precisely it needs to know how the noise can modify the transmitted word.

**Definition 5** A  $q$ -ary symmetric channel (SC for short) is a channel with the following properties:

- (a) the component of a transmitted word (an element of  $\mathbb{F}_q$  that here we name generally “symbol”) can be changed by the noise only to another element of  $\mathbb{F}_q$ ;
- (b) the probability that a symbol becomes another one is the same for all pairs of symbols;
- (c) the probability that a symbol changes during the transmission does not depend on its position;
- (d) if the  $i$ -th component is changed, then this fact does not affect the probability of change for the  $j$ -th components, even if  $j$  is close to  $i$ .

To these channel properties it is usually added a *source property*:

- *all words are equally likely to be transmitted.*

The  $q$ -ary SC is a model that rarely can describe real channels. For example the assumption (d) is not reasonable in practice: if a symbol is corrupted during the transmission there is a high probability that some errors happened in the neighborhood. Despite this fact, the classical approach accepts the assumptions of the SC, since it permits a simpler construction of the theory. The ways of getting around the troubles generated by this “false” assumption in practice are different case by case and these are not investigated here. From now we will assume that the channel

is a SC, and that the probability that a symbol changes to another is less than the probability that it is uncorrupted by noise.

In our assumptions, by Example 1, it is quite evident that a simple criterion to construct “good” codes would be to try to separate as much as possible the words of code inside  $(\mathbb{F}_q)^n$ .

**Definition 6** The (*Hamming*) *distance*  $d_H(u, v)$  between two vectors  $u, v \in (\mathbb{F}_q)^n$  is the number of coordinates in which  $u$  and  $v$  differ.

**Definition 7** The (*Hamming*) *weight* of a vector  $u \in (\mathbb{F}_q)^n$  is the number  $\mathbf{w}(u)$  of its nonzero coordinates, i.e.  $\mathbf{w}(u) = d_H(u, 0)$ .

**Definition 8** The *distance of a code*  $C$  is the smallest distance between distinct words:

$$d_H(C) = \min\{d_H(c_i, c_j) \mid c_i, c_j \in C, c_i \neq c_j\}.$$

*Remark 1* If  $C$  is a linear code, the distance  $d_H(C)$  is the same as the minimum weight of nonzero words:

$$d_H(C) = \min\{\mathbf{w}(c) \mid c \in C, c \neq 0\}.$$

If we know the distance  $d = d_H(C)$  of an  $[n, k]_q$  code, then we can refer to the code as an  $[n, k, d]_q$  code.

**Definition 9** Let  $C$  be an  $[n, k]_q$  code and let  $A_i$  be the number of words of  $C$  of weight  $i$ . The sequence  $\{A_i\}_{i=1}^n$  is called the *weight distribution* of  $C$ .

Note that in a linear code  $A_0 = 1$  and  $\min_{i>0}\{i \mid A_i \neq 0\} = d_H(C)$ .

The distance of a code  $C$  is important to determine the *error correction capability* of  $C$  (that is, the numbers of errors that the code can correct) and its *error detection capability* (that is, the numbers of errors that the code can detect). In fact, we can see the noise as a perturbation that moves a word into some other vector. If the distance between the words is great, there is a low probability that the noise can move a codeword near to another one. To be more precise, we have:

**Theorem 1** Let  $C$  an  $[n, k, d]_q$  code, then

- (a)  $C$  has detection capability  $\ell = d - 1$
- (b)  $C$  has correction capability  $t = \lfloor \frac{d-1}{2} \rfloor$ .

From now on  $t$  denotes the correction capability of the code.

*Example 2* The code in the Example 1 has distance  $d = 3$ . Its detection capability is  $\ell = 2$  and its correction capability is  $t = 1$ .

The following proposition gives an upper bound on the distance of a code in terms of the length and the dimension.

**Proposition 1** (Singleton Bound) *For an  $[n, k, d]_q$  code*

$$d \leq n - k + 1.$$

A code achieving this bound is called *maximum distance separable (MDS)*.

## 2.3 Decoding Linear Codes

In the previous section we have seen that the essence of decoding is to guess which word was sent when a vector  $y$  is received. This means that  $y$  will be decoded as one of the words which is most “likely” to have been sent.

**Proposition 2** *If the transmission uses a  $q$ -ary SC and the probability that a symbol changes into another one is less than the probability that a symbol is uncorrupted by noise, the word sent with the highest probability is the word “nearest” (in the sense of the Hamming distance) to the received vector. If no more than  $t$  (the error correction capability) errors have occurred, this word is unique.*

*Proof* See Hoffman (1991). □

In Example 1 we have informally described this process. We now formally describe the decoding procedure in the linear case. It should be noted that for the remainder  $C$  denotes an  $[n, k]_q$  code.

Let  $c, e, y \in (\mathbb{F}_q)^n$  be the transmitted word, the error, and the received vector, respectively. Then:

$$c + e = y.$$

Given  $y$ , our goal is to determine an  $e$  of minimal weight such that  $y - e$  is in  $C$ . Of course, this vector might not be unique, since there may be more than one word nearest to  $y$ , but if the weight of  $e$  is less than  $t$ , then it is unique. By applying the parity-check matrix  $H$  to  $y$ , we get:

$$Hy^T = H(c + e)^T = He^T = s.$$

**Definition 10** The elements in  $(\mathbb{F}_q)^{n-k}$ ,  $s = Hy^T$ , are called *syndromes*. We say that  $s$  is the syndrome corresponding to  $y$ .

Note that the syndrome depends only on the occurred error  $e$  and not on the particular transmitted word.

Given  $a$  in  $(\mathbb{F}_q)^n$ , we denote the coset  $\{a + c \mid c \in C\}$  by  $a + C$ .  $(\mathbb{F}_q)^n$  can be partitioned into  $q^{n-k}$  cosets of size  $q^k$ . Two vectors  $a, b \in (\mathbb{F}_q)^n$  belong to the same

coset if and only if  $a - b \in C$ . The following fact is just a reformulation of our arguments.

**Theorem 2** Let  $C$  be an  $[n, k, d]_q$  code. Two vectors  $a, b \in (\mathbb{F}_q)^n$  are in the same coset if and only if they have the same syndrome.

**Definition 11** Let  $C$  be an  $[n, k, d]_q$  code. For any coset  $a + C$  and any vector  $v \in a + C$ , we say that  $v$  is a *coset leader* if it is an element of minimum weight in the coset.

**Definition 12** If  $s$  is a syndrome corresponding to an error of weight  $\mathbf{w}(s) \leq t$ , then we say that  $s$  is a *correctable syndrome*.

**Theorem 3** (Correctable syndrome) *If no more than  $t$  errors occurred (i.e.  $\mathbf{w}(e) \leq t$ ), then there exists only one error  $e$  corresponding to the correctable syndrome  $s = He^T$  and  $e$  is the unique coset leader of  $e + C$ .*

We are ready to describe the decoding algorithm. Let  $y$  be a received vector. We want to find an error vector  $e$  of smallest weight such that  $y - e \in C$ . This is equivalent to finding a vector  $e$  of smallest weight in the coset containing  $y$ .

#### Decoding linear codes:

1. after receiving a vector  $y \in (\mathbb{F}_q)^n$ , compute the syndrome  $s = Hy^T$ ;
2. find  $z$ , a coset leader of the corresponding coset;
3. the decoded word is  $c = y - z$ ;
4. recover the message  $m$  from  $c$  (in case of systematic encoding  $m$  consists of first  $k$  components of  $c$ ).

*Remark 2* (Complexity of decoding linear codes) The procedure described above requires some preliminary operations to construct a matrix (named *standard array*) that contains the  $2^n$  vectors of  $(\mathbb{F}_q)^n$  ordered by coset. Then the complexity of the decoding procedure is exponential in terms of memory occupancy.

In Barg et al. (1999), Berlekamp et al. (1978) and (Vardy 1997) it is shown that the general decoding problem for linear codes and the general problem of finding the distance of a linear code are both NP-complete. This suggests that no algorithm exists that decodes linear codes in a polynomial time.

### 3 Some Bounds on Codes

We have seen that the distance  $d$  is an important parameter for a code. A fundamental problem in coding theory is, given the length and the number of codewords (dimension if the code is linear), to determine a code with largest distance, or equivalently, to find the largest code of a given length and distance.

The following definition is useful to state some bounds on codes more clearly.

**Definition 13** Let  $n, d$  be positive integers with  $d \leq n$ . Then the number  $A_q(n, d)$  denotes the maximum number of codewords in a code over  $\mathbb{F}_q$  of length  $n$  and distance  $d$ . This maximum, when restricted to linear code, is denoted by  $B_q(n, d)$ .

Clearly it can be  $B_q(n, d) < A_q(n, d)$ . Then, given  $n$  and  $d$ , if we look at the largest possible code, we have sometimes to use nonlinear codes.

We recall some classical bounds that restrict the existence of codes with given parameters. For any  $x \in (\mathbb{F}_q)^n$  and any positive number  $r$ , let  $B_r(x)$  be the sphere of radius  $r$  centered in  $x$ , with respect to the Hamming distance. Note that the size of  $B_r(x)$  is independent of  $x$  and depends only on  $r, q$  and  $n$ . Let  $V_q(n, r)$  denote the number of elements in  $B_r(x)$  for any  $x \in (\mathbb{F}_q)^n$ . For any  $y \in B_r(x)$ , there are  $(q - 1)$  possible values for each of the  $r$  positions in which  $x$  and  $y$  differ. So we see that

$$V_q(n, r) = \sum_{i=0}^r \binom{n}{i} (q - 1)^i.$$

From the fact that the spheres of radius  $t = \lfloor \frac{d-1}{2} \rfloor$  about codewords are pairwise disjoint, the *sphere packing bound* (or *Hamming bound*) immediately follows:

$$A_q(n, d) \leq \frac{q^n}{V_q(n, t)}.$$

We rewrite the *Singleton bound* (see Proposition 1)

$$A_q(n, d) \leq q^{n+1-d}.$$

Abbreviating  $\gamma = \frac{q-1}{q}$  and assuming  $\gamma n < d$  there holds the *Plotkin bound*, which says that

$$A_q(n, d) \leq \frac{d}{d - \gamma n}.$$

The *Elias bound*, as an extensive refinement of the Plotkin bound, states that for every  $t \in \mathbb{R}$  with  $t < \gamma n$  and  $t^2 - 2t\gamma n + d\gamma n > 0$  there holds

$$A_q(n, d) \leq \frac{\gamma nd}{t^2 - 2t\gamma n + d\gamma n} \cdot \frac{q^n}{V_q(n, t)}.$$

We conclude with a lower bound, the *Gilbert–Varshamov bound*

$$A_q(n, d) \geq B_q(n, d) \geq \frac{q^n}{V_q(n, d - 1)}.$$

## 4 Cyclic Codes

### 4.1 An Algebraic Correspondence

**Definition 14** An  $[n, k, d]_q$  linear code  $C$  is *cyclic* if the cyclic shift of a word is also a word, i.e.

$$(c_0, \dots, c_{n-1}) \in C \implies (c_{n-1}, c_0, \dots, c_{n-2}) \in C.$$

To describe algebraic properties of cyclic codes, we need to introduce a new structure. We consider the univariate polynomial ring  $\mathbb{F}_q[x]$  and the ideal  $I = \langle x^n - 1 \rangle$ . We denote by  $R$  the ring  $\mathbb{F}_q[x]/I$ . We construct a bijective correspondence between the vectors of  $(\mathbb{F}_q)^n$  and the residue classes of polynomials in  $R$ :

$$\mathbf{v} = (v_0, \dots, v_{n-1}) \longleftrightarrow v_0 + v_1x + \dots + v_{n-1}x^{n-1}.$$

We can view linear codes as subsets of the ring  $R$ , thanks to the correspondence above. The following theorem points out the algebraic structure of cyclic codes.

**Theorem 4** Let  $C$  be an  $[n, k, d]_q$  code, then  $C$  is cyclic if and only if  $C$  is an ideal of  $R$ .

*Proof* Multiplying by  $x$  modulo  $x^n - 1$  corresponds to a cyclic shift:

$$(c_0, c_1, \dots, c_{n-1}) \rightarrow (c_{n-1}, c_0, \dots, c_{n-2})$$

$$x(c_0 + c_1x + \dots + c_{n-1}x^{n-1}) = c_{n-1} + c_0x + \dots + c_{n-2}x^{n-2}. \quad \square$$

Since  $R$  is a principal ideal ring, if  $C$  is not trivial there exists a unique monic polynomial  $g$  that generates  $C$ . We call  $g$  the *generator polynomial* of  $C$ . Note that  $g$  divides  $x^n - 1$  in  $\mathbb{F}_q[x]$ . If the dimension of the code  $C$  is  $k$ , the generator polynomial has degree  $n - k$ .

A generator matrix can easily be given by using the coefficients of the generator polynomial  $g = \sum_{i=0}^{n-k} g_i x^i$ :

$$G = \begin{pmatrix} g \\ xg \\ \vdots \\ x^k g \end{pmatrix} = \begin{pmatrix} g_0 & g_1 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & g_0 & \dots & g_{n-k-1} & g_{n-k} & 0 & \dots \\ \vdots & & \ddots & & \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & g_0 & g_1 & \dots & g_{n-k} \end{pmatrix}.$$

Moreover, a polynomial  $f$  in  $R$  belongs to the code  $C$  if and only if there exists  $q$  in  $R$  such that  $qg = f$ .

Since the generator polynomial is a divisor of  $x^n - 1$  and is unique, the *parity-check* polynomial of  $C$  is well defined as the polynomial  $h(x)$  in  $R$  such that  $h(x) = (x^n - 1)/g(x)$ . The parity-check polynomial provides a simple way to check if an  $f(x)$  in  $R$  belongs to  $C$ , since

$$f(x) \in C \Leftrightarrow f(x) = q(x)g(x) \Leftrightarrow f(x)h(x) = q(x)(g(x)h(x)) = 0 \text{ in } R.$$

**Proposition 3** Let  $h(x), g(x)$  be, respectively, the parity-check and the generator polynomial of the cyclic code  $C$ . The dual code  $C^\perp$  is cyclic with generator polynomial

$$g^\perp(x) = x^{\deg(h)} h(x^{-1}).$$

*Proof* The generator matrix obtained by  $g^\perp(x)$  has the form:

$$H = \begin{pmatrix} & h_k & \dots & h_1 & h_0 \\ h_k & \dots & h_1 & h_0 & \\ & \vdots & & & \\ h_k & \dots & h_1 & h_0 & \end{pmatrix}.$$

Given  $c$  in  $R$ , the  $i$ -th component of  $H \cdot c^T$  is  $x^i h(x)c(x)$ , which vanishes if and only if  $c \in C$ .  $\square$

## 4.2 Encoding and Decoding with Cyclic Codes

The properties of cyclic codes suggest a very simple method to encode a message. Let  $C$  be an  $[n, k, d]_q$  cyclic code with generator polynomial  $g$ , then  $C$  is capable of encoding  $q$ -ary messages of length  $k$  and requires  $n - k$  redundancy symbols.

Let  $m = (m_0, \dots, m_{k-1})$  be a message to encode, we consider its polynomial representation  $m(x)$  in  $R$ . To obtain an associated word it is sufficient to multiply  $m(x)$  by the generator polynomial  $g(x)$ :

$$c(x) = m(x)g(x) \in C.$$

Even if this way to encode is the simpler, another procedure is used to obtain a systematic encoding, which again exploits some properties of the polynomial ring.

Given the message  $m(x)$ , multiply it by  $x^{n-k}$  and divide the result by  $g$ , obtaining:

$$m(x)x^{n-k} = q(x)g(x) + r(x)$$

where  $\deg(r(x)) < \deg(g(x)) = n - k$ . So the remainder can be thought of as an  $(n - k)$ -vector. Joining the  $k$ -vector  $m$  with the  $(n - k)$ -vector  $r$  we obtain an  $n$ -vector  $c$ , which is the encoded word, i.e.:

$$c(x) = m(x)x^{n-k} + r(x).$$

This way, in absence of errors the decoding is immediate: the message is formed by the last  $k$  components of the received word.

On the other hand, the receiver does not know if no errors have occurred during transmission, but it is sufficient to check if the remainder of the division of the received polynomial by  $g$  is equal to zero to state that it is most likely that no errors have occurred.

It is not hard to prove that if an error  $e$  occurred during the transmission, the remainder of the division by  $g$  in the procedure below gives exactly the syndrome associated to  $e$ , and then we can find  $e$  in the same way as described for linear codes.

Other decoding procedures exist for particular cyclic codes, such as the BCH codes (Bose and Ray-Chaudhuri 1960), which work faster than the procedure above. (See Sect. 7.)

### 4.3 Zeros of Cyclic Codes

Cyclic codes of length  $n$  over  $\mathbb{F}_q$  are generated by divisors of  $x^n - 1$ . Let

$$x^n - 1 = \prod_{j=1}^r f_j, \quad f_j \text{ irreducible over } \mathbb{F}_q.$$

Then to any cyclic code of length  $n$  over  $\mathbb{F}_q$  there corresponds a subset of  $\{f_j\}_{j=1}^r$ . A very interesting case <sup>1</sup> is when  $\text{GCD}(n, q) = 1$ . Let  $\mathbb{F} = \mathbb{F}_{q^m}$  be the splitting field of  $x^n - 1$  over  $\mathbb{F}_q$  and let  $\alpha$  be a primitive  $n$ -th root of unity over  $\mathbb{F}_q$ . We have:

$$x^n - 1 = \prod_{i=0}^{n-1} (x - \alpha^i).$$

In this case the generator polynomial of  $C$  has powers of  $\alpha$  as roots. We recall that, given  $g \in \mathbb{F}_q[x]$ , if  $g(\alpha^i) = 0$  then  $g(\alpha^{qi}) = 0$ .

**Definition 15** Let  $C$  be an  $[n, k, d]_q$  cyclic code with generator polynomial  $g_C$ , with  $\text{GCD}(n, q) = 1$ . The set:

$$S_{C,\alpha} = S_C = \{i_1, \dots, i_{n-k} \mid g_C(\alpha^{i_j}) = 0, j = 1, \dots, n - k\}$$

---

<sup>1</sup>In Castagnoli et al. (1991) is shown that if there exists a family of “good” codes  $\{C_m\}_m$  over  $\mathbb{F}_q$  of lengths  $m$  with  $\text{GCD}(m, q) \neq 1$ , there exists a family  $\{C'_n\}_n$  with  $\text{GCD}(n, q) = 1$  with the same properties.

is called the *complete defining set* of  $C$ .

We can collect the integers modulo  $n$  into  $q$ -cyclotomic classes  $C_i$ :

$$\{0, \dots, n-1\} = \bigcup C_i, \quad C_i = \{i, q_i, \dots, q^r i\},$$

where  $r$  is the smallest positive integer such that  $i \equiv iq^r \pmod{n}$ . So the complete defining set of a cyclic code is collection of  $q$ -cyclotomic classes.

From now on we fix a primitive  $n$ -th root of unity  $\alpha$  and we write  $S_{C,\alpha} = S_C$ . A cyclic code is defined by its complete defining set, since

$$C = \{c \in R \mid c(\alpha^i) = 0, i \in S_C\} \iff g_C = \prod_{i \in S_C} (x - \alpha^i).$$

By this fact it follows that

$$H = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \cdots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \cdots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{i_{n-k}} & \alpha^{2i_{n-k}} & \cdots & \alpha^{(n-1)i_{n-k}} \end{pmatrix}$$

is a parity-check (defined over  $\mathbb{F}_{q^m}$ ) matrix for  $C$ , since

$$Hc^T = \begin{pmatrix} c(\alpha^{i_1}) \\ c(\alpha^{i_2}) \\ \vdots \\ c(\alpha^{i_{n-k}}) \end{pmatrix} = \underline{0} \Leftrightarrow c \in C.$$

*Remark 3*  $H$  maybe defined over  $\mathbb{F}_{q^m}$ , but  $C$  is its null space over  $\mathbb{F}_q$ .

*Remark 4* We note that, as  $S_C$  is partitioned into cyclotomic classes, there are some subsets  $S'_C$  of  $S_C$  any of them sufficient to specify the code unambiguously and we call any such  $S'_C$  a *defining set*.

## 5 Some Examples of Cyclic Codes

### 5.1 Hamming and Simplex Codes

**Definition 16** A code which attains the Hamming bound (see Sect. 3) is called a *perfect* code.

In other words, a code is said to be perfect if for every possible vector  $v$  in  $(\mathbb{F}_q)^n$  there is a unique word  $c \in C$  such that  $d_H(v, c) \leq t$ .

Let  $C$  be an  $[n, n - r, d]_q$  code with parity-check matrix  $H \in (\mathbb{F}_q)^{r \times n}$ . We denote by  $\{H_i\}_{i=1}^n$  the set of columns of  $H$ . We observe that if two columns  $H_i, H_j$  belongs to the same line in  $(\mathbb{F}_q)^r$  (i.e.  $H_j = \lambda H_i$ ), then the vector

$$c = (0, \dots, 0, \underset{i}{-\lambda}, 0, \dots, 0, \underset{j}{1}, 0, \dots, 0)$$

belongs to  $C$ , since  $Hc^T = 0$ . Then  $d(C) \leq 2$ . On other hand, if we construct a parity-check matrix  $H$  such that the columns  $H_i$  belong to different lines, the corresponding linear code has distance at least 3.

**Definition 17** (Hamming 1950) An *Hamming Code* is a linear code for which the set of columns of  $H \in (\mathbb{F}_q)^{n \times r}$  contains all nonzero vectors in  $(\mathbb{F}_q)^r$ .

By the definition above, given two columns  $H_i, H_j$  of  $H$ , there exists a third column  $H_k$  of  $H$ , and  $\lambda \in \mathbb{F}_q$  such that  $H_k = \lambda(H_i + H_j)$ . This fact implies that

$$c = (0, \dots, 0, \underset{i}{-\lambda}, 0, \dots, 0, \underset{j}{-\lambda}, 0, \dots, 0, \underset{k}{1}, 0, \dots, 0)$$

is a word, and hence the minimum distance of a Hamming code is 3. In the vector space  $(\mathbb{F}_q)^r$  there are  $n = \frac{q^r - 1}{q - 1}$  distinct lines, each with  $q - 1$  elements different from zero. Hence:

**Proposition 4** An  $[n, k, d]_q$  code is a Hamming code, if and only if  $n = \frac{q^r - 1}{q - 1}$ ,  $k = n - r$ ,  $d = 3$ , for some  $r \in \mathbb{N}^*$ .

On the other hand, a direct computation shows that:

**Proposition 5** The Hamming codes are perfect codes.

*Example 3* Let  $C$  be the  $[7, 4, 3]_2$  code with parity-check matrix:

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

Then  $C$  is an  $[7, 4, 3]$  Hamming code. Note that the columns of  $H$  are exactly the non-zero vectors of  $(\mathbb{F}_2)^3$ .

The following theorem states that Hamming codes are cyclic.

**Theorem 5** Let  $n = (q^r - 1)/(q - 1)$ . If  $\text{GCD}(n, q - 1) = 1$ , then the cyclic code over  $\mathbb{F}_q$  of length  $n$  with defining set  $\{1\}$  is a  $[n, n - r, 3]$  Hamming code.

*Proof* By Proposition 4 it is sufficient to show that the distance of  $C$  is equal to 3. The Hamming bound applied to  $C$  ensures that the distance cannot be greater than 3; we show that it can not be 2 (it is obvious that it is not one). Let  $\alpha$  be a primitive  $n$ -th root of unity over  $\mathbb{F}_q$  such that  $c(\alpha) = 0$  for  $c$  in  $C$ . If  $c$  is a word of weight 2 with nonzero coefficients  $c_i$  and  $c_j$  ( $i < j$ ), then  $c_i\alpha^i + c_j\alpha^j = 0$ . Then  $\alpha^{j-i} = -c_i/c_j$ . Since  $-c_i/c_j \in \mathbb{F}_q^*$ ,  $\alpha^{(j-i)(q-1)} = 1$ . Now  $\text{GCD}(n, q-1) = 1$  implies that  $\alpha^{j-i} = 1$ , but this is a contradiction since  $0 < j - i < n$  and the order of  $\alpha$  is  $n$ .  $\square$

*Example 4* The Hamming code of Example 3 can be viewed as the  $[7, 4, 3]_2$  cyclic code with generator polynomial  $g = x^3 + x + 1$ .

We have seen that the dual code of a cyclic code is cyclic itself. This means in particular that the dual of a Hamming code is cyclic.

**Definition 18** The dual of a Hamming code is called a *simplex code*.

The simplex code has the following property:

**Proposition 6** A simplex code is a  $[(q^r - 1)/(q - 1), r, q^{r-1}]$  constant weight code over  $\mathbb{F}_q$ .

## 5.2 Quadratic Residue Codes

Let  $n$  be an odd prime. We denote by  $\mathcal{Q}_n \subset \{1, \dots, n-1\}$  the set of *quadratic residues modulo n*, i.e.:

$$\mathcal{Q}_n = \{k \mid k \equiv x^2 \pmod{n} \text{ for some } x \in \mathbb{Z}\}.$$

If  $q$  is a quadratic residue modulo  $n$ , it is easy to see that  $\mathcal{Q}_n$  is a collection of  $q$ -cyclotomic classes with cardinality  $(n-1)/2$ . Then we can give the following definition.

**Definition 19** Let  $n$  be a positive integer relatively prime to  $q$  and let  $\alpha$  be a primitive  $n$ -th root of unity. Suppose that  $n$  is an odd prime and  $q$  is a quadratic residue modulo  $n$ . The  $[n, (n-1)/2+1]_q$  cyclic code with complete defining set  $\mathcal{Q}_n$  is called *quadratic residue code*.

*Example 5* The  $[23, 12, 7]_2$  quadratic residue code is the *perfect binary Golay code*.

## 6 BCH Codes

**Theorem 6 (BCH bound)** Let  $C$  be an  $[n, k, d]_q$  cyclic code with defining set  $S_C = \{i_1, \dots, i_{n-k}\}$  and let  $(n, q) = 1$ . Suppose there are  $\delta - 1$  consecutive numbers

in  $S_C$ , say  $\{m_0, m_0 + 1, \dots, m_0 + \delta - 2\} \subset S_C$ . Then

$$d \geq \delta.$$

**Definition 20** Let  $S = (m_0, m_0 + 1, \dots, m_0 + \delta - 2)$  be such that

$$0 \leq m_0 \leq \dots \leq m_0 + \delta - 2 \leq n - 1$$

If  $C$  is the  $[n, k, d]_q$  cyclic code with defining set  $S$ , we say that  $C$  is a *BCH code* of *designed distance*  $\delta$ . The BCH code is called *narrow sense* if  $m_0 = 1$  and it is called *primitive* if  $n = q^m - 1$ .

*Example 6* We consider the polynomial  $x^7 - 1$  over  $\mathbb{F}_2$ :

$$\begin{array}{ccc} f_0 & f_1 & f_3 \\ \parallel & \parallel & \parallel \\ x^7 - 1 = (x + 1) (x^3 + x^2 + 1) (x^3 + x + 1) \end{array}$$

Let  $C$  be the cyclic code generated by  $g = f_0 \cdot f_1$ . Then  $S_C = \{0, 1, 2, 4\}$  with respect to a primitive  $n$ -th root of unity  $\alpha$  s.t.  $f_1(\alpha) = 0$ .  $C$  is a  $[7, 3, d]_2$  code with  $S_C = \{0, 1, 2, 4\}$  and so it is a BCH code of designed distance  $\delta = 4$ . The BCH bound ensures that the minimum distance is at least 4. On the other hand, the generator polynomial

$$g(x) = x^4 + x^2 + x + 1$$

has weight 4 and we finally can state that  $d = 4$ .

## 6.1 On the Optimality of BCH Codes

**Definition 21** Given  $n$  and  $d$  two integers, a code is said to be *optimal* if it has maximal size in the class of codes with length  $n$  and distance  $d$ .

**Theorem 7** Narrow sense primitive binary BCH codes of fixed minimum distance are optimal when the length is large, but the relative distance

$$d/n \rightarrow 0.$$

In other words, consider such an  $[n, k, d]_2$  BCH code, with  $t = \lfloor \frac{d-1}{2} \rfloor$ , then  $k \geq n - mt$ . Then there does not exist a  $t+1$  correcting code with the same length and dimension.

*Proof* Let  $t$  be fixed, and let  $n = 2^m - 1$  go to infinity. Then

$$\begin{aligned} V_2(n, t+1) &= \sum_{i \leq t+1} \binom{n}{i} > \binom{n}{t+1} = \frac{n!}{(t+1)!(n-t-1)!} \\ &= O\left(\frac{1}{(t+1)!} \cdot n^{t+1}\right) \sim \frac{1}{(t+1)!} 2^{m(t+1)} \gg 2^{mt} > 2^{n-k}. \end{aligned}$$

This means that the Hamming bound is exceeded for the parameters  $n, k$  and  $t + 1$ , which implies that a  $t + 1$  error correcting code does not exist.  $\square$

A precise evaluation of the length  $n$  such that an  $[n, k, n - mt]$  BCH code is optimal is given in Berlekamp (1984), p. 299.

We now define a subclass of BCH codes that are always optimal (Reed and Solomon 1960).

**Definition 22** A *Reed Solomon* code over  $\mathbb{F}_q$  is a BCH code with length  $n = q - 1$ .

Note that if  $n = q - 1$  then  $x^n - 1$  splits into linear factors. If the designed distance is  $d$ , then the generator polynomial of a RS code has the form  $g(x) = (x - \alpha^{i_0})(x - \alpha^{i_0+1}) \cdots (x - \alpha^{i_0+d-1})$  and  $k = n - d + 1$ . It follows that RS codes are MDS codes.

## 7 Decoding BCH Codes

There are several algorithms for decoding BCH codes. In this section we briefly discuss the method, first developed in 1975 by Sugiyama et al. (1975), that uses the extended Euclidean algorithm to solve the key equation. Note that the Berlekamp–Massey (1968, 1969) algorithm is more used. Some alternative decoding algorithms are detailed in Mora and Orsini (2009), Guerrini and Rimoldi (2009).

Let  $C$  be a BCH code of length  $n$  over  $\mathbb{F}_q$ , with designed distance  $\delta = 2t + 1$  (where  $t$  is the error correction capability of the code), and let  $\alpha$  be a primitive  $n$ -th root of unity in  $\mathbb{F}_{q^m}$ . We consider a word  $c(x) = c_0 + \cdots + c_{n-1}x^{n-1}$  and we assume that the received word is  $v(x) = v_0 + \cdots + v_{n-1}x^{n-1}$ . Then the error vector can be represented by the *error polynomial*

$$e(x) = v(x) - c(x) = e_0 + e_1x + \cdots + e_{n-1}x^{n-1}.$$

If the weight of  $e$  is  $\mu \leq t$ , let

$$L = \{l \mid e_l \neq 0, 0 \leq l \leq n - 1\}$$

be the set of the *error positions*, and  $\{\alpha^l \mid l \in L\}$  the set of the *error locators*. Then the *classical error locator polynomial* is defined by

$$\sigma(x) = \prod_{l \in L} (1 - x\alpha^l),$$

i.e. the univariate polynomial which has as zeros the reciprocal of the error locations. The error locations can also be obtained by the *plain error locator polynomial*, that is

$$L_e(x) = \prod_{l \in L} (x - \alpha^l).$$

The *error evaluator polynomial* is defined by

$$\omega(x) = \sum_{l \in L} e_l \alpha^l \prod_{i \in L \setminus \{l\}} (1 - x \alpha^i).$$

We find the two polynomials  $\sigma(x)$  and  $\omega(x)$  to correct errors: an error is in position  $l$  if and only if  $\sigma(\alpha^{-l}) = 0$  and in this case the value of the error is:

$$e_l = -\alpha^l \frac{\omega(\alpha^{-l})}{\sigma'(\alpha^{-l})}, \quad (1)$$

in fact, since the derivative  $\sigma'(x) = \sum_{l \in L} -\alpha^l \prod_{i \neq l} (1 - x \alpha^i)$ , so  $\sigma'(\alpha^{-l}) = -\alpha^l \prod_{i \neq l} (1 - \alpha^{i-l})$  and  $\sigma'(\alpha^{-l}) \neq 0$ . The goal of decoding can be reduced to determine the error locator polynomial and apply a search of the roots (Chien 1964) to obtain the error positions. We need the following lemma later on.

**Lemma 1** *The polynomials  $\sigma(x)$  and  $\omega(x)$  are relatively prime.*

*Proof* It is obvious, since no zero of  $\sigma(x)$  is a zero of  $\omega(x)$ .  $\square$

We are now ready to describe the decoding algorithm.

### The First Step: the Key Equation

At the first step we calculate the syndrome of the received vector  $v(x)$ :

$$\begin{aligned} Hv^T &= \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \alpha^{\delta-1} & \alpha^{2(\delta-1)} & \dots & \alpha^{(\delta-1)(n-1)} \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{pmatrix} = \begin{pmatrix} e(\alpha) \\ e(\alpha^2) \\ \vdots \\ e(\alpha^{\delta-1}) \end{pmatrix} \\ &= \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_{2t} \end{pmatrix} \end{aligned}$$

We define the *syndrome polynomial*:

$$S(x) = S_1 + S_2 x + \dots + S_{2t} x^{2t-1},$$

where  $S_i = e(\alpha^i) = \sum_{l \in L} e_l \alpha^{il}$ ,  $i = 1, \dots, 2t$ . The following theorem establishes a relation among  $\sigma(x)$ ,  $\omega(x)$  and  $S(x)$ .

**Theorem 8** (The key equation) *The polynomials  $\sigma(x)$  and  $\omega(x)$  satisfy:*

$$\sigma(x)S(x) \equiv \omega(x) \pmod{x^{2t}} \quad (\text{key equation})$$

If there exist two polynomials  $\sigma_1(x)$ ,  $\omega_1(x)$ , such that  $\deg(\omega_1(x)) < \deg(\sigma_1(x)) \leq t$  and that satisfy the key equation, then there is a polynomial  $\lambda(x)$  such that  $\sigma_1(x) = \lambda(x)\sigma(x)$  and  $\omega_1(x) = \lambda(x)\omega(x)$ .

*Proof* Interchanging summations and the sum formula for a geometric series, we get

$$\begin{aligned} S(x) &= \sum_{j=1}^{2t} e(\alpha^j)x^{j-1} = \sum_{j=1}^{2t} \sum_{l \in L} e_l \alpha^{jl} x^{j-1} \\ &= \sum_{l \in L} e_l \alpha^l \sum_{j=1}^{2t} (\alpha^l x)^{j-1} = \sum_{l \in L} e_l \alpha^l \frac{1 - (\alpha^l x)^{2t}}{1 - \alpha^l x}. \end{aligned}$$

Thus

$$\sigma(x)S(x) = \prod_{i \in L} (1 - \alpha^i x)S(x) = \sum_{l \in L} e_l \alpha^l (1 - (\alpha^l x)^{2t}) \prod_{i \neq l \in L} (1 - \alpha^i x),$$

and then

$$\sigma(x)S(x) \equiv \sum_{l \in L} e_l \alpha^l \prod_{i \neq l \in L} (1 - \alpha^i x) \equiv \omega(x) \pmod{x^{2t}}.$$

Suppose we have another pair  $(\sigma_1(x), \omega_1(x))$  such that

$$\sigma_1(x)S(x) \equiv \omega_1(x) \pmod{x^{2t}}$$

and  $\deg(\omega_1(x)) < \deg(\sigma_1(x)) \leq t$ . Then

$$\sigma(x)\omega_1(x) \equiv \sigma_1(x)\omega(x) \pmod{x^{2t}}$$

and the degrees of  $\sigma(x)\omega_1(x)$  and  $\sigma_1(x)\omega(x)$  are strictly smaller than  $2t$ . Since  $GCD(\sigma(x), \omega(x)) = 1$  by Lemma 1, there exists a polynomial  $\lambda(x)$  s.t.  $\sigma_1(x) = \lambda(x)\sigma(x)$  and  $\omega_1(x) = \lambda(x)\omega(x)$ .  $\square$

## The Second Step: the Extended Euclidean Algorithm

Once we have the syndrome polynomial  $S(x)$ , the second step of the decoding algorithm consists of finding  $\sigma(x)$  and  $\omega(x)$ , using the key equation.

**Theorem 9** (Bezout's Identity) *Let  $\mathcal{K}$  be a field and  $f(x), g(x) \in \mathcal{K}[x]$ . Let us denote  $d(x) = \gcd(f(x), g(x))$ . Then there are  $u(x), v(x) \in \mathcal{K}[x] \setminus \{0\}$ , such that:*

$$f(x)u(x) + g(x)v(x) = d(x).$$

It is well known that it is possible to find the greatest common divisor  $d(x)$  and the polynomials  $u(x)$  and  $v(x)$  in Bezout's identity using the Extended Euclidean Algorithm (EEA). Suppose that  $\deg(f(x)) > \deg(g(x))$ , then let:

$$\begin{aligned} u_{-1} &= 1, & v_{-1} &= 0, & d_{-1} &= f(x), \\ u_0 &= 0, & v_0 &= 1, & d_0 &= g(x). \end{aligned}$$

The first step of the Euclidean algorithm is:

$$d_1(x) = d_{-1}(x) - q_1(x)d_0(x) = f(x) - q_1(x)g(x),$$

so that

$$u_1(x) = 1, v_1(x) = -q_1(x) \quad \text{and}$$

$$\deg(d_1) < \deg(d_0) \quad \text{and} \quad \deg(v_1) < \deg(d_{-1}) - \deg(d_0).$$

From the  $j$ -th step, we get:

$$\begin{aligned} d_j(x) &= d_{j-2}(x) - q_j(x)d_{j-1}(x) \\ &= u_{j-2}(x)f(x) + v_{j-2}(x)g(x) - q_j(x)[u_{j-1}(x)f(x) + v_{j-1}(x)g(x)] \\ &= [-q_j(x)u_{j-1}(x) + u_{j-2}(x)]f(x) \\ &\quad + [-q_j(x)v_{j-1}(x)f(x) + v_{j-2}(x)]g(x). \end{aligned}$$

This means:

$$u_j(x) = -q_j(x)u_{j-1}(x) + u_{j-2}(x) \quad \text{and} \quad v_j(x) = -q_j(x)v_{j-1}(x) + v_{j-2}(x)$$

with  $\deg(d_j) \leq \deg(d_{j-1})$ ,  $\deg(u_j) = \sum_{i=2}^j \deg(q_i)$ ,  $\deg(v_j) = \sum_{i=2}^j \deg(q_i)$  and  $\deg(v_j) = \deg(f) - \deg(d_{j-1})$ . The algorithm proceeds by dividing the previous remainder by the current remainder until this becomes zero.

---

STEP 1	$d_{-1}(x) = q_1(x)d_0(x) + d_1(x),$	$\deg(d_1) < \deg(d_0)$
STEP 2	$d_0(x) = q_2(x)d_1(x) + d_2(x),$	$\deg(d_2) < \deg(d_1)$
$\vdots$	$\vdots$	
STEP $j$	$d_{j-2}(x) = q_j(x)d_{j-1}(x) + d_j(x),$	$\deg(d_j) < \deg(d_{j-1})$
$\vdots$	$\vdots$	
STEP $k$	$d_{k-1}(x) = q_{k+1}(x)d_k(x)$	

---

We conclude that the  $\text{GCD}(f(x), g(x)) = \text{GCD}(d_{-1}(x), d_0(x)) = d_k(x)$ .

We would like to be able to find  $\omega(x)$  and  $\sigma(x)$  using the Euclidean algorithm. First we observe that  $\deg(\sigma(x)) \leq t$  and  $\deg(\omega(x)) \leq t - 1$ . For this reason we apply the EEA to the known polynomials  $f(x) = x^{2t}$  and  $g(x) = S(x)$ , until we find a  $d_{k-1}(x)$  such that:

$$\deg(d_{k-1}(x)) \geq t \quad \text{and} \quad \deg(d_k(x)) \leq t - 1.$$

In this way we obtain a polynomial  $d_k(x)$  such that:

$$d_k(x) = x^{2t} u_k(x) + S(x) v_k(x), \quad (2)$$

with  $\deg(v_k(x)) = \deg(x^{2t}) - \deg(d_{k-1}(x)) \leq 2t - t = t$ .

**Theorem 10** *Let  $d_k(x)$  and  $v_k(x)$  as in (2). Then the polynomials  $v_k(x)$  and  $d_k(x)$  are scalar multiples of  $\sigma(x)$  and  $\omega(x)$ , respectively, i.e.:*

$$\sigma(x) = \lambda v_k(x) \quad \omega(x) = \lambda d_k(x),$$

for some scalar  $\lambda \in \mathbb{F}_q$ .

We can determine  $\lambda$  by  $\sigma(0) = 1$ , i.e.  $\lambda = v_k(0)^{-1}$ . So we have:

$$\sigma(x) = \frac{v_k(x)}{v_k(0)}, \quad \omega(x) = \frac{d_k(x)}{v_k(0)}.$$

### The Third Step: Determining the Error Values

In the last step we have to calculate the error values. In the binary case it is immediate. Otherwise we can use the relations

$$e_l = -\alpha^l \frac{\omega(\alpha^{-l})}{\sigma'(\alpha^{-l})}, \quad l = 1, \dots, \mu.$$

See also Forney (1965).

## 8 On the Asymptotic Properties of Cyclic Codes

There is a longstanding question which is to know whether the class of cyclic codes is *asymptotically good*. Let us recall that a sequence of linear binary  $[n_i, k_i, d_i]_2$  codes  $C_i$  is asymptotically good if

$$\liminf \frac{k_i}{n_i} > 0, \quad \text{and} \quad \liminf \frac{d_i}{n_i} > 0.$$

The first explicit construction of an asymptotically good sequence of codes is due to Justesen (1972), but the codes are not cyclic. Although it is known that the class of BCH codes is not asymptotically good (Camion 1969; Lin and Weldon 1967), (see MacWilliams and Sloane 1977 for a proof), we do not know if there is a family of asymptotically good cyclic codes. Still on the negative side, Castagnoli (Castagnoli 1989) has shown that, if the length  $n_i$  goes to infinity while having a fixed set of prime factors, then there is no asymptotically good family of codes  $C_i$  of length  $n_i$ . Other negative results are in Berman 1967. Known partial positive results are due to Kasami (1974), for *quasi-cyclic codes*.<sup>2</sup> Bazzi and Mitter (2006) have shown that there exists an asymptotically good family of linear codes which are very close to cyclic codes. Willems and Martínez-Pérez (2006) have shown that there exists an asymptotically good family of cyclic codes, provided there exists an asymptotically good family of linear codes  $C_i$  with special properties on their lengths  $n_i$ . So, although some progress has been achieved, the question is still open.

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

The second and third author would like to thank M. Sala (their supervisor).

## References

- A. M. Barg, E. Krouk, and H. C. A. van Tilborg, *On the complexity of minimum distance decoding of long linear codes*, IEEE Trans. on Inf. Th. **45** (1999), no. 5, 1392–1405.
- L. M. Bazzi and S. K. Mitter, *Some randomized code constructions from group actions*, IEEE Trans. on Inf. Th. **52** (2006), 3210–3219.
- E. R. Berlekamp, *Algebraic coding theory*, McGraw–Hill, New York, 1968.
- E. R. Berlekamp, *Algebraic coding theory (revised edition)*, Aegean Park, Walnut Creek, 1984.
- E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. on Inf. Th. **24** (1978), no. 3, 384–386.
- S. D. Berman, *Semisimple cyclic and Abelian codes. II*, Cybernetics **3** (1967), no. 3, 17–23.
- R. E. Blahut, *Theory and practice of error control codes*, Addison–Wesley, Reading, 1983.
- R. C. Bose and D. K. Ray-Chaudhuri, *On a class of error correcting binary group codes*, Information and Control **3** (1960), 68–79.
- P. Camion, *A proof of some properties of Reed–Muller codes by means of the normal basis theorem*, Combinatorial mathematics and its application, Univ. of North Carolina, Chapel Hill, 1969, pp. 371–376.
- G. Castagnoli, *On the asymptotic badness of cyclic codes with block-lengths composed from a fixed set of prime factors*, LNCS, vol. **357**, Springer, Berlin, 1989, pp. 164–168.
- G. Castagnoli, J. L. Massey, P. A. Schoeller, and N. von Seeman, *On repeated-root cyclic codes*, IEEE Trans. on Inf. Th. **37** (1991), 337–342.
- R. T. Chien, *Cyclic decoding procedure for the Bose–Chaudhuri–Hocquenghem codes*, IEEE Trans. on Inf. Th. **10** (1964), 357–363.
- G. D. Forney, *On decoding BCH codes*, IEEE Trans. on Inf. Th. **11** (1965), 549–557.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick’s approach to recent developments*, this volume, 2009, pp. 197–218.

---

<sup>2</sup>A code is quasi-cyclic code if it is invariant by a power of the cyclic shift.

- R. W. Hamming, *Error detecting and error correcting codes*, Bell Systems Technical Journal **29** (1950), 147–160.
- D. G. Hoffman, *Coding theory: The essential*, Dekker, New York, 1991.
- J. Justesen, *A class of constructive asymptotically good algebraic codes*, IEEE Trans. on Inf. Th. **18** (1972), no. 5, 652–656.
- T. Kasami, *A Gilbert-Varshamov bound for quasi-cyclic codes of rate 1/2*, IEEE Trans. on Inf. Th. **20** (1974), 679–679.
- S. Lin, *An introduction to error-correcting codes*, Prentice Hall, New York, 1970.
- S. Lin and E. J. Weldon, *Long BCH codes are bad*, Information Control **11** (1967), 445–451.
- F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes. I and II*, North-Holland, Amsterdam, 1977.
- C. Martinez-Perez and W. Willems, *Is the class of cyclic codes asymptotically good?* IEEE Trans. on Inf. Th. **52** (2006), no. 2, 696–700.
- J. L. Massey, *Shift-register synthesis and BCH decoding*, IEEE Trans. on Inf. Th. **15** (1969), 122–127.
- T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, this volume, 2009, pp. 69–91.
- W. W. Peterson and E. J. Weldon Jr., *Error-correcting codes*, second ed., MIT Press, Cambridge, 1972.
- V. Pless, *Introduction to the theory of error-correcting codes*, Wiley, New York, 1982.
- V. S. Pless, W. C. Huffman, and R. A. Brualdi (eds.), *Handbook of coding theory, vols. I, II*, North-Holland, Amsterdam, 1998.
- I. S. Reed and G. Solomon, *Polynomial codes over certain finite fields*, J. Soc. Indust. Appl. Math. **8** (1960), 300–304.
- C. E. Shannon, *A mathematical theory of communication*, Bell System Tech. J. **27** (1948), 379–423, 623–656.
- Y. Sugiyama, S. Kasahara, S. Hirasawa, and T. Namekawa, *A method for solving key equation for decoding Goppa codes*, Inform. Contr. **27** (1975), 87–99.
- J. H. van Lint, *Introduction to coding theory*, third ed., Graduate Texts in Mathematics, vol. **86**, Springer, Berlin, 1999.
- A. Vardy, *Algorithmic complexity in coding theory and the minimum distance problem*, Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, 1997, pp. 92–109.

# Decoding Cyclic Codes: the Cooper Philosophy

Teo Mora and Emmanuela Orsini

**Abstract** In 1990 Cooper suggested to use Gröbner basis computations in order to deduce error locator polynomials of cyclic codes.

The aim of this tutorial is to show, with illuminating examples, how Cooper's approach has been refined up to give both an *online decoder* and *general error locator polynomials*.

## 1 Introduction

In this tutorial we assume that the reader is familiar with the notation for linear and cyclic codes adopted in Augot et al. (2009). In particular, we will use without comments concepts like: generator polynomial, defining set, correctable syndrome, error polynomial, classical error locator polynomial and plain error locator polynomial. We also assume that the reader is familiar with Gröbner bases (Buchberger 1965, 1985, 2006) especially with the notation in Mora (2009).

In 1990 Cooper (1990, 1991, 1993) suggested to use Gröbner basis computations in order to correct cyclic codes (for a different approach see Guerrini and Rimoldi 2009). Let  $C$  be a binary BCH code correcting up to  $t$  errors,  $\bar{s} = (s_1, \dots, s_{2t-1})$  be the syndrome vector associated to a received word. Cooper's idea consisted in interpreting the error locations of  $C$  as the roots of the syndrome equation system:

$$f_i := \sum_{j=1}^t z_j^{2i-1} - s_{2i-1} = 0, \quad 1 \leq i \leq t,$$

and, consequently, let  $\mathbb{F}_{2^m}$  be some extension field of  $\mathbb{F}_2$ , the plain error locator polynomial as the monic generator  $g(z_1)$  of the principal ideal

$$\left\{ \sum_{i=1}^t g_i f_i, g_i \in \mathbb{F}_2(s_1, \dots, s_{2t-1})[z_1, \dots, z_t] \right\} \cap \mathbb{F}_2(s_1, \dots, s_{2t-1})[z_1],$$

---

T. Mora  
DISI, University of Genova, Genova, Italy  
e-mail: [theomora@disi.unige.it](mailto:theomora@disi.unige.it)

E. Orsini  
Department of Mathematics, University of Pisa, Pisa, Italy  
e-mail: [orsini@posso.dm.unipi.it](mailto:orsini@posso.dm.unipi.it)

which can be directly computed via the elimination property of lexicographical Gröbner bases.

In a series of papers Chen et al. (1994a, 1994b, 1994c) improved and generalized Cooper's approach to decoding. In particular, for a  $q$ -ary  $[n, k, d]$  cyclic codes, with correction capability  $t$ , they made the following alternative proposals:

1. denoting, for an error with weight  $\mu$ ,  $z_1, \dots, z_\mu$  the error locations,  $y_1, \dots, y_\mu$  the error values,  $s_1, \dots, s_{n-k} \in \mathbb{F}_{q^m}$  the associated syndromes, they interpreted (Chen et al. 1994c) the coefficients of the plain error locator polynomial as the elementary symmetric functions

$$\sigma_j(z_1, \dots, z_\mu) = (-1)^j \sum_{1 \leq l_1 \leq \dots \leq l_\mu \leq \mu} z_{l_1} \cdots z_{l_\mu}, \quad 1 \leq j \leq \mu,$$

and the syndromes as the *power sum functions*,  $s_i = \sum_{j=1}^\mu y_j z_j^i$ , and suggested to deduce the  $\sigma_j$ 's from the (known)  $s_i$ 's via a Gröbner basis computation of the ideal generated by the Newton identities;

2. they considered (Chen et al. 1994a) the *syndrome variety*

$$\left\{ \begin{array}{l} (s_1, \dots, s_{n-k}, y_1, \dots, y_t, z_1, \dots, z_t) \in (\mathbb{F}_{q^m})^{n-k+2t} : s_i \\ = \sum_{j=1}^\mu y_j z_j^i, \quad 1 \leq i \leq n-k \end{array} \right\}$$

and proposed to deduce via a Gröbner basis pre-computation in

$$\mathbb{F}_q[x_1, \dots, x_{n-k}, y_1, \dots, y_t, z_1, \dots, z_t]$$

a series of polynomials  $g_\mu(x_1, \dots, x_{n-k}, Z)$ ,  $\mu \leq t$  such that, for any error with weight  $\mu$  and associated syndromes  $s_1, \dots, s_{n-k} \in \mathbb{F}_{q^m}$ ,  $g_\mu(s_1, \dots, s_{n-k}, Z)$  in  $\mathbb{F}_{q^m}[Z]$  is the plain error locator polynomial.

Their suggestions were improved and refined in (respectively) Augot et al. (2003) and Caboara (2002), Loustaunau and York (1997); remark that

1. requires to perform for each received vector up to  $t$  Gröbner basis computations; the  $\mu$ -th computation deducing the unknown  $\sigma_1, \dots, \sigma_\mu$  in terms of the known syndromes  $s_1, \dots, s_n \in \mathbb{F}_{q^m}$ ;
2. requires a pre-computation of a Gröbner basis into a polynomial ring in  $2t+n-k$  variables.

Both computations are therefore not-necessarily feasible, the first since it requires an *on line* computation, the second since the syndrome variety has too many roots so that the Gröbner basis is less feasible to compute.

The investigation on the structure of the syndrome variety and on its Gröbner basis shows that most of its roots are spurious (Chen et al. 1994a) and that

the pre-computed polynomials  $g_\mu(x_1, \dots, x_n, Z)$  have the telescopic relations (Berlekamp 1968; Caboara 2002)

$$g_\mu = Zg_{\mu-1} + c(x_1, \dots, x_n).$$

To improve (Orsini and Sala 2005) the pre-computation it was sufficient to add equations removing the spurious roots. This new idea permitted to prove the existence of a computable *general error locator polynomial*  $\mathcal{L}$ , that is, a polynomial that satisfies the following property:

given a syndrome vector  $s \in (\mathbb{F}_{q^m})^{n-k}$  corresponding to an error with weight  $\mu \leq t$ , the  $t$  roots of  $\mathcal{L}$  (evaluated at  $s$ ) are the  $\mu$  error locations plus zero counted with multiplicity  $t - \mu$ .

This tutorial has the following structure. In the second section we present Cooper's idea of using Gröbner bases to decode binary BCH codes. In the third and fourth sections we describe Chen et al. ideas and we introduce the syndrome variety. The fifth section applies the Gianni–Kalkbrener Gröbner shape theorem to describe the structure of the syndrome variety. Section six introduces the general error locator polynomial for cyclic codes. Section seven is devoted to the *on line decoder* due to Augot et al. based on Newton's identities and Waring formulas.

## 2 Decoding Binary BCH Codes

We now describe the decoding algorithm proposed by Cooper (1990, 1991) to correct a primitive binary BCH codes of length  $n = 2^m - 1$ .

Let  $\alpha \in \mathbb{F}_{2^m}$  be a primitive  $n$ -th root of unity and  $C$  a primitive BCH code over  $\mathbb{F}_2$ , with defining set  $S = \{2i + 1, 0 \leq i < t\}$ . From the BCH bound we know that  $C$  can correct at least  $t$  errors.

We analyze the decoding process. Once the decoder receives a vector  $v \in (\mathbb{F}_2)^n$ , it computes the associated syndrome  $s \in (\mathbb{F}_{2^m})^{2t}$  and then uses it to find the unknown error locations  $\alpha^j$ . We introduce the variables  $Z = (z_1, \dots, z_t)$ , where  $z_j$  stands for the error location  $\alpha^j$ ,  $j = 1, \dots, t$ . Thus we obtain the following system of  $t$  polynomials in  $\mathbb{F}_{2^m}[Z]$ :

$$\mathcal{F}_C : \left\{ f_i : \sum_{j=1}^t z_j^{2i-1} - s_{2i-1}, \quad i = 1, \dots, t \right\}.$$

The error locations form a solution  $(\xi_1, \dots, \xi_t) \in (\mathbb{F}_{2^m})^t$  of  $\mathcal{F}_C$ . In this way an error correction procedure is a method of solving the nonlinear polynomial system  $\mathcal{F}_C$  for  $z_1, \dots, z_t$ . Sometimes finding this solution could be difficult and ineffective. Cooper's idea is to transform the system  $\mathcal{F}_C$  to another simpler system of equations having the same roots.

Let  $I$  be the ideal generated by  $\mathcal{F}_C$  in  $\mathbb{F}_{2^m}[Z]$  and  $\mathcal{V}(I)$  the set of its roots. Let  $G$  be the reduced Gröbner basis of  $I$  w.r.t. the lex ordering  $<$  induced by  $z_1 < \dots < z_t$ ; we denote by  $g \in \mathbb{F}_{2^m}[z_1]$  the unique polynomial such that  $G \cap \mathbb{F}_{2^m}[z_1] = \{g\}$ . To find the error locations, it is useful to define  $\mathsf{E}$  to be the set of error locations:

$$\mathsf{E} = \{\xi_1, \dots, \xi_\mu\} \quad (1)$$

and  $\mathsf{Z}$  the set of all components of the zeros of  $\mathcal{F}_C$ :

$$\mathsf{Z} = \{\xi \mid (\xi, a_2, \dots, a_t) \in \mathcal{V}(I)\}. \quad (2)$$

**Theorem 1** (Cooper 1991) *Let  $G$ ,  $I$  and  $g$  be as above. The following hold:*

- (a)  $\mathsf{E} = \mathsf{Z} = \{\xi \mid g(\xi) = 0\}$ ;
- (b)  $|\mathsf{E}| = \mu = \deg(g) \leq t$ ;
- (c)  $L_e(z) = g(z) = \prod_{\xi \in \mathsf{Z}} (z - \xi)$ , i.e.  $g$  is the polynomial whose roots are the error locators;
- (d)  $\sigma(z) = z^\mu g(z^{-1})$ , where  $\sigma$  is the classic error locator.

*Remark 1* There is in Cooper a designed ambiguity; the arithmetic is performed on the  $s_i$  in  $\mathbb{F}_2[s_i, i \in S]$  but are interpreted as performed on  $s_i = s_i(\alpha)$  in  $\mathbb{F}_{2^m}$ . All over this section we have deliberately maintained this ambiguity which will be solved in the next section; we have done so based on the interpretation of error locator polynomials suggested in Berlekamp (1968): an error locator polynomial is a cascade of devices, each evaluating a rational function  $a_l(s_i) \in \mathbb{F}_2(s_i)$  and connected by gates activated by the value of polynomials  $\beta(s_i) \in \mathbb{F}_2[s_i]$ ; at arrival of the word, the devices are properly connected, by evaluation of  $\beta(s_i) \in \mathbb{F}$  producing an expression  $\sum_{l=1}^\mu a_l(s_i)z \in \mathbb{F}_2(s_i)[z]$ , whose evaluation returns the error locator polynomial  $\sum_{l=1}^\mu a_l(s_i)z \in \mathbb{F}_2(s_i)[z]$ .

*Example 1* (Cooper 1991) Let  $C$  be a BCH code over  $\mathbb{F}_2$  and defining set  $S = \{1, 3\}$ . We want to find the classical error locator polynomial  $\sigma(z)$ . As  $t = 2$ , we set  $\mathcal{P} := \mathbb{F}_2[s_1, s_3][z_1, z_2]$ . Then

$$\mathsf{I} := \mathbb{I}(z_1 + z_2 + s_1, z_1^3 + z_2^3 + s_3) \subset \mathcal{P}$$

and the reduced Gröbner basis w.r.t. the lex ordering is

$$G = \{z_1^2 s_1 + z_1 s_1^2 + s_1^3 + s_3, z_2 + z_1 + s_1\}.$$

So  $g(z) = z^2 s_1 + z s_1^2 + s_1^3 + s_3$ , id est (cf. Berlekamp 1968, Example 5.6, pp. 138–139)

$$\sigma(z) = 1 + z s_1 + z^2 \left( \frac{s_1^3 + s_3}{s_1} \right).$$

*Example 2* (Cooper 1991) Let  $C$  be a BCH code over  $\mathbb{F}_2$ , defining set  $S = \{1, 3, 5\}$  and  $t = 3$ . As in the previous example we set  $\mathcal{P} := \mathbb{F}_2[s_1, s_3, s_5][z_1, z_2, z_3]$ . Then

$$\mathcal{I} := \mathbb{I}(z_1 + z_2 + z_3 + s_1, z_1^3 + z_2^3 + z_3^3 + s_3, z_1^5 + z_2^5 + z_3^5 + s_5) \subset \mathcal{P}$$

and the reduced Gröbner basis w.r.t. the lex ordering, with  $z_1 < z_2 < z_3$ , is

$$\begin{aligned} G = \{ &z_1^3 s_1^3 + z_1^3 s_3 + z_1^2 s_1^4 + z_1^2 s_1 s_3 + z_1 s_1^2 s_3 + z_1 s_5 + s_1^6 + s_1^3 s_3 + s_1 s_5 + s_3^2, \\ &z_2^2 s_1^3 + z_2^2 s_3 + z_2 z_1 s_1^3 + z_2 z_1 s_3 + z_2 s_1^4 + z_2 s_1 s_3 + z_1^2 s_1^3 + z_1^2 s_3 + z_1 s_1^4 \\ &+ z_1 s_1 s_3 + s_1^2 s_3 + s_5, \\ &z_2^2 z_1 + z_2^2 s_1 + z_2 z_1^2 + z_2 s_1^2 + z_1^2 s_1 + z_1 s_1^2 + s_1^3 + s_3, z_3 + z_2 + z_1 + s_1 \}, \end{aligned}$$

so that

$$g(z) = z^3(s_1^3 + s_3) + z^2(s_1^4 + s_1 s_3) + z(s_1^2 s_3 + s_5) + s_1^6 + s_1^3 s_3 + s_1 s_5 + s_3^2$$

$$\text{and } \sigma(z) = 1 + z s_1 + z^2 \left( \frac{s_1^2 s_3 + s_5}{s_1^3 + s_3} \right) + z^3 \left( \frac{s_1^6 + s_1^3 s_3 + s_1 s_5 + s_3^2}{s_1^3 + s_3} \right).$$

In the following example we perform decoding.

*Example 3* Let  $C$  be the binary BCH  $[15, 5, 7]$  code. This code has defining set  $\{1, 3, 5\}$ . If we set  $\beta_1 := s_1^3 + s_3$ ,  $\beta_2 := s_1^2 s_3 + s_5$ , and

$$\beta_3 := s_1^6 + s_1^3 s_3 + s_1 s_5 + s_3^2 = s_1 \beta_2 + \beta_1^2,$$

we obtain:

$$\sigma(z) = 1 + z s_1 + z^2 \beta_2 \beta_1^{-1} + z^3 \beta_3 \beta_1^{-1}.$$

- (i) Suppose that the error polynomial is  $e(x) = x^3$ . Obviously the decoder does not know the error polynomial, but it receives a vector in  $(\mathbb{F}_2)^{15}$  and it calculates the syndrome components, which in this case are:

$$s_1 = \alpha^3, s_3 = \alpha^9, s_5 = 1.$$

So  $\beta_1 = 0$ ,  $\beta_2 = 0$ ,  $\beta_3 = 0$  and  $\sigma(z) = 1 + z\alpha^3$ . The decoder correctly concludes that the error location is  $\alpha^3$ .

- (ii) If the error polynomial is  $e(x) = x^3 + x^2$ , the syndromes are:

$$s_1 = \alpha^6, s_3 = \alpha^5, s_5 = \alpha^5,$$

that is,  $\beta_1 = \alpha^{11}$ ,  $\beta_2 = \alpha$ ,  $\beta_3 = 0$  and

$$\sigma(z) = 1 + z\alpha^6 + z^2\alpha^5 = (1 + z\alpha^2)(1 + z\alpha^3).$$

(iii) Let  $e(x) = x^3 + x^2 + x$  be the error polynomial, then we have:

$$s_1 = \alpha^{11}, s_3 = \alpha^{11}, s_5 = 0,$$

that is,  $\beta_1 = \alpha^5, \beta_2 = \alpha^3, \beta_3 = \alpha^{11}$  and then

$$\sigma(z) = 1 + z\alpha^{11} + z^2\alpha^{13} + z^3\alpha^6 = (1 + z\alpha)(1 + z\alpha^2)(1 + z\alpha^3).$$

## 3 Gröbner Bases for Cyclic Codes

### 3.1 Decoding Binary Cyclic Codes

Chen et al. (1994c) generalize the Cooper's idea of using Gröbner techniques to decoding binary cyclic codes.

We consider a cyclic code  $C$  over  $\mathbb{F}_2$  with length  $n$  and defining set  $S$ . As usual we denote by  $\mu$  the number of errors which occurred and we name  $v$  an integer such that  $0 < v \leq t$  and  $\mu \leq v$ . Using the  $z_j$ 's variables for the error locations (which are  $n$ -th roots of unity), we can consider the following system where each syndrome  $s_i$  represents a value ( $s_i \in \mathbb{F}_{2^m}$ ):

$$\mathcal{F}_{\text{CRHT}_2} : \left\{ \left\{ \sum_{j=1}^v z_j^i - s_i, i \in S \right\} \cup \{z_j^{n+1} - z_j, 1 \leq j \leq v\} \right\} \subset \mathbb{F}_{2^m}[z_1, \dots, z_v].$$

Let  $E$  and  $Z$  be as in (1) and (2). The system  $\mathcal{F}_{\text{CRHT}_2}$  defines an ideal  $I = \mathbb{I}(\mathcal{F}_{\text{CRHT}_2})$  in  $\mathbb{F}_{2^m}[z_1, \dots, z_v]$ . The zero set of this ideal gives the error locations and, consequently, the error vector that occurred in the transmission. Gröbner basis computation can be used to find the solutions of this system.

Let  $G \subset \mathbb{F}_{2^m}[z_1, \dots, z_v]$  be the reduced Gröbner basis of  $I$  w.r.t. the lex ordering with  $z_1 < \dots < z_v$ , and  $g(z_1) \in \mathbb{F}_{2^m}[z_1]$  such that  $g(z_1) = G \cap \mathbb{F}_{2^m}[z_1]$ .

**Proposition 1** (Chen et al. 1994c) *We have:*

- (a)  $E \subseteq Z = \{\xi \mid g(\xi) = 0\}$ ;
- (b)  $|E| = \mu \leq v = \deg(g)$ .

Compare Theorem 1(a)–(b) and Proposition 1, which is a generalization of the previous. As regards (c) and (d) the following theorem describes the relation between  $g$  and the plain error locator polynomial  $L_e(z)$  in function of  $\mu$  (the weight of the error) and hence implies a decoding algorithm for any binary cyclic code up to its true minimum distance.

**Theorem 2** (Chen et al. 1994c) *We have:*

- (i) If  $v = \mu$  then  $\mathcal{V}(I)$  consists of all coordinate permutations of the root  $(\xi_1, \dots, \xi_\mu)$ ,  $\mathsf{E} = \mathbb{Z}$ ,  $L_e(z) = g(z)$  and  $\sigma(z) = z^\mu g(z^{-1})$ .
- (ii) If  $v = \mu + 1$  then  $(0, \xi_1, \dots, \xi_\mu) \in \mathcal{V}(I)$ ,  $\mathsf{E} = \mathbb{Z} \cup \{0\}$ , and  $g(z) = z(z^\mu \sigma(z^{-1})) = zL_e(z)$ .
- (iii) If  $v \geq \mu + 2$  then  $(\zeta, \zeta, \xi_1, \dots, \xi_\mu, 0, \dots, 0) \in \mathcal{V}(I)$ ,  $\forall \zeta \in \mathbb{F}_{2^m}$ ,  $\mathsf{E} = \mathbb{F}_{2^m}$  and  $g(z) = z^{n+1} - z$ .
- (iv) If  $v < \mu$  then  $G = \{1\}$ .

From this theorem we easily deduce a decoding algorithm for all binary cyclic codes. We will see some examples.

*Example 4* Let  $C$  be the binary cyclic code [21, 6, 7] with defining set  $S = \{1, 5, 9\}$ . The splitting field is  $\mathbb{F}_{2^6}$  and  $t = 3$ .

1. We first suppose that two errors occurred with the error polynomial  $e(x) = 1 + x$ . Obviously the decoder does not know  $\mu$  and  $e(x)$ , but it calculates the syndrome components, which are  $s_1 = 1 + \alpha$ ,  $s_5 = 1 + \alpha^5$  and  $s_9 = 1 + \alpha^9$ . We set  $v = 2$ . Then the associated polynomial system  $\mathcal{F}_{\text{CRHT}_2}$  is

$$\{z_1 + z_2 + (1 + \alpha), z_1^5 + z_2^5 + (1 + \alpha^5), z_1^9 + z_2^9 + (1 + \alpha^9), z_1^{22} - z_1, z_2^{22} - z_2\}$$

We obtain  $g(z) = z^2 + (1 + \alpha)z + \alpha = (z + 1)(z + \alpha) = L_e(z)$ .

2. Let  $e(x) = 1 + x + x^3$  be the error polynomial. The syndrome components are  $s_1 = 1 + \alpha + \alpha^3$ ,  $s_5 = 1 + \alpha^3 + \alpha^{15}$  and  $s_9 = 1 + \alpha^9 + \alpha^{27}$ . We set  $v = 2$ . Then  $\mathcal{F}_{\text{CRHT}_2}$  is

$$\{z_1 + z_2 + s_1, z_1^5 + z_2^5 + s_5, z_1^9 + z_2^9 + s_9, z_1^{22} - z_1, z_2^{22} - z_2\}$$

and the reduced Gröbner basis of  $I(\mathcal{F}_{\text{CRHT}_2})$  is  $G = \{1\}$ . So we set  $v = 3$ , the associated polynomial system  $\mathcal{F}_{\text{CRHT}_2}$  is

$$\{z_1 + z_2 + z_3 + s_1, z_1^5 + z_2^5 + z_3^5 + s_5, z_1^9 + z_2^9 + z_3^9 + s_9, z_1^{22} - z_1, z_2^{22} - z_2, z_3^{22} - z_3\}$$

and

$$\begin{aligned} g(z) &= z^3 + (\alpha^3 + \alpha + 1)z^2 + (\alpha^4 + \alpha^3 + \alpha)z + \alpha^4 = (z + 1)(z + \alpha)(z + \alpha^3) \\ &= L_e(z). \end{aligned}$$

### 3.2 Decoding Cyclic Codes over $\mathbb{F}_q$

Chen et al. (1994a) generalize Cooper's approach to  $q$ -adic codes proposing a solution for decoding an error whose weight  $\mu$  is assumed known. They also give an alternative approach via Newton's identities in the binary case.

If we consider a cyclic code over  $\mathbb{F}_q$ , we use the variables  $y = (y_1, \dots, y_\mu)$  for the error values. We suppose that we know the number of errors  $\mu$ . As before, our goal is to find the error locations and the corresponding error values from the known syndromes  $s_i \in \mathbb{F}_{q^m}$ ,  $i \in S_C$ . So we consider the polynomial system in  $\mathbb{F}_{q^m}[z_1, \dots, z_\mu, y_1, \dots, y_\mu]$ :

$$\mathcal{F}_{\text{CRHT}_q} : \left\{ \left\{ \sum_{j=1}^{\mu} y_j z_j^i - s_i, i \in S_C \right\} \cup \{z_j^{n+1} - z_j, 1 \leq j \leq \mu\} \cup t\{y_j^{q-1} - 1, 1 \leq j \leq \mu\} \right\}.$$

Let  $I$  be the ideal in  $\mathbb{F}_{q^m}[z_1, \dots, z_\mu, y_1, \dots, y_\mu]$  generated by  $\mathcal{F}_{\text{CRHT}_q}$ , and  $G$  the reduced Gröbner basis of  $I$  w.r.t. the lex ordering  $<$  induced by  $z_1 < \dots < z_\mu < y_1 < \dots < y_\mu$ . Then we generalize the definitions (1) and (2). Let  $\mathcal{V}(I) \subset \mathbb{F}^{2\mu}$  be the roots of  $I$ , we set

$$Z := \{\xi \mid (\xi, a_2, \dots, a_\mu, e_1, \dots, e_\mu) \in \mathcal{V}(I)\}, \quad E := \{\xi_1, \dots, \xi_\mu\}$$

the set of the error locations of an error with weight  $\mu$ .

**Theorem 3** (Chen et al. 1994a) *Let  $g$  be the monic polynomial in  $G \cap \mathbb{F}[x_1]$ . We have:*

- (a)  $E = Z = \{\xi \mid g(\xi) = 0\}$ ;
- (b)  $\#E = \mu = \deg(g) \leq t$ ;
- (c)  $L_e(z) = g(z) = \prod_{\xi \in Z} (z - \xi)$ ;
- (d)  $\sigma(z) = z^\mu g(z^{-1})$ .

### 3.3 A New System with the Newton Identities

Let  $\mathbb{F} := \mathbb{F}_2$ .

Denoting  $\sigma_j$ ,  $1 \leq j \leq \mu$ , the  $j$ -th elementary symmetric function on the  $z_i$ 's, the plain error locator polynomial is  $L_e(z) = 1 + \sum_{j=1}^{\mu} \sigma_j z^j$ . The second decoding scheme proposed in Chen et al. (1994a) is based on the relations among all syndromes  $s_i$ ,  $i = 1, \dots, n$ , and coefficients  $\sigma_j$  of  $L_e(z)$ , given by the following theorem.

**Theorem 4** (Newton identities) *Let  $s_i = \sum_{j=1}^{\mu} z_j^i$  (as in  $\mathcal{F}_{\text{CRHT}_2}$ ), then the following identities hold:*

$$\begin{cases} s_i + \sum_{j=1}^{i-1} \sigma_j s_{i-j} + i\sigma_i = 0 & 1 \leq i \leq \mu \\ s_i + \sum_{j=1}^{\mu} \sigma_j s_{i-j} = 0 & \mu < i < n. \end{cases} \quad (3)$$

*Remark 2* If  $2\mu \leq n$ , polynomial  $L_e(z)$  can be uniquely determined from (3).

We now need some more notation. We denote by  $R = \{\ell_1, \dots, \ell_r\}$  a set of representatives for the cyclotomic cosets of  $\{i, 1 \leq i \leq n, i \notin S\}$ . We use variables  $(T_1, \dots, T_\mu)$  and we set that  $T_i$  stands for  $\sigma_i$ ,  $1 \leq i \leq \mu$ , and variables  $(U_1, \dots, U_r)$  for  $(s_{\ell_1}, \dots, s_{\ell_r})$ . Then let  $\mathcal{P} := \mathbb{F}[T_1, \dots, T_\mu, U_1, \dots, U_r]$  and let  $\pi$  be the evaluation defined by

$$\pi : K[T_1, \dots, T_\mu, X_1, \dots, X_n] \longrightarrow \mathcal{P}, \quad \pi(X_i) := \begin{cases} s_i \in \mathbb{F} & i \in S_C \\ U_j^{2^\alpha} & i = 2^\alpha \ell_j \notin S_C \end{cases}$$

We consider the set  $\mathcal{F}_N$  of polynomials in  $\mathcal{P}$ :

$$\left\{ \pi \left( X_i + \sum_{j=1}^{\mu} T_j X_{i-j} \right), \mu < i < n \right\} \cup \{ U_j^{2^m} - U_j, 1 \leq j \leq r \} \\ \cup \{ T_l^{2^m} - T_l, 1 \leq l \leq \mu \}.$$

**Theorem 5** (Chen et al. 1994a) *For each  $l$ ,  $1 \leq l \leq \mu$ , let  $g_l \in \mathbb{F}[T_l]$  be the monic generator polynomial of  $I(\mathcal{F}_N) \cap \mathbb{F}[T_l]$ . Then  $g_l = T_l - \sigma_l$ .*

*Remark 3* Any  $g_l$  can be found in an appropriate Gröbner basis.

## 4 The CRHT Syndrome Variety

In the decoding algorithms presented up to now, we have to do, for any word to be decoded, a Gröbner basis computation with syndromes considered as parameters, which are calculated from the received word and substituted into the system. Moreover, different Gröbner basis computations must be performed for different potential error weights, until the true weight of the actual error is obtained.

In Chen et al. (1994b) a new method is described in which we calculate the Gröbner basis as a “preprocessing”, with the syndromes taken as variables  $x_i$ . In this way the system has more variables, but we have to calculate the Gröbner basis only once and then simply evaluate it at the actual syndromes each time a word is received.

We use the variables  $x, z$  and  $y$  with the usual meaning (syndromes, locations, values) and we consider system  $\mathcal{F}_{\text{CRHT}} \subset \mathbb{F}_q[x_1, \dots, x_{n-k}, z_t, \dots, z_1, y_1, \dots, y_t]$ :

$$\left\{ \left\{ \sum_{j=1}^t y_j z_j^i - x_i, i \in S \right\} \cup \{ z_j^{n+1} - z_j, 1 \leq j \leq t \} \cup \{ y_j^{q-1} - 1, 1 \leq j \leq t \} \right\}.$$

Let  $\mathcal{V}(I) \subset (\mathbb{F}_{q^m})^{2\mu}$  and  $G$  be the reduced Gröbner basis of  $I = \mathbb{I}(\mathcal{F}_{\text{CRHT}})$  w.r.t.  $\text{lex} <$  with  $x_1 < \dots < x_{n-k} < z_t < \dots < z_1 < y_1 < \dots < y_t$ .

**Table 1** CRHT decoding algorithm

---

$\mu := 1$
<b>While</b> $c_{\mu,0}(s_1, \dots, s_{n-k}) = 0$ <b>do</b> $\mu := \mu + 1$
$g := \gcd(g_\mu(s_1, \dots, s_{n-k}, 0, \dots, 0, z), z^n - 1)$
$\sigma(z) := z^\mu g(z^{-1})$

---

*Remark 4* The ideal  $I$  is zero-dimensional. From now on we refer to  $I$  as the *syndrome ideal* and to  $\mathcal{V}(I)$  as the *syndrome variety*.

The decoding algorithm presented in Chen et al. (1994b) is built on this claim: *the Gröbner basis  $G$  contains for each  $i$ ,  $1 \leq i \leq t$ , a single element*

$$g_i \in \mathbb{F}_q[x_1, \dots, x_{n-k}, z_t, \dots, z_{t-i+1}]$$

with positive degree in  $z_{t-i+1}$ .

*Remark 5* This claim is clearly not true, as shown in Loustaunau and York (1997).

**Theorem 6** (Chen et al. 1994b) *Let  $e$  be the error vector of weight  $\mu' \leq t$  and  $\mathbf{s} = (s_1, \dots, s_{n-k})$  the syndrome vector.*

*Under the assumption above and setting*

$$g_i(x_1, \dots, x_{n-k}, 0, \dots, 0, z_{t-i+1}) = \sum_{j=0}^{n_i} c_{i,j} z_{t-i+1}^j,$$

*we have that*

1. *The following conditions are equivalent:*
  - (a) *there are exactly  $\mu$  errors;*
  - (b)  $c_{1,0}(\mathbf{s}) = \dots = c_{t-\mu,0}(\mathbf{s}) = 0 \neq c_{t-\mu+1,0}(\mathbf{s});$
2.  $L_e(z) = \gcd(g_{t-\mu}(\mathbf{s}, 0, z), z^n - 1).$

From the theorem we directly design the decoding algorithm (see Table 1).

The proposed algorithm needs the assumption that the related Gröbner basis has a particular structure, but in Loustaunau and York (1997) Loustaunau and York remark that the CRHT assumption, in general, does not hold and they make a weak proposal to correct the CRHT algorithm. Moreover, they observe that the suggested Gröbner computation cannot be performed by the best software and hardware of the period (1997), therefore suggest to use the FGLM algorithm (the ideal is 0-dimensional). Their remark is particular significant, since the same software/hardware is able to compute Cooper's ideal Example 3 within 18 secs.

## 5 The Gianni–Kalkbrener Shape Theorem

The structure of the Gröbner basis of a zero-dimensional ideal has been deeply analyzed in Gianni (1989) and Kalkbrener (1989). Caboara (2002) gives a correct

and optimized version of the CRHT decoding algorithm, based on the Gianni–Kalkbrener Gröbner shape theorem.

Let  $\mathbb{F}$  be a field and  $\overline{\mathbb{F}}$  its algebraic closure. We set  $\mathcal{P} = \mathbb{F}[x_1, \dots, x_n]$ . For any  $f \in \mathcal{P}$ , we will denote by  $\mathbf{T}(f)$  the *leading term* of  $f$  (w.r.t. a fixed term ordering); and, for any set  $H \subset \mathcal{P}$ ,  $\mathbf{T}\{H\}$  denotes the set  $\{\mathbf{T}(h) \mid h \in H\}$ .

We will use the lexicographical ordering  $<$  induced by  $x_1 < \dots < x_n$ . In order to describe the structure of the Gröbner basis of an ideal, we need to consider  $\mathcal{P}$  also as univariate polynomials in the variable  $x_n$  with coefficients in the polynomial ring  $\mathbb{F}[x_1, \dots, x_{n-1}]$ . For any element  $f \in \mathcal{P}$  we have:

$$f = \sum_{k=0}^h b_k(x_1, \dots, x_{n-1}) x_n^k = Tp(f) + \dots + Lp(f) x_n^h,$$

where we will denote by  $Lp(f) = b_h(x_1, \dots, x_{n-1})$  the *leading polynomial* and by  $Tp(f) = b_0(x_1, \dots, x_{n-1})$  the *trailing polynomial* of  $f$ .

**Definition 1** Let  $I \subset \mathcal{P}$  be an ideal and  $d$  an integer such that  $d \leq n$ . The  $d$ -th elimination ideal  $I_d$  is the ideal of  $\mathbb{F}[x_1, \dots, x_d]$  defined by  $I_d = I \cap \mathbb{F}[x_1, \dots, x_d]$ .

We consider an ideal  $I \subset \mathcal{P}$  and we name  $\mathcal{V}(I_d) \subset \overline{\mathbb{F}}^d$  the set of the roots of  $I_d$ . Let  $G = \{g_1, \dots, g_s\}$  be a Gröbner basis of  $I \subset \mathcal{P}$  w.r.t.  $<$ , ordered so that  $\mathbf{T}(g_1) < \dots < \mathbf{T}(g_s)$ . For any  $\iota \leq n$ , let  $G_\iota$  be  $G \cap \mathbb{F}[x_1, \dots, x_\iota]$  and

$$\forall \ell \in \mathbb{N}, \quad G_{\iota\ell} := \{g \in G_\iota \setminus G_{\iota-1} \mid \deg_{x_\iota}(g) = \ell\},$$

so that each  $G_\iota$  can be decomposed into blocks of polynomials according to their degree with respect to the variable  $x_\iota$ :  $G_\iota = \sqcup_\ell G_{\iota\ell}$ . In this way, if  $g \in G_{\iota\ell}$ , we have:

- $g \in \mathbb{F}[x_1, \dots, x_{\iota-1}][x_\iota] \setminus \mathbb{F}[x_1, \dots, x_{\iota-1}]$ ;
- $\deg_{x_\iota}(g) = \ell$ , i.e.  $g = Lp(g)x_\iota^\ell + \dots + Tp(g)$ .

**Theorem 7** (Gianni 1989; Kalkbrener 1989) *Let  $\alpha := (a_1, \dots, a_d) \in \mathcal{V}(I_d)$  and  $\Phi_\alpha$  s.t.*

$$\begin{aligned} \Phi_\alpha : \mathcal{P} &\rightarrow \mathbb{F}[x_{d+1}, \dots, x_n] \\ f(x_1, \dots, x_n) &\mapsto f(\alpha, x_{d+1}, \dots, x_n). \end{aligned}$$

*Let  $\epsilon$  be the minimal value such that  $\Phi_\alpha(Lp(g_\epsilon)) \neq 0$  and  $j, \delta$  the values such that  $g_\epsilon \in G_{j\delta}$ . Then*

1.  $j = d + 1$ ;
2. for each  $g \in G_{\iota\ell}$ :
  - if  $\iota \leq d$  then  $\Phi_\alpha(g) = 0$ ;
  - if  $\iota = d + 1 = j$ ,  $\ell < \delta$  then  $\Phi_\alpha(g) = 0$ ;

3.  $\Phi_\alpha(g_\epsilon) = \gcd(\Phi_\alpha(g) : g \in G_{d+1}) \in \overline{\mathbb{F}}[x_{d+1}]$ ;
4. for each  $a \in \overline{\mathbb{F}}$ ;

$$(a_1, \dots, a_d, a) \in \mathcal{V}(I_{d+1}) \iff \Phi_\alpha(g_\epsilon)(a) = 0.$$

This theorem allows us to improve the CRHT–algorithm.

We use variables  $(x_1, \dots, x_{n-k}), (z_1, \dots, z_t)$  and  $(y_1, \dots, y_t)$  as in  $\mathcal{F}_{\text{CRHT}_q}$ , and we set

$$\mathcal{Q} := \mathbb{F}_q[x_1, \dots, x_{n-k}] \quad \text{and} \quad \mathcal{P} := \mathbb{F}_q[x_1, \dots, x_{n-k}, z_t, \dots, z_1, y_1, \dots, y_t].$$

Then we consider the following equations:

$$f_i := \sum_{l=1}^t y_l z_l^j - x_i, \quad h_j := z_j^{n+1} - z_j, \quad \lambda_j := y_j^q - 1, \quad \chi_i := x_i^{q^m} - x_i.$$

We obtain the polynomial equations system:

$$\mathcal{F}_{\text{CM}} = \{f_i, h_j, \lambda_j, \chi_i : 1 \leq j \leq t, 1 \leq i \leq n-k\} \subset \mathcal{P}.$$

*Remark 6* With respect to  $\mathcal{F}_{\text{CRHT}_q}$  this system adds the relations  $x_i^{q^m} = x_i$  satisfied by the syndromes. The role of the polynomials  $h_j, \lambda_j, \chi_j$ , is noteworthy, since they remove all the roots that are in algebraic extensions outside  $\mathbb{F}$  and they make the other roots simple. This means that the syndrome ideal  $I$ , which is a zero-dimensional ideal, is also radical.

Let  $G$  be the reduced Gröbner basis of the  $I$  w.r.t. the lex ordering  $<$  induced by  $x_1 < \dots < x_{n-k} < z_t < \dots < z_1 < y_1 < \dots < y_t$ . Let us then denote, for each  $\iota \leq n$  and each  $\ell \in \mathbb{N}$

$$G_\iota := G \cap \mathcal{Q}[z_t, \dots, z_1] \quad \text{and} \quad G_{\iota\ell} := \{g \in G_\iota \setminus G_{\iota+1} : \deg_{x_\iota}(g) = \ell\}.$$

Moreover, we enumerate each  $G_{\iota\ell}$  as

$$G_{\iota\ell} := \{g_{\iota\ell 1}, \dots, g_{\iota\ell j_{\iota\ell}}\}, \quad \mathbf{T}(g_{\iota\ell 1}) < \dots < \mathbf{T}(g_{\iota\ell j_{\iota\ell}}).$$

**Theorem 8** *With the above notation, we have:*

- if  $\ell < \iota$  then  $G_{\iota\ell} = \emptyset$ ;
- if  $\ell > \iota$  then  $\ell = n+1$ ,  $G_{\iota\ell} = \{z_\iota^{n+1} - z_\iota\}$ .

For each  $g \in G_u$ ,

$$Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0 \iff g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) \neq 0.$$

If the error has weight  $\mu$ , then, for each  $g \in G_u$ ,

1. if  $\iota < \mu$  then  $g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = 0$ ;

2. if  $\iota = \mu$  and  $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$  then

$$0 \neq g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) = z_\mu^\mu L_e(z_\mu);$$

3. if  $\iota = \mu + 1$  and  $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$  then

$$g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = z_\iota \cdot (z_\iota^\mu L_e(z_\iota));$$

4. if  $\iota > \mu + 1$  and  $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$  then

$$z_\iota \cdot (z_\iota^\mu L_e(z_\iota)) \mid g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota).$$

*Example 5* We consider the cyclic code [15, 5, 7] over  $\mathbb{F}_2$  and defining set {1, 3, 5}. The syndrome ideal  $I$  is generated by:

$$\begin{aligned} &\{z_1 + z_2 + z_3 + x_1, z_1^3 + z_2^3 + z_3^3 + x_3, z_1^5 + z_2^5 + z_3^5 + x_5, x_1^{16} + x_1, \\ &x_2^{16} + x_2, x_3^{16} + x_3, z_1^{16} + z_1, z_2^{16} + z_2, z_3^{16} + z_3\}. \end{aligned}$$

The relevant part of the reduced Gröbner basis of  $I$  is<sup>1</sup>

$$\begin{aligned} g_{3,3,1} = & z_3^3(\mathbf{x_2x_3^3 + x_2}) + z_3^2x_1x_2x_3^3 + z_3^2x_1x_2 + z_3x_1^{11}x_2^3 + z_3x_1^8x_2^4x_3^3 \\ & + z_3x_1^6x_2^3x_3 + z_3x_1^5x_2^{10} + z_3x_1^5x_2^5x_3^3 + z_3x_1^5x_3^3 + z_3x_1^4x_2^2x_3^2 + z_3x_1^3x_2^4x_3 \\ & + z_3x_1^2x_2^{11}x_3^3 + z_3x_1^2x_2^{11} + z_3x_1^2x_2^6x_3^3 + z_3x_1^2x_2^6 + z_3x_1x_2^8x_3^2 + z_3x_1x_2^3x_3^2 \\ & + z_3x_2^{10}x_3 + z_3x_2^5x_3 + z_3x_3 + x_1^{12}x_2^3 + x_1^8x_2x_3^2 + x_1^7x_2^8x_3 + x_1^7x_2^3x_3 \\ & + x_1^6x_2^{10} + x_1^6x_3^3 + x_1^5x_2^{12}x_3^2 + x_1^4x_2^9x_3 + x_1^3x_2^{11} + x_1^3x_2^6x_3^3 + x_1^3x_2^6 \\ & + x_1^3x_2x_3^3 + x_1^3x_2 + x_1^2x_2^{13}x_3^2 + x_1x_2^{15}x_3 + x_1x_2^{10}x_3 + x_1x_3 \\ & + x_2^{12}x_3^3 + x_2^7x_3^3 + x_2^2, \end{aligned}$$

$$\begin{aligned} g_{3,3,2} = & z_3^3(\mathbf{x_2^5 + x_3^3}) + z_3^2x_1x_2^5 + z_3^2x_1x_3^3 + z_3x_1^{11}x_2^2 + z_3x_1^8x_2^{13}x_3^3 + z_3x_1^8x_2^8 \\ & + z_3x_1^8x_2^3 + z_3x_1^7x_2^5x_3^2 + z_3x_1^6x_2^7x_3 + z_3x_1^5x_2^{14}x_3^3 + z_3x_1^5x_2^9x_3^3 + z_3x_1^5x_2^9 \\ & + z_3x_1^4x_2x_3^2 + z_3x_1^3x_2^{13}x_3 + z_3x_1^2x_2^{10}x_3^3 + z_3x_1^2x_2^5x_3^3 + z_3x_1^2x_2^5 + z_3x_1^2 \\ & + z_3x_1x_2^{12}x_3^2 + z_3x_1x_2^7x_3^2 + z_3x_2^9x_3 + x_1^{12}x_2^2 + x_1^8x_2^5x_3^2 + x_1^7x_2^{12}x_3 \\ & + x_1^7x_2^2x_3 + x_1^6x_2^9x_3^3 + x_1^6x_2^4 + x_1^5x_2x_3^2 + x_1^4x_2^{13}x_3 + x_1^3x_2^{15} \\ & + x_1^3x_2^{10}x_3^3 + x_1^3x_2^{10} + x_1^3x_3^3 + x_1^3 + x_1^2x_2^2x_3^2 + x_1x_2^9x_3 + x_2^{11}, \end{aligned}$$

$$g_{3,3,3} = z_3^3(\mathbf{x_1 + x_2x_3^2}) + z_3^2x_1^2 + z_3^2x_1x_2^2x_3^2 + z_3x_1^{12}x_2^2 + z_3x_1^8x_2^5x_3^2 + z_3x_1^8x_3^2$$

---

<sup>1</sup>The **bold** polynomials are the leading polynomials, the **typewriter** ones are the trailing polynomials.

$$\begin{aligned}
& + z_3x_1^7x_2^{12}x_3 + z_3x_1^7x_2^7x_3 + z_3x_1^7x_2^2x_3 + z_3x_1^6x_2^{14}x_3^3 + z_3x_1^6x_2^9x_3^3 + z_3x_1^6x_2^9 \\
& + z_3x_1^5x_2^{11}x_3^2 + z_3x_1^5x_2x_3^2 + z_3x_1^4x_2^{13}x_3 + z_3x_1^4x_2^8x_3 + z_3x_1^4x_2^3x_3 \\
& + z_3x_1^3x_2^{10}x_3^3 + z_3x_1^3x_2^{10} + z_3x_1^3x_2^5x_3^3 + z_3x_1^3x_2^5 + z_3x_1^2x_2^{12}x_3^2 + z_3x_1x_2^4x_3 \\
& + z_3x_2^{11}x_3^3 + z_3x_2^6x_3^3 + x_1^{10}x_2^3 + x_1^8x_2^{12}x_3 + x_1^8x_2^7x_3 + x_1^7x_2^4x_3^3 \\
& + x_1^6x_2^{11}x_3^2 + x_1^6x_2^6x_3^2 + x_1^5x_2^8x_3 + x_1^5x_2^3x_3 + x_1^4x_2^{15} + x_1^4x_2^{10} \\
& + x_1^4x_2^5x_3^3 + x_1^3x_2^7x_3^2 + x_1^2x_2^4x_3 + x_1x_2^{11} + x_1x_2^6x_3^3 + x_1x_2^6 \\
& + x_2^{13}x_3^2 + x_2^8x_3^2,
\end{aligned}$$

$$g_{3 \ 16 \ 1} = z_3^{16} + z_3,$$

$$\begin{aligned}
g_{2 \ 2 \ 1} &= z_2^2(\mathbf{x}_2\mathbf{x}_3^3 + \mathbf{x}_2) + z_2z_3(\mathbf{x}_2\mathbf{x}_3^3 + \mathbf{x}_2) + z_2x_1(\mathbf{x}_2\mathbf{x}_3^3 + \mathbf{x}_2) + z_3^2x_2x_3^3 + z_3^2x_2 \\
& + z_3x_1x_2x_3^3 + z_3x_1x_2 + x_1^{11}x_2^3 + x_1^8x_2^4x_3^3 + x_1^7x_2x_3^2 + x_1^6x_2^3x_3 \\
& + x_1^5x_2^{10} + x_1^5x_2^5x_3^3 + x_1^5x_2^3 + x_1^4x_2^2x_3^2 + x_1^3x_2^4x_3 + x_1^2x_2^{11}x_3^3 + x_1^2x_2^{11} \\
& + x_1^2x_2^6x_3^3 + x_1^2x_2^6 + x_1x_2^8x_3^2 + x_1x_2^3x_2^2 + x_2^{10}x_3 + x_2^5x_3 + x_3, \\
g_{2 \ 2 \ 2} &= z_2^2(\mathbf{x}_2^5 + \mathbf{x}_3^3) + z_2z_3(\mathbf{x}_2^5 + \mathbf{x}_3^3) + z_2x_1(\mathbf{x}_2^5 + \mathbf{x}_3^3) + z_3^2x_2^5 + z_3^2x_3^3 + z_3x_1x_2^5 \\
& + z_3x_1x_3^3 + x_1^{11}x_2^2 + x_1^8x_2^{13}x_3^3 + x_1^8x_2^8 + x_1^8x_2^3 + x_1^7x_2^5x_3^2 + x_1^5x_2^9x_3 \\
& + x_1^5x_2^9 + x_1^4x_2x_3^2 + x_1^3x_2^{13}x_3 + x_1^2x_2^{10}x_3^3 + x_1^2x_2^5x_3^3 + x_1^2x_2^5 + x_1^2 \\
& + x_1x_2^{12}x_3^2 + x_1x_2^7x_3^2 + x_2^9x_3, \\
g_{2 \ 2 \ 3} &= z_2^2(\mathbf{x}_1 + \mathbf{x}_2^2\mathbf{x}_3^2) + z_2z_3(\mathbf{x}_1 + \mathbf{x}_2^2\mathbf{x}_3^2) + z_2x_1(\mathbf{x}_1 + \mathbf{x}_2^2\mathbf{x}_3^2) + z_3^2x_1 + z_3^2x_2^2x_3^2 \\
& + z_3x_1^2 + z_3x_1x_2^2x_3^2 + x_1^{12}x_2^2 + x_1^8x_2^5x_3^2 + x_1^8x_2^2 + x_1^7x_2^{12}x_3 + x_1^7x_2^7x_3 \\
& + x_1^7x_2^2x_3 + x_1^6x_2^{14}x_3^3 + x_1^6x_2^9x_3^3 + x_1^6x_2^9 + x_1^5x_2^{11}x_3^2 + x_1^5x_2x_3^2 \\
& + x_1^4x_2^{13}x_3 + x_1^4x_2^8x_3 + x_1^4x_2^3x_3 + x_1^3x_2^{10}x_3^3 + x_1^3x_2^{10} + x_1^3x_2^5x_3^3 \\
& + x_1^3x_2^5 + x_1^2x_2^{12}x_3^2 + x_1x_2^4x_3 + x_2^{11}x_3^3 + x_2^6x_3^3,
\end{aligned}$$

$$\begin{aligned}
g_{2 \ 2 \ 4} &= z_2^2(\mathbf{z}_3 + \mathbf{x}_2^2\mathbf{x}_3^2) + z_2z_3(\mathbf{z}_3 + \mathbf{x}_2^2\mathbf{x}_3^2) + z_2x_1(\mathbf{z}_3 + \mathbf{x}_2^2\mathbf{x}_3^2) + z_3^2x_2^2x_3^2 + z_3x_1x_2^2x_3^2 \\
& + x_1^{12}x_2^2 + x_1^8x_2^5x_3^2 + x_1^8x_2^2 + x_1^7x_2^{12}x_3 + x_1^7x_2^7x_3 + x_1^7x_2^2x_3 \\
& + x_1^6x_2^{14}x_3^3 + x_1^6x_2^9x_3^3 + x_1^6x_2^9 + x_1^5x_2^{11}x_3^2 + x_1^5x_2x_3^2 + x_1^4x_2^{13}x_3 \\
& + x_1^4x_2^8x_3 + x_1^4x_2^3x_3 + x_1^3x_2^{10}x_3^3 + x_1^3x_2^{10} + x_1^3x_2^5x_3^3 + x_1^3x_2^5 + x_1^3 \\
& + x_1^2x_2^{12}x_3^2 + x_1x_2^4x_3 + x_2^{11}x_3^3 + x_2^6x_3^3 + x_2,
\end{aligned}$$

$$g_{2 \ 16 \ 1} = z_2^{16} + z_2,$$

$$g_{1 \ 1 \ 1} = z_1 + z_2 + z_3 + x_1,$$

that we can rewrite compactly as

$$\begin{aligned}
 g_{3\ 3\ 1} &= z_3^3(\mathbf{x}_2\mathbf{x}_3^3 + \mathbf{x}_2) + \cdots + A \\
 g_{3\ 3\ 2} &= z_3^3(\mathbf{x}_2^5 + \mathbf{x}_3^3) + \cdots + B \\
 g_{3\ 3\ 3} &= z_3^3(\mathbf{x}_1 + \mathbf{x}_2^2\mathbf{x}_3^2) + \cdots + C \\
 g_{3\ 16\ 1} &= z_3^{16} + z_3, \\
 g_{2\ 2\ 1} &= z_2^2(\mathbf{x}_2\mathbf{x}_3^3 + \mathbf{x}_2) + \cdots + D \\
 g_{2\ 2\ 2} &= z_2^2(\mathbf{x}_2^5 + \mathbf{x}_3^3) + \cdots + E \\
 g_{2\ 2\ 3} &= z_2^2(\mathbf{x}_1 + \mathbf{x}_2^2\mathbf{x}_3^2) + \cdots + F \\
 g_{2\ 2\ 4} &= z_2^2(\mathbf{z}_3 + \mathbf{x}_2^2\mathbf{x}_3^2) + \cdots + G \\
 g_{2\ 16\ 1} &= z_2^{16} + z_2, \quad g_{1\ 1\ 1} = z_1 + z_2 + z_3 + x_1.
 \end{aligned}$$

If we restrict our attention to the leading polynomials we note that  $Lp(g_{3\ 3\ 1}) = Lp(g_{2\ 2\ 1})$ ,  $Lp(g_{3\ 3\ 2}) = Lp(g_{2\ 2\ 2})$  and  $Lp(g_{3\ 3\ 3}) = Lp(g_{2\ 2\ 3})$ . Moreover we can observe a telescopic behavior, namely:

$$\begin{aligned}
 g_{3\ 3\ 1}(x_1, x_2, x_3, z_3) &= z_3 g_{2\ 2\ 1}(x_1, x_2, x_3, 0, z_3) + Tp(g_{3\ 3\ 1})(x_1, x_2, x_3), \\
 g_{3\ 3\ 2}(x_1, x_2, x_3, z_3) &= z_3 g_{2\ 2\ 2}(x_1, x_2, x_3, 0, z_3) + Tp(g_{3\ 3\ 2})(x_1, x_2, x_3), \\
 g_{3\ 3\ 3}(x_1, x_2, x_3, z_3) &= z_3 g_{2\ 2\ 3}(x_1, x_2, x_3, 0, z_3) + Tp(g_{3\ 3\ 3})(x_1, x_2, x_3), \\
 g_{2\ 2\ *}(*, x_1, x_2, x_3, 0, z_2) &= z_2 Lp(g_{2\ 2\ *})(z_2 + x_1) + Tp(g_{2\ 2\ *})(x_1, x_2, x_3).
 \end{aligned}$$

We conclude this section with the algorithm proposed in Caboara (2002), which accepts as input a syndrome vector and outputs an error locator polynomial (see Table 2).

The decoder performs the following branching:

$$\begin{array}{lll}
 s_2 s_3^3 + s_2 \neq 0 \ A \neq 0 & & \rightarrow g_{3\ 3\ 1} \\
 A = 0 \quad D \neq 0 & & \rightarrow g_{2\ 2\ 1} \\
 D = 0 & & \rightarrow g_{1\ 1\ 1} \\
 s_2 s_3^3 + s_2 = 0 \ s_2^5 + s_3^3 \neq 0 \ B \neq 0 & & \rightarrow g_{3\ 3\ 2} \\
 B = 0 \quad E \neq 0 & & \rightarrow g_{2\ 2\ 2} \\
 E = 0 & & \rightarrow g_{1\ 1\ 1} \\
 s_2^5 + s_3^3 = 0 \ s_1 + s_2^2 s_3^2 \neq 0 \ C \neq 0 & & \rightarrow g_{3\ 3\ 3} \\
 C = 0 \quad F \neq 0 & & \rightarrow g_{2\ 2\ 3} \\
 F = 0 & & \rightarrow g_{1\ 1\ 1} \\
 s_1 + s_2^2 s_3^2 = 0 \ s_2^2 s_3^2 \neq 0 \ G \neq 0 & & \rightarrow g_{2\ 2\ 4} \\
 G = 0 \ s_1 \neq 0 \rightarrow g_{1\ 1\ 1} \\
 s_1 = 0 \rightarrow 1 & & \\
 s_2^2 s_3^2 = 0 \ s_1 \neq 0 & & \rightarrow g_{1\ 1\ 1} \\
 s_1 = 0 & & \rightarrow 1
 \end{array}$$

**Table 2** Caboara decoding algorithm

<b>Input</b> $\mu := t, g := 1,$
<b>Repeat</b>
$j := 0$
<b>Repeat</b> $j := j + 1$
<b>Until</b> $Lp(g_{\mu\mu j})(s, 0) \neq 0$ <b>or</b> $j > j_{\mu\mu}$
<b>if</b> $j > j_{\mu\mu}$ <b>then</b> $\mu := \mu - 1$ <b>else</b>
<b>if</b> $Tp(g_{\mu\mu j})(s, 0) = 0$ <b>do</b> $\mu := \mu - 1$
<b>else</b> $g(z) := g_{\mu\mu j}(s, 0, z);$
<b>Until</b> $g \neq 1$ <b>or</b> $\mu = 0$
<b>Output</b> $\mu, x^\mu g(x^{-1})$

*Remark 7* (Caboara 2002) reports also a proposal (suggested by M. Sala) of computing and processing, for each  $\mu$ ,  $1 \leq \mu \leq t$ , the Gröbner basis of the ideal, encoding only the case in which there are *exactly*  $\mu$  errors and performing a post-processing using Gröbner technology in order to improve the syndrome test. The result (still for  $\{1, 3, 5\}$ ) is a very promising decision tree:

$$\begin{aligned}
 s_2 = 0 & \quad s_3 = 0 \quad \implies L = 1 \\
 s_2 = 0 & \quad s_3 \neq 0 \quad \implies L = 1 + zs_1 + z^2s_1^2 \\
 s_2^5 + 1 = 0 & \quad s_3 = 0 \quad s_1 = 0 \quad \implies L = 1 + z^3s_2 \\
 s_2^5 + 1 = 0 & \quad s_3 = 0 \quad s_1 \neq 0 \quad \implies L = 1 + zs_1 \\
 & \quad \quad \quad + z^2(s_1^{11}s_2^2s_1^5s_2^4) + z^3s_1^9s_2^3 \\
 s_2^5 + 1 = 0 & \quad s_3 \neq 0 \quad \sigma = 0 \quad \implies L = 1 + zs_1 \\
 s_2^5 + 1 = 0 & \quad s_3 \neq 0 \quad \sigma \neq 0 \quad \implies L = 1 + zs_1 \\
 & \quad \quad \quad + z^2(s_1^2 + s_2^4s_3^4)(s_1^5s_3^2 + \sigma^{-1}) \\
 & \quad \quad \quad + z^3s_1^2s_2^2s_3(s_1^5 + s_2^{10}s_3^{10})\sigma^{-1} \\
 s_2^6 + s_2 \neq 0 & \quad s_3 = 0 \quad \implies L = 1 + zs_1 + z^2s_1^2s_2^5 \\
 s_2^6 + s_2 \neq 0 & \quad s_3^3 \neq 0 \quad s_1 = 0 \quad \implies L = 1 + z^2s_2^{-1}s_3 + z^3s_2 \\
 s_2^6 + s_2 \neq 0 & \quad s_3^3 \neq 0 \quad \rho = 0 \quad \implies L = 1 + zs_1 + z^2s_2^9s_3 \\
 s_2^6 + s_2 \neq 0 & \quad s_3^3 \neq 0 \quad s_1\rho \neq 0 \quad \implies L = 1 + zs_1 \\
 & \quad \quad \quad + z^2(s_1^5s_2^8s_3 + s_1^3s_2^2s_3^2)\rho^{-1} \\
 & \quad \quad \quad + z^2(s_1^2s_2^9s_3 + s_2^{13}s_3^2)\rho^{-1} \\
 & \quad \quad \quad + z^3s_1^4s_2^3s_3 \\
 & \quad \quad \quad + z^3s_1^3s_2^5 + s_1s_2^{-1}s_3 + s_2^{-4},
 \end{aligned}$$

$$\begin{aligned}
 \text{where } \rho &:= s_1^2 + s_1s_2^2s_3^2 + s_2^{-1}s_3 \\
 \sigma &:= s_1 + s_2^2s_3^2.
 \end{aligned}$$

## 6 The General Error Locator Polynomial

If we consider the syndrome variety  $\mathcal{V}(\mathcal{F}_{CM})$ , then we have that, for any given correctable syndrome  $\mathbf{s} \in (\mathbb{F}_{q^m})^{n-k}$ , there are some points in  $\mathcal{V}(\mathcal{F}_{CM})$  that uniquely determine the error locations and the error values. Unfortunately, in  $\mathcal{V}(\mathcal{F}_{CM})$  there are also other points that do not correspond directly to error vectors. Such points are of type:

$$(\xi_1, \dots, \xi_\mu, \zeta, \zeta, 0, \underbrace{0, \dots, 0}_{t-(\mu+2)}, \bar{y}_1, \dots, \bar{y}_\mu, Y, -Y, y_1, \dots, y_{t-(\mu+2)}),$$

with  $\zeta$  any  $n$ -th root of unity,  $Y, y_j$  arbitrary elements in  $\mathbb{F}_q$  and  $\bar{y}_j$  in  $\mathbb{F}_q$  the error values corresponding to the error locations  $\xi_j$ . In Orsini and Sala (2005) a new syndrome variety is proposed, which permits to eliminate these spurious solutions and to define the general error locator polynomial.

We consider  $[n, k, d]$  cyclic codes over  $\mathbb{F}_q$ , with  $(q, n) = 1$ . We need the following definition.

**Definition 2** Let  $n \in \mathbb{N}$  be an integer. We denote by  $p_{l\tilde{l}} \in K[z_1, \dots, z_t]$  the polynomial:

$$p_{l\tilde{l}} := \frac{z_l^n - z_{\tilde{l}}^n}{z_l - z_{\tilde{l}}}, \quad 1 \leq l < \tilde{l} \leq t.$$

We consider a new syndrome ideal  $I = \mathbb{I}(V(\mathcal{F}_{OS}))$ , the  $\mathcal{OS}$  ideal, as:

$$\mathcal{F}_{OS} = \{f_i, h_j, \chi_i, \lambda_j, p_{l\tilde{l}}, 1 \leq l < \tilde{l} \leq t, 1 \leq i \leq n-k, j \in S\} \subset \mathcal{P},$$

where

$$\begin{aligned} f_i &:= \sum_{l=1}^t y_l z_l^i - x_i, \quad p_{l\tilde{l}} := z_{\tilde{l}} z_l p_{l\tilde{l}}, \\ h_j &:= z_j^{n+1} - z_j, \quad \lambda_j := y_j^{q-1} - 1, \quad \chi_i := x_i^{q^m} - x_i. \end{aligned}$$

Let  $G$  be the reduced Gröbner basis of  $I$  w.r.t. the lex ordering with  $x_1 < \dots < x_{n-k} < z_t < \dots < z_1 < y_1 < \dots < y_t$ . We have

**Theorem 9** (Orsini and Sala 2005) *Let  $I$  and  $G$  be as above. Then:*

1.  $G \cap \mathcal{Q}[z_1, \dots, z_t] = \bigcup_{i=1}^t G_i$ ;
2.  $G_i = \bigcup_{\delta=1}^i G_{i\delta}$  and  $G_{i\delta} \neq \emptyset$ ,  $1 \leq i \leq t$  and  $1 \leq \delta \leq i$ ;
3.  $G_{ii} = \{g_{ii1}\}$ ,  $1 \leq i \leq t$ , i.e. exactly one polynomial exists with degree  $i$  w.r.t. the variable  $z_i$  in  $G_i$ ;
4.  $Lt(g_{ii1}) = z_i^i$ ,  $Lp(g_{ii1}) = 1$ ;
5. if  $1 \leq i \leq t$  and  $1 \leq \delta \leq i-1$ , then  $\forall g \in G_{i\delta}$ ,  $Tp(g) = 0$ .

Let  $g_{tt1}$  be the unique polynomial with degree  $t$  w.r.t. variable  $z_t$  in  $G_t$ :

$$g_{tt1} = z_t^t + \sum_{l=1}^t a_{t-l} z_t^{t-l}.$$

The following properties are equivalent:

- there are exactly  $\mu$  errors;
- $a_{t-l}(s) = 0$  for  $l > \mu$  and  $a_{t-\mu}(s) \neq 0$ ;
- $g_{tt1}(s, z_t) = z^{t-\mu}(L_e(z))$ ;

and imply that  $\sigma(z) = z^\mu g_{tt1}(s, z^{-1})$ . This means that  $g_{tt1}$  is a monic polynomial in  $\mathbb{Q}[z]$  which satisfies the following property:

given a syndrome vector  $s = (s_1, \dots, s_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$  corresponding to an error with weight  $\mu \leq t$ , then its  $t$  roots are the  $\mu$  error locations plus zero counted with multiplicity  $t - \mu$ ,

and is called a *general error locator polynomial* of  $C$ .

**Theorem 10** (Orsini and Sala 2005) *Every cyclic code possesses a general error locator polynomial.*

Once we have computed a general error locator polynomial for the code  $C$ , the decoding algorithm is straightforward (see Table 3).

*Remark 8* The existence of such polynomials for larger class of linear codes is proved in Giorgetti and Sala (2006, 2009).

*Example 6* We consider the cyclic code of Example 5 with the  $\mathcal{OS}$  syndrome ideal. The result is already relatively small

$$\begin{aligned} g_{331} = & \mathbf{z}_3^3 + z_3^2 x_1 + z_3(x_3 x_2^9 + x_3 x_2^8 x_1^3 + x_3 x_2^4 + x_3 x_2 x_1^9) \\ & + z_3(x_2^{15} x_1^2 + x_2^{14} x_1^5 + x_2^{13} x_1^8 + x_2^{12} x_1^{11} + x_2^{11} x_1^{14}) \\ & + z_3(x_2^{10} x_1^2 + x_2^7 x_1^{11} + x_2^6 x_1^{14} + x_2^5 x_1^2 + x_2^3 x_1^8 + x_2^2 x_1^{11} + x_1^2) \\ & + x_3 x_2^9 x_1 + x_3 x_2^8 x_1^4 + x_3 x_2^4 x_1 + x_3 x_2 x_1^{10} + x_2^{15} x_1^3 + x_2^{14} x_1^6 + x_2^{13} x_1^9 \\ & + x_2^{12} x_1^{12} + x_2^{11} x_1^{15} + x_2^{10} x_1^3 + x_2^7 x_1^{12} + x_2^6 x_1^{15} + x_2^5 x_1^3 + x_2^3 x_1^9 + x_2^2 x_1^{12} + x_2 \end{aligned}$$

**Table 3** Orsini–Sala decoding algorithm

---

<b>Input</b> $\mathbf{s} = (s_1, \dots, s_{n-k})$
$\mu = t$
<b>While</b> $a_{t-\mu}(s_1, \dots, s_{n-k}) = 0$ <b>do</b>
$\mu := \mu - 1$ ;
<b>Output</b> $\mu, L_e(z)$

---

but clever guessing inspired by eye-inspection gives a more compact presentation

$$g_{331} = A^3 + AE + B$$

where

$$\begin{aligned} A &:= x_1 + z_3, & B &:= x_2 + x_1^3, & C &:= x_3 + x_1^5, \\ D &:= x_2^8 + x_2^7x_1^3 + x_2^3 + x_1^9, & E &:= x_1^2(B^{15} - 1) - Cx_2D. \end{aligned}$$

The efficiency of this algorithm obviously depends on the sparsity of the general error locator polynomial. Even if at present there is no known theoretical proof of the sparsity of general error locator polynomials, there is some experimental evidence, at least in the binary case. In Mora et al. (2006) and Orsini and Sala (2007) it is shown that this algorithm may be applied efficiently to all binary cyclic code with  $t \leq 2$  and length  $n$  less than 63, as we now detail. Recalling that the following trivial theorem holds for each binary cyclic codes with  $t \leq 2$ .

**Theorem 11** *Let  $C$  be a code with  $t = 1$  and  $\mathbf{s}$  a correctable syndrome, then the general error locator polynomial is  $\mathcal{L}_C(X, z) = z + a$ , where  $a \in \mathbb{F}_2[X]$ . Moreover, there is one error if and only if  $a(\mathbf{s}) \neq 0$  and in that case the error location is  $a(\mathbf{s})$ .*

*Let  $C$  be a code with  $t = 2$ ,  $\mathbf{s}$  a correctable syndrome and  $\bar{z}_1$  and  $\bar{z}_2$  the error locations. Then  $\mathcal{L}_C(X, z) = z^2 + az + b$ , where  $a, b \in \mathbb{F}_2[X]$ , and  $b(\mathbf{s}) = \bar{z}_1\bar{z}_2$ ,  $a(\mathbf{s}) = \bar{z}_1 + \bar{z}_2$ . Moreover, there are two errors if and only if  $b(\mathbf{s}) \neq 0$ , and there is an error if and only if  $b(\mathbf{s}) = 0$  and  $a(\mathbf{s}) \neq 0$ .*

Let us now state the main theorems of Mora et al. (2006):

**Theorem 12** *Let  $C$  be a binary  $[n, k, d]$  code with  $n \leq 61$  and  $d = 3, 4$  [ $t = 1$ ]. We denote by  $S$  a defining set of  $C$  and  $\mathcal{L}_C \in \mathbb{F}_q[x_1, \dots, x_{n-k}][z]$  a general error locator polynomial. Then there are only four cases:*

- (1)  *$C$  has a defining set of type  $S = \{m\}$ , with  $(n, m) = 1$ . Then there exists an integer  $k$  modulo  $n$  such that  $\mathcal{L}_C = z + x_1^k$ .*
- (2)  *$C$  has a defining set of type  $S = \{m, h\}$ , with  $(m, h) = 1$ . Then there exist two integers  $m'$  and  $h'$  modulo  $n$  such that*

$$\mathcal{L}_C = z + x_1^{m'} x_2^{h'}.$$

- (3)  *$C$  is a sub-code of a code  $C'$  of type (1) or (2) and  $\mathcal{L}_C = \mathcal{L}_{C'}$ .*
- (4)  *$C$  is equivalent to a code  $C'$  of type (1), (2) or (3) and  $\mathcal{L}_C$  can be trivially obtained from  $\mathcal{L}_{C'}$ .*

The following theorem shows an interesting property for a wide class of 2-error correcting codes.

**Theorem 13** *Let  $C$  be a code with length  $3 \leq n \leq 125$  ( $n \neq 105$ ) and distance  $d = 5, 6$ . Then  $C$  is equivalent to a code  $D$  s.t.  $1 \in S$ .*

From this it is easy to prove that if  $C$  is a binary  $[n, k, d]$  code with  $7 \leq n < 63$  ( $n$  odd) and  $d = 5, 6$ , then

$$\mathcal{L}_C = z^2 + x_1 z + b(x_1, \dots, x_{n-k}),$$

where  $b(x_1, \dots, x_{n-k}) \subset \mathbb{F}_2[x_1, \dots, x_{n-k}]$ .

**Theorem 14** *Let  $C$  be a binary  $[n, k, d]$ -code with  $7 \leq n < 63$  ( $n$  odd) and  $d = 5, 6$ ,  $[t = 2]$ . Then there are seven cases:*

1.  $n$  is such that the code with defining set  $\{0, 1\}$  has distance  $d \geq 5$ ;
2.  $C$  is a BCH code, i.e.  $S = \{1, 3\}$  and

$$b = x_1^{n-1} (x_1^3 + x_2);$$

3.  $C$  admits a defining set  $S = \{1, n-1, l\}$ , with  $l = 0, n/3$ , and

$$b = \begin{cases} x_1 x_2^{-1} (1 + x_3) & l = 0 \\ \frac{x_2^3 + 1}{x_1^{n/3} x_2^{2/3n} x_3 + 1} & l = n/3; \end{cases}$$

4.  $C$  admits a defining set  $S = \{1, n/l\}$ , for some  $l \geq 3$ ;
5.  $C$  is one of the following:  
 $n = 31$  and  $S = \{1, 15\}$ ,  $n = 31$  and  $S = \{1, 5\}$ ,  $n = 45$  and  $S = \{1, 21\}$ ,  $n = 51$  and  $S = \{1, 9\}$ ,  $n = 51$  and  $S = \{0, 1, 5\}$ ;
6.  $C$  is a sub-code of one of the codes of the above cases;
7.  $C$  is equivalent to one of the codes of the above cases.

In all cases  $b$  is very short and in most cases a formula can be given.

## 7 A Newton-Based Decoder

A different approach based on Newton identities (3) has been recently proposed by Augot et al. (2007) (see also Augot et al. 2003): unlike Orsini and Sala (2005), whose aim is to produce a single *general* locator, they follow the suggestion given by Caboara (2002) (Remark 7) of splitting the computation according to the potential weights. Denote

$$\mathcal{F}_\mu^{(\hat{\sigma})} := \left\{ \hat{\sigma}_j - (-1)^j \sum_{1 \leq l_1 \leq \dots \leq l_j \leq \mu} z_{l_1} \cdots z_{l_j}, 1 \leq j \leq \mu \right\}$$

$$\subset \mathbb{F}[\hat{\sigma}_1, \dots, \hat{\sigma}_\mu, z_1, \dots, z_\mu],$$

$$\mathcal{F}_\mu^{(X)} := \left\{ x_i - \sum_{j=1}^{\mu} z_j^i, 1 \leq i \leq \mu+n \right\} \cup \{x_{i+n} - x_i, 1 \leq i \leq \mu\} \subset \mathbb{F}[X, z_1, \dots, z_\mu],$$

$$\mathfrak{l}_\mu \subset \mathbb{F}[\hat{\sigma}, X, Z] = \mathbb{F}[\hat{\sigma}_1, \dots, \hat{\sigma}_\mu, x_1, \dots, x_{\mu+n}, z_1, \dots, z_\mu] := \mathcal{Q}$$

the ideal generated by  $\mathcal{F}_\mu^{(\hat{\sigma})} \cup \mathcal{F}_\mu^{(X)}$ ,  $\Delta_\mu := \prod_{i=1}^\mu z_i \prod_{1 \leq i < j \leq \mu} (z_i - z_j)$  and  $\mathfrak{l}_\mu^\infty{}^2 = \{f \in \mathcal{Q} : \exists n \in \mathbb{N} : f \Delta_\mu^n \in \mathfrak{l}_\mu\} \cap \mathbb{F}[\hat{\sigma}_1, \dots, \hat{\sigma}_\mu, x_1, \dots, x_{\mu+n}]$ .

**Fact 15** (Augot et al. 2007) Denoting by  $G_\mu$  the Gröbner basis of  $\mathfrak{l}_\mu^\infty$  w.r.t. the lex ordering induced by  $\hat{\sigma}_i < x_l, l \notin S$  and  $\hat{\sigma}_i > x_l, l \in S$ ,  $T_\mu := G_\mu \cap \mathbb{F}[x_l : l \in S]$ , the following hold

1.  $\mathfrak{l}_\mu^\infty$  is a radical 0-dimensional ideal;
2. its roots  $(\sigma_i, s_l)$  are exactly the values  $\sigma_i = (-1)^j \sum_{1 \leq l_1 \leq \dots \leq l_j \leq \mu} e_{l_1}$  and  $s_l = \sum_{j=1}^\mu e_j^l$  where  $e_1, \dots, e_\mu$  run among the error locations of the words of weight exactly  $\mu$ ;
3. for each  $i, 1 \leq i \leq \mu$  there are  $p_{i\mu}, q_{i\mu} \in \mathbb{F}[x_l : l \in S]$  such that  $p_{i\mu}\sigma_i - q_{i\mu} \in G_\mu$ ;
4. for an error  $e$  and the corresponding syndromes  $(s_l : l \in S)$  we have
  - the weight of  $e$  is  $\mu$  if and only if  $t(s_l) = 0$  for each  $t \in T_\mu$
  - the corresponding error locator polynomial is  $1 + \sum_{i=1}^\mu \frac{q_{i\mu}(s_l)}{p_{i\mu}(s_l)} z^i$ .

Thus the associated decoding algorithm consists in

1. (precomputation) For each weight  $\mu$  compute the Gröbner basis  $G_\mu$  of  $\mathfrak{l}_\mu^\infty$  w.r.t. the lex ordering induced by  $\hat{\sigma}_i < x_l, l \notin S$  and  $\hat{\sigma}_i > x_l, l \in S$ ,
2. (precomputation) For each  $\mu$  and each  $i$  extract the polynomials  $p_{i\mu}, q_{i\mu} \in \mathbb{F}[x_l : l \in S]$  such that  $p_{i\mu}\sigma_i - q_{i\mu} \in G_\mu$ ,
3. (precomputation) For each  $\mu$ , identify the set  $T_\mu := G_\mu \cap \mathbb{F}[x_l : l \in S]$ ,
4. (on line) for any received word
  - (a) compute the corresponding syndromes  $(s_l : l \in S)$
  - (b) evaluating  $t(s_l), t \in T_\mu$ , deduce  $\mu$
  - (c) return  $L_e(z) := 1 + \sum_{i=1}^\mu \frac{q_{i\mu}(s_l)}{p_{i\mu}(s_l)} z^i$ .

*Remark 9* Unfortunately, Augot et al. (2007) avoid discussing the size of the data, thus preventing the reader from making a fair comparison with the results of Orsini and Sala (2005). Mainly on the basis of the results of Alonso et al. (1996) the gut feeling of the first author is that while Augot et al. (2007) loses against Orsini and Sala (2005) as regards space ( $\mu$  different error locator polynomials have necessarily to be stored) probably one should prefer Augot et al. (2007) as regards time. The reader can in any case reach his own opinion comparing Orsini and Sala (2005) data (Example 6) with the best available approximation of Augot et al. (2007) data, namely Remark 7.

For an approach to decode linear codes with Gröbner bases, see Bulygin and Pellikaan (2009).

---

<sup>2</sup> $\mathfrak{l}_\mu^\infty$  can be computed as  $\mathfrak{l}_\mu^\infty = \bar{\mathfrak{l}}_\mu \cap \mathbb{F}[\hat{\sigma}_1, \dots, \hat{\sigma}_\mu, x_1, \dots, x_{\mu+n}]$  where  $\bar{\mathfrak{l}}_\mu \subset \mathcal{Q}[T]$  is the ideal generated by  $\mathcal{F}_\mu^{(\hat{\sigma})} \cup \mathcal{F}_\mu^{(X)} \cup \{1 - \Delta_\mu T\}$ .

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

The second author would like to thank her supervisors T. Mora and M. Sala.

## References

- M.-E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann, *Zeros, multiplicities, and idempotents for zero-dimensional systems*, Proceedings of MEGA 1994, Birkhäuser, Basel, 1996, pp. 1–15.
- D. Augot, M. Bardet, and J.-C. Faugère, *Efficient decoding of (binary) cyclic codes above the correction capacity of the code using Gröbner bases*, Proc. of ISIT 2003, 2003, pp. 362.
- D. Augot, M. Bardet, and J.-C. Faugère, *On formulas for decoding binary cyclic codes*, Proc. of ISIT 2007, 2007, pp. 2646–2650.
- D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- E. R. Berlekamp, *Algebraic coding theory*, McGraw–Hill, New York, 1968.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basislemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*. Multidimensional systems theory. Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- S. Bulygin and R. Pellikaan, *Decoding linear error-correcting codes up to half the minimum distance with Gröbner bases*, this volume, 2009, pp. 361–365.
- M. Caboara, *The Chen-Reed-Helleseth-Truong decoding algorithm and the Gianni-Kalkbrenner Gröbner shape theorem*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 3, 209–232.
- X. Chen, I. S. Reed, T. Helleseth and K. Truong, *General principles for the algebraic decoding of cyclic codes*, IEEE Trans. on Inf. Th. **40** (1994a), 1661–1663.
- X. Chen, I. S. Reed, T. Helleseth and T. K. Truong, *Algebraic decoding of cyclic codes: a polynomial ideal point of view*, Contemp. Math., vol. **168**, Amer. Math. Soc., Providence, 1994b, pp. 15–22.
- X. Chen, I. S. Reed, T. Helleseth and T. K. Truong, *Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance*, IEEE Trans. on Inf. Th. **40** (1994c), no. 5, 1654–1661.
- A.B. III Cooper, *Direct solution of BCH decoding equations*, Comm., Cont. and Sign. Proc. (1990), 281–286.
- A. B. III Cooper, *Finding BCH error locator polynomials in one step*, Electronic Letters **27** (1991), no. 22, 2090–2091.
- A. B. III Cooper, *Toward a new method of decoding algebraic codes using Gröbner bases*, Transactions of the Tenth Army Conference on Applied Mathematics and Computing (1992), vol. **93**, U.S. Army, 1993, pp. 1–11.
- P. Gianni, *Properties of Gröbner bases under specializations*, Proc. of EUROCAL1987, LNCS, vol. **378**, Springer, Berlin, 1989, pp. 293–297.
- M. Giorgatti and M. Sala, *A commutative algebra approach to linear codes*, BCRI preprint, [www.bcri.ucc.ie](http://www.bcri.ucc.ie), 58, UCC, Cork, Ireland, 2006.
- M. Giorgatti and M. Sala, *A commutative algebra approach to linear codes*, Journal of Algebra accepted (2009), 38.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick's approach to recent developments*, this volume, 2009, pp. 197–218.
- M. Kalkbrenner, *Solving systems of algebraic equations by using Gröbner bases*, Proc. of EUROCAL1987, LNCS, vol. **378**, Springer, Berlin, 1989, pp. 282–292.

- P. Loustaunau and E. V. York, *On the decoding of cyclic codes using Gröbner bases*, AAECC **8** (1997), no. 6, 469–483.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- T. Mora, E. Orsini and M. Sala, *General error locator polynomials for binary cyclic codes with  $t \leq 2$  and  $n < 63$* , BCRI preprint, [www.bcri.ucc.ie](http://www.bcri.ucc.ie) 43, UCC, Cork, Ireland, 2006.
- E. Orsini and M. Sala, *Correcting errors and erasures via the syndrome variety*, J. Pure Appl. Algebra **200** (2005), 191–226.
- E. Orsini and M. Sala, *General error locator polynomials for binary cyclic codes with  $t \leq 2$  and  $n < 63$* , IEEE Trans. on Inf. Th. **53** (2007), 1095–1107.

# A Tutorial on AG Code Construction from a Gröbner Basis Perspective

Douglas A. Leonard

## 1 Introduction

This tutorial is meant to stimulate interplay between those working with AG codes and those working with Gröbner bases (Buchberger 1965, 2006), as well as making this material more accessible to those not trained in algebraic geometry. It will be presented in two separate chapters. This first chapter is meant primarily as an introduction to AG codes, but includes material on producing proper *one-point* descriptions of these codes as well. The second Chap. (Leonard 2009) contains material about *syndrome decoding* and *list decoding*. For encoding of AG codes see Little (2009). The terminology chosen is therefore that of the easily understood concepts of *multivariate polynomial rings* and *ideals* of relations among the variables, which is closer to the notation of *function fields*, and much more useful computationally than the more standard algebraic geometry terminology. It is also an approach that generalizes to other projective varieties. Gröbner basics are covered elsewhere (Mora 2009) in this volume, but there are sections introducing the needed concepts about weighted orderings and the needed concepts for describing RS and AG codes; given that the readers of this paper may know one topic but not the other, or may know both but have not seen this material expressed in a language that can be used by both. (Those familiar only with RS codes and not AG codes, should be able to catch on by paying attention to the parity-check or generator functions used in the examples, since all the algorithms are generalizations of algorithms known for RS codes.)

History and bibliography, at least through 1998, is sufficiently covered by Høholdt, Pellikaan, and van Lint in Chap. 10 of the Handbook of Coding Theory (Høholdt et al. 1998), but some interesting papers in the literature about producing one-point descriptions relative to this viewpoint are Leonard (2001, 2009), Leonard and Pellikaan (2003). There are also some recent papers related to finding good AG codes (Beelen et al. 2006; Garcia and Stichtenoth 2005), which in turn cite the literature of that topic. Examples are freely “borrowed” from existing literature because they are described and/or worked differently here, reflecting more or less a decade of extra insight. Though there is some background material throughout, this is not

---

D.A. Leonard

Department of Mathematics and Statistics, Auburn University, Auburn, AL, 36849, USA  
e-mail: [leonada@auburn.edu](mailto:leonada@auburn.edu)

meant as a “survey” paper, but strictly as a “tutorial” for Gröbner basis methods that can be applied to AG codes to properly define them (this chapter) and decode them (Leonard 2009).

*Linear codes* are described in Augot et al. (2009). They are subspaces of dimension  $k$  of  $(\mathbb{F}_q)^n$ ,  $n$  being the *wordlength*. Such codes are the row-spaces of  $k \times n$  *generator matrices* and have  $(n - k) \times n$  *parity-check matrices* (generators for the *orthogonal complements* or *dual codes*), given either explicitly or implicitly.

Consider an *evaluation matrix*  $\text{Eval}$  with entries  $\text{Eval}_{i,j} := f_i(P_j)$  for some points  $P_j$  and functions  $f_i$ . The classic *RS codes* (short for *Reed–Solomon*) have both generator and parity-check matrices of this form, with points  $P_j$  the non-zero elements of  $\mathbb{F}_q$  (with extended RS codes using 0 as well) and functions  $f_i := x^i$  for some appropriate consecutive sequence of indices  $i$ .

*Functionally decoded* RS codes (those with  $\text{Eval}$  as parity-check matrix) generally use *syndromes*  $s_i$ , the entries of  $\underline{s} := \underline{r}\text{Eval}^T = \underline{e}\text{Eval}^T$  for decoding by determining the error  $\underline{e}$  with weight less than  $d/2$ ,  $d$  the *minimum distance* of the code. Algorithms such as *Berlekamp–Massey* or the *extended Euclidean algorithm* are used to reduce a syndrome matrix efficiently, to produce an *error-locator polynomial*, with roots the *error positions*. There are various other algorithms that can then be invoked to calculate the *error magnitudes*. *Functionally encoded* RS codes (those with  $\text{Eval}$  as generator matrix) on the other hand, lend themselves to *list-decoding* techniques for recovering a list of messages with some corresponding to codewords  $\underline{c}$  close to the received word  $\underline{r}$ .

RS codes are *maximum-distance-separable* codes, satisfying the *Singleton bound*,  $k + d = n + 1$ . But the wordlength is bounded by the field size  $q$ , so that creating longer length codes requires increasing the field size, or paying a severe penalty to use *subfield subcodes*, as with *BCH* codes.

To generalize these codes to *AG codes* (short for *algebraic geometry codes* in recognition of their origin), first start by thinking of the elements  $\alpha \in \mathbb{F}_q$  as *affine points*, then change them to *projective points*  $(\alpha : 1)$  on the *projective line* over  $\overline{\mathbb{F}}_q$ , the *algebraic closure* of  $\mathbb{F}_q$ ; having one extra point  $P_\infty := (1 : 0)$ , at which the *rational homogeneous functions*  $(x_1/x_0)^i$  have all  $i$  of their *poles*. Then  $\text{Eval}_{i,j} = (x_1/x_0)^i((\alpha_j : 1)) = (\alpha_j/1)^i$ . (So far this is merely a notational change to motivate the generalization.)

Then consider replacing the projective line by a *projective variety*,

$$\mathcal{X} := \{(x_N : \dots : x_1 : x_0) \in \mathbf{P}^N(\overline{\mathbb{F}}_q) : f(x_N, \dots, x_1, x_0) = 0 \text{ for all } f \in \mathcal{I}\}$$

for some *ideal*  $\mathcal{I}$  of defining relations (polynomials that should be zero at points of  $\mathcal{X}$ ).  $\mathcal{X}$  is a *projective curve*, roughly speaking, when there is only one *independent* variable relative to these relations. Some such curves have many points and easily described parameters.

Simple examples of this are *Hermitian curves* defined projectively by

$$\left(\frac{x_2}{x_0}\right)^q + \left(\frac{x_2}{x_0}\right) - \left(\frac{x_1}{x_0}\right)^{q+1} = 0$$

or in easier affine terms (without the *homogenizing variable*  $x_0$ ) by  $x_2^q + x_2 - x_1^{q+1} = 0$  having  $1 + q^3$  points rational over  $\mathbb{F}_{q^2}$ . There are more complicated examples, such as those from the *towers* defined by Garcia and Stichtenoth (1996), described in affine terms by

$$x_{i+1}^q + x_{i+1} = \frac{x_i^q}{x_i^{q-1} + 1}, \quad 1 \leq i \leq N.$$

(Warning such descriptions are *not* unique, as shall be mentioned in the section on curve definition below.)

These AG codes pay a *penalty*,  $g$ , relative to the Singleton bound, namely that  $k + d \geq n + 1 - g$ . They were initially of interest as a source of codes with  $k/n + d/n \geq 1 - (g - 1)/n \geq 1 - 1/(\sqrt{q} - 1)$ , the *Tsfasman–Vlăduț–Zink bound*, which can be better than the more traditional *Gilbert–Varshamov bound*, meaning that they had a reasonably good tradeoff between *information rate* and *relative distance*. Moreover, they were codes with structure to them as well as having good parameters, meaning that they probably could be efficiently decoded. While these codes can have length much larger than  $q$ , it is at most the *Hasse–Weil bound*  $n \leq q + 1 + 2g\sqrt{q}$ .

This chapter of the tutorial will discuss producing a reasonable description (in terms of the function space of an evaluation matrix) of AG codes from less useful definitions (as above), while the subsequent chapter (Leonard 2009) will be devoted to syndrome decoding algorithms for functionally decoded AG codes (called *geometric Goppa codes*  $\mathcal{C}^*(\mathcal{D}, \mathcal{G})$  in the Handbook of Coding Theory Høholdt et al. 1998 and *dual codes* in Sakata 2009) and list-decoding algorithms for functionally encoded AG codes (called *geometric Reed–Solomon codes*  $\mathcal{C}(\mathcal{D}, \mathcal{G})$  in Høholdt et al. 1998 and *primary codes* in Sakata 2009). All are topics intimately related to *ideals* and their *Gröbner bases* and  $\Delta$ -sets (or *footprints* or *standard monomial bases*). However, in general, in the literature, for various reasons, this purely ideal-theoretic approach is often slighted.

There are example programs for some of this material, written in MAGMA, as opposed to some generic pseudo-code. Those with access to MAGMA (MAGMA et al. 2008; Bosma et al. 1997; Cannon and Playoust 1996) can run these easily enough, while those without such access will have to treat it as fairly readable pseudo-code. More example calculations and programs should be available on the author's Auburn University website: [www.dms.auburn.edu/~leonada](http://www.dms.auburn.edu/~leonada)

## 2 Traditional AG Approach

A more traditional algebraic geometry approach to curves might start with *divisors*,  $\mathcal{D} := \sum_P n_P \cdot P$ , with (finite) *degree*  $\deg(\mathcal{D}) := \sum_P n_P$ , which are useful additive bookkeeping devices for keeping track of zeros and poles of functions. As an example, relative to the Klein quartic, the divisors

$$\left( \frac{x_1}{x_0} \right) = -2 \cdot P - 1 \cdot Q + 3 \cdot R, \quad \left( \frac{x_2}{x_0} \right) = -3 \cdot P + 2 \cdot Q + 1 \cdot R$$

denote that both functions have 3 poles and 3 zeros restricted to the support  $P := (1 : 0 : 0)$ ,  $Q := (0 : 1 : 0)$ , and  $R := (0 : 0 : 1)$ . In general, (*equivalence classes of rational homogeneous functions modulo the curve*) necessarily have the same (finite) number of poles as zeros, except for the identically zero function itself. More significantly, these functions have *Laurent series expansions* at each point  $P$  on the curve, written in terms of some *local parameter*,  $t_P$ , a function having a simple zero at  $P$ . So, for instance,

$$\frac{x_1}{x_0} = t_P^{-2} + \cdots = t_Q^{-1} = t_R^3 + \cdots, \quad \frac{x_2}{x_0} = t_P^{-3} + \cdots = t_Q^2 + \cdots = t_R^1$$

if  $t_P := x_1/x_2$ ,  $t_Q := x_0/x_1$ , and  $t_R := x_2/x_0$ .

Divisors are also useful in defining vector spaces modulo the curve; namely

$$\mathcal{L}(\mathcal{D}) := \{0\} \cup \{f : (f) + \mathcal{D} \succeq 0\},$$

meaning that the *valuation*  $v_P(f)$ , the *trailing exponent* in the Laurent series expansion of  $f$  at  $P$  satisfies  $v_P(f) + n_P(\mathcal{D}) \geq 0$ .

*Differentials*,  $df$ , can also be defined modulo the curve; and by rewriting  $df = f_P dt(P)$ , the divisor  $(df) := \sum_P v_P(f_P) \cdot P$  can be defined as well. If  $\omega$  is a fixed differential, then the *Riemann–Roch theorem* is:

$$\dim(\mathcal{L}(\mathcal{D})) - \dim(\mathcal{L}((\omega) - \mathcal{D})) = \deg(\mathcal{D}) - g + 1$$

for  $g$  the *genus* of the underlying (smooth projective) curve. And the corollary that  $\deg((\omega)) = 2g - 2$  suggests a method of computing said genus.

Here the divisor  $\mathcal{D}$  in the definitions of AG codes above will always be of the form  $\sum_{j=1}^n 1 \cdot P_j$ , denoting the points used for evaluation. The divisor  $\mathcal{G}$ , with support disjoint from that of  $\mathcal{D}$ , will define the vector space  $\mathcal{L}(\mathcal{G})$  of functions used for evaluation. And for *one-point AG codes* this means  $\mathcal{G} = m \cdot P_\infty$ , so that these functions will have at most a pole of order  $m$  at  $P_\infty$  and no other poles. The code  $\mathcal{C}^*(\mathcal{D}, \mathcal{G}) = \mathcal{C}(\mathcal{D}, (\omega) + \mathcal{D} - \mathcal{G})$ , but the encoding is usually described in terms of the map  $f\omega \mapsto (\text{Res}(P_j, f\omega) : 1 \leq j \leq n)$ , so that the *Residue theorem*:

$$\sum_P \text{Res}(P, hf\omega) = 0$$

can be invoked to show the duality of the two codes, with residue having the standard meaning as the coefficient of  $1/t(P)$  in the Laurent series expansion.<sup>1</sup>

Riemann–Roch can also be used to show that  $\dim(\mathcal{L}(\mathcal{G})) \geq \deg(\mathcal{G}) - (g - 1)$  with equality when  $\deg(\mathcal{G}) > \deg(\omega)$ . And it can be used as well to show that  $d \geq (n - k) - (g - 1)$ , explaining  $g$  as a penalty relative to the Singleton bound alluded to earlier.

<sup>1</sup>In the literature all that is usually found is the use of entries  $f(P)$  to define a generator matrix  $G$  or parity-check matrix  $H$ , with very little reference to the other, defined by evaluating  $\text{Res}(P, f\omega)$ . This is probably because those using functional encoding to get  $G$  do not usually use  $H$ , and those using it to define  $H$ , limit their investigations to decoding, ignoring  $G$ .

### 3 Weighted Total-Degree Orders

There are many good books covering introductory material on Gröbner bases, with Cox et al. (2007) being the author's favorite. The classical papers by Bruno Buchberger himself are Buchberger (1965, 1970, 1998, 1985, 2006). Much information is in the Gröbner Technology section of this volume, written by Mora (2009); but the emphasis there is not on the orderings (called *term orders* there). So the following is an addendum to Mora (2009) for this purpose. The (default) *lexicographical order* is based on comparing products by considering their *indices* (that is, *exponents*) lexicographically. So, for instance, in the multivariate polynomial ring  $\mathbb{F}[x_3, x_2, x_1]$  the order looks like

$$1 \prec x_1 \prec x_1^2 \prec \cdots \prec x_2 \prec x_2 x_1 \prec \cdots \prec x_2^2 \prec \cdots \prec x_3 \prec \cdots$$

which can be described by  $x_3^{i_3} x_2^{i_2} x_1^{i_1} \succ x_3^{j_3} x_2^{j_2} x_1^{j_1}$  iff  $(i_3 > j_3)$  or  $(i_3 = j_3 \text{ and } i_2 > j_2)$  or  $(i_3 = j_3 \text{ and } i_2 = j_2 \text{ and } i_1 > j_1)$ . The *total-degree orders* of interest here are the *grevlex* and *wtdeg* orders (short for *graded reverse lexicographical* and *weighted total degree*), in which (weighted) total degree is the first concern.

These can be defined in ways similar to that above; but the non-singular matrices

$$A_{\text{grevlex}} := \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad A_{\text{wtdeg}} := \begin{pmatrix} w_3 & w_2 & w_1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

can be used to reduce these to the lexicographical case by converting the column vector of exponents

$$\begin{pmatrix} i_3 \\ i_2 \\ i_1 \end{pmatrix} \quad \text{to} \quad \begin{pmatrix} i_3 + i_2 + i_1 \\ i_3 + i_2 \\ i_3 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} w_3 i_3 + w_2 i_2 + w_1 i_1 \\ i_3 + i_2 \\ i_3 \end{pmatrix}$$

respectively.

### 4 Hermitian Codes and Affine-Variety Codes

The most common (and easily implemented) examples of *one-point AG codes* are those from *Hermitian curves* alluded to above; but, at the same time, these least exemplify the general theory, as shall be described here. Let  $q := p^m$ ,  $p$  a prime. Then the affine equation  $\text{Trace}(y) := y^q + y = x^{q+1} =: \text{Norm}(x)$  defines an Hermitian curve in characteristic  $p$  by

$$\mathcal{X} := \{(x_2 : x_1 : x_0) \in \mathbf{P}^2(\overline{\mathbb{F}}_p) : x_2^q x_0 + x_2 x_0^q - x_1^{q+1} = 0\}$$

The rational functions  $f_q := x_1/x_0$  and  $f_{q+1} := x_2/x_0$  have divisors

$$(f_q) = (-q) \cdot P_\infty + \sum_{i=0}^{q-1} 1 \cdot P_i, \quad (f_{q+1}) = (-q-1) \cdot P_\infty + (q+1) \cdot P_0$$

with  $P_\infty := (1 : 0 : 0)$  and  $P_i := (\alpha_i : 0 : 1)$ ,  $\alpha_0 := 0$ , and  $\alpha_i^{q-1} + 1 = 0$ ,  $1 \leq i \leq q-1$  defined over  $\mathbb{F}_{q^2}$  if  $p$  is odd,  $\mathbb{F}_q$  if  $p$  is even.

These curves are already in one-point form, since  $f_q$  and  $f_{q+1}$  clearly have no poles except at  $P_\infty$ . Also the differential  $\omega := d(f_q)$  has divisor  $(\omega) = (q-2)(q+1) \cdot P_\infty$ . So here  $\omega = 1d(t_P)$  for all  $P \neq P_\infty$ . And the genus  $g = q(q-1)/2$ , meaning the (rational) functions  $f_{(q+1)i+qj} := f_{q+1}^i f_q^j$ ,  $0 \leq i < q$ ,  $0 \leq j$  can be used to index the rows of both generator and parity-check matrices for the corresponding linear codes, normally chosen over  $\mathbb{F}_{q^2}$ , since there are  $1 + q^3 = q^2 + 1 + 2g\sqrt{q^2}$  rational points there.

If  $\mathbf{I}_{q^2} := \langle f_{q+1}^{q^2} - f_{q+1}, f_q^{q^2} - f_q \rangle$ , then  $\mathbf{I} := \langle \mathbf{I}_{q^2}, f_{q+1}^q - f_q^{q+1} + f_{q+1} \rangle$  has Gröbner basis  $\langle f_{q+1}^q - f_q^{q+1} + f_{q+1}, f_q^{q^2} - f_q \rangle$ , with

$$\Delta(I) = \{f_{q+1}^i f_q^j, 0 \leq i < q, 0 \leq j < q^2\}$$

of size  $n = q^3$ . In the examples below, with  $q = 4$ , the elements of  $\{0, \gamma^0, \gamma^5, \gamma^{10}\}$  have trace 0, those of  $\{\gamma^1, \gamma^2, \gamma^4, \gamma^8\}$  have trace 1, those of  $\{\gamma^6, \gamma^9, \gamma^7, \gamma^{13}\}$  have trace  $\gamma^5$ , and those of  $\{\gamma^{12}, \gamma^3, \gamma^{14}, \gamma^{11}\}$  have trace  $\gamma^{10}$ . And  $x^i = 1, \gamma^5, \gamma^{10}$  for  $i \equiv 0, 1, 2 \pmod{3}$  respectively. This describes all 64 points other than  $P_\infty$ .

Some of the reasons why these curves are atypical are:

1. most good curves are not given initially in one-point form;
2. most curves (in affine form) are not described by only one defining relation in two variables;
3. there are usually functions (used to define either the generator or parity-check matrices) that are not monomials in the given variables;
4. most codes need different sets of functions to define generator and parity-check matrices, because  $(\omega)$  is not usually a multiple of  $P_\infty$ .

It is also possible to think of AG codes strictly in terms of ideals in multivariate polynomial rings, meaning it is possible to choose a (necessarily finite) set of affine points  $V \subseteq (\mathbb{F}_q)^s$ , produce by interpolation the ideal  $\mathbf{I}(V)$  having  $V$  as its variety, and think of  $\mathcal{X}$  as having intersection  $V$  with  $(\mathbb{F}_q)^s$ , without ever really defining it. The codes gotten by evaluating some  $f \in \Delta(I)$  on the points of  $V$  are called *affine-variety codes* for obvious reasons. *Reed–Muller codes* (and hence *extended Reed–Solomon codes* as well) can be viewed this way with  $V := \mathbb{F}_q^s$  and  $\mathbf{I} := \mathbf{I}_q := \langle x_1^q - x_1, \dots, x_s^q - x_s \rangle$ . Conceivably it is possible to consider all these as AG codes, but it is probably better, in general, to limit the usage of the term *AG code* to  $\mathbf{V} \subseteq \mathbf{V}(I)$  for  $I$  describing a curve (or surface), since any linear code over  $\mathbb{F}_q$  can be viewed as an affine-variety code.

A recent interesting approach is that of Order Domain codes (Geil 2009).

## 5 Curve Definition

Producing descriptions of codes from curves can be done in various ways. For instance, the standard projective description of the *Klein quartic* in characteristic 2 (which has 24 rational points over  $\mathbb{F}_8$ ) is  $x_2^3x_0 + x_1^3x_2 + x_0^3x_1 = 0$ , but a proper description for use as a *one-point AG code* is given in affine terms by  $f_7^2 + f_5f_3^3 + f_7 = 0$ ,  $f_7f_5 + f_3^4 + f_5 = 0$ , and  $f_5^2 + f_7f_3 = 0$  for  $f_3 := x_2/x_0$ ,  $f_5 := x_2x_1/x_0^2$ , and  $f_7 := x_2x_1^2/x_0^3$ , this being an induced Gröbner basis gotten from the standard description after changing variables. (21 of the  $24 - 1$  affine points over  $\mathbb{F}_8 := \mathbb{F}_2[\beta]/(1 + \beta + \beta^3)$  have  $f_3(P) := \beta^{5i}$ ,  $f_5(P) := \beta^{6i}\delta_j$  and  $f_7(P) := \delta_j^2$  for  $\delta_j^3 + \delta_j + 1 = 0$ ; the other two having  $f_3(P) = 0 = f_5(P)$  and  $f_7(P) \in \{0, 1\}$ .) And there is a two-point description given by  $x_1^3x_2 + x_2^3(x_2 + x_1 + x_0) + (x_2 + x_1 + x_0)^3x_1 = 0$  with the 2 points at infinity, rational over  $\mathbb{F}_4$ , not  $\mathbb{F}_8$ .

*One-point AG codes*, that is, those described by the functions from  $\mathcal{L}(m \cdot P_\infty)$  (with poles only at  $P_\infty$  and of pole order at most  $m$  there), will have an  $\overline{\mathbb{F}}_q[f_\rho]$ -module basis of size  $\rho$  with elements having smallest possible pole sizes for each  $i \pmod{\rho}$ ; said functions describing the multivariate polynomial ring used, and an ideal of relations describing the multiplication of those functions, modulo  $\mathcal{X}$ . For some results on the relations between multi-point codes and one-point codes, see Matthews (2009).

The problem is that  $\mathcal{X}$  is not usually given in a such a friendly manner. Consider the curve described by:

$$x_2^2 + x_2 = \frac{x_1^2}{x_1 + 1}, \quad x_3^2 + x_3 = \frac{x_2^2}{x_2 + 1}$$

in characteristic 2, an example from the second towers found by Garcia and Stichtenoth. While this may define the curve  $\mathcal{X}$ , it doesn't allow for evaluation at several points at which some of the variables have poles. (In fact some points  $P$  cannot even be described in terms of the projective coordinates  $x_3(P)$ ,  $x_2(P)$ ,  $x_1(P)$ , and  $x_0(P)$ .)

The following MAGMA code first uses a change of variables to  $x_4$ ,  $x_6 := x_2(x_4 + 1)$  and  $x_7 := x_1(x_2 + 1)(x_4 + 1)$  with poles of orders 4, 6, and 7 at some point  $P_\infty$  and no other poles, followed by a computation of a value  $\Delta \in \mathbb{F}_2[x_4]$  such that  $\mathcal{L}(m \cdot P_\infty) \subseteq 1/\Delta\mathbb{F}_2\langle 1, y_6, y_7, y_7y_6 \rangle/\mathcal{I}$  for  $\mathcal{I} := \langle x_7^2 + x_7(x_6 + x_4) + x_6x_4^2, x_6^2 + x_6(x_4 + 1) + x_4^2(x_4 + 1) \rangle$ , followed by an implementation of the author's *q-th power algorithm* (that suffices for this and some similar examples) to produce the *missing function*  $y_5$ , to get a full set of parity-check functions  $\mathcal{L}(m \cdot P_\infty)$  for the AG codes related to this curve.

```
q:=2;
F:=FiniteField(q);
wt:=[7,6,4];
n:=#wt;r:=n-1;
P<x1,x2,x7,x6,x4>:=PolynomialRing(F,n+r);
e1:=x1*(x1+1)*(x2+1)-x2^2;
e2:=x2*(x2+1)*(x4+1)-x4^2;
def6:=x2*(x4+1)-x6;
```

```

def7:=x1*(x2+1)*(x4+1)-x7;
I:=ideal<P|e1,e2,def6,def7>;
EI:=EliminationIdeal(I,r);
GI:=GroebnerBasis(EI);
JM:=JacobianMatrix(GI);
M:=Minors(JM,n-1);
J:=ideal<P|GI,M>;
GJ:=GroebnerBasis(J);
d:=GJ[#GJ];
"-----";
R:=PolynomialRing(F,n,"grevlexw",wt);
AssignNames(~R,["y7","y6","y4"]);
x:=R.n;
hPR:=hom<P->R|0,0,R.1,R.2,R.3>;
wR:=function(ff)
  return &+[Degree(LeadingMonomial(ff),R.i)*wt[i]: i in [1..n]];
end function;
IR:=ideal<R|[GI[i]@hPR:i in [1..#GI]]>;
GR:=GroebnerBasis(IR);
QR:=quo<R|IR,R.n-1>;
N0:=MonomialBasis(QR);
h0:=hom<QR->R|[R.i:i in [1..n]]>;
M0:=h0(N0);
dR:=d@hPR;"Delta=",dR;
"-----";
for i in [1..#M0] do
  "B[,wR(M0[i])-wR(dR),"]=",M0[i];
end for;
DR:=(dR)^(q-1);
nf:=function(gg)
  return NormalForm(gg^q,IR);
end function;
LT:=function(ff) return LeadingTerm(ff); end function;
LM:=function(ff) return LeadingMonomial(ff); end function;
LC:=function(ff) return LeadingCoefficient(ff); end function;
N:=wR(dR)+1+Maximum([wR(M0[i]):i in [1..#M0]]);
zero:=[R|0: i in [1..N]];

change:=true;
while change do
  change:=false;
  g_new:=zero;
  k:=zero;
  Bnew:=[];
  for i in [1..#M0] do
    j:=1+wR(M0[i]);
    g_new[j]:=M0[i];
    k[j]:=nf(g_new[j]);
  end for;
"-----";
  loop:=true;
  i:=1;
  while loop and (i le N) do
    loop:=false;
    empty:=false;
    if g_new[i] eq 0 then
      loop:=true;
      i+=1;
      empty:=true;
    end if;
    if empty eq false then
      if k[i] eq 0 then
        Append(~Bnew,i);
        "B[,i-wR(dR)-1,"]=",g_new[i];
        loop:=true;
        i+=1;
      else

```

```

for j in [1..#M0] do
    bool,qu:=IsDivisibleBy(LT(k[i]),LT(M0[j]*dR));
    if bool then
        k[i]:=qu*M0[j]*dR;
        loop:=true;
        break;
    end if;
end for;
if (i gt 1) and (loop eq false) then
    lki:=LM(k[i]);
    for j in [1..i-1] do
        if k[j] ne 0 then
            if lki eq LM(k[j]) then
                lc:=LC(k[i])/LC(k[j]);
                g_new[i]:=lc*g_new[j];
                k[i]:=lc*k[j];
                loop:=true;
                change:=true;
                break;
            end if;
        end if;
    end for;
end if;
if loop eq false then
    j:=i+wt[n];
    g_new[j]:=g_new[i]*x;
    k[j]:=k[i]*x^q;
    loop:=true;
    change:=true;
    i+=1;
    end if;
end if;
end if;
end while;
M0:=[g_new[i]:i in Bnew];
end while;

W:=wR(dR)+1;
"-----";
S:=PolynomialRing(F,4,"grevlexw",[7,6,5,4]);
AssignNames(~S,[ "s7","s6","s5","s4"]);
hSR:=hom<S->R|g_new[7+W],g_new[6+W],g_new[5+W],g_new[W]*x>;
nn:=wt[n];
hRS:=hom<R->S|0,0,S.nn>;
BB:=[R|0: i in [1..nn]];
for i in Bnew do
    j:=wR(LT(g_new[i])) mod nn +1;
    BB[j]:=g_new[i];
end for;
s:=[S.(nn+1-i):i in [1..nn]];
s[1]:=1;
module:=function(i,j)

ff:=NormalForm((s[i]*s[j])@hSR,IR) div dR;
MM:=[R|0: i in [1..nn]];
while ff ne 0 do
    j:=wR(LM(ff)) mod nn +1;
    k:=LT(ff) div LT(BB[j]);
    ff:=k*BB[j];
    MM[j]:=k;
end while;
return MM;
end function;
SS:=[S|];
for i in [2..nn] do
    for j in [2..i] do
        mm:=module(i,j)@hRS;
    end for;
end for;

```

```

ms:= &+[mm[k]*s[k]:k in [1..nn]];
nij:=(s[i]*s[j])-ms;
Append(~SS,nij);
end for;
end for;
SI:=ideal<S|SS>;
SG:=GroebnerBasis(SI);
"A Grobner basis for the induced ideal is",SG;

```

Since there are other methods of producing the these functions (such as using MAGMA's *IntegralClosure* function or SINGULAR's *normal* function), perhaps a few words of explanation are in order, though there are various descriptions available in Leonard (2001), Leonard and Pellikaan (2003), and Leonard (2009). The *integral closure*  $ic(R)$  of a ring  $R$  (in this case,  $R = \mathbb{F}_2[x_7, x_6, x_4]/\mathcal{I}$ ) in its *field of fractions* is the *ring* of all elements satisfying a monic (that is, having leading coefficient 1) polynomial with coefficients from  $R$  (in this case,  $ic(R) = \bigcup_m \mathcal{L}(m \cdot P_\infty)$ ). Standard methods produce  $ic(R)$  by finding a nested sequence of larger and larger rings living between  $R$  and  $ic(R)$ . The *q-th power algorithm* finds a module  $M_0$  of the form  $\Delta^{-1}R$ , (with  $\Delta$  only involving the independent variable(s), in this case just  $x_4$ ) known to contain  $ic(R)$ , and then produces a nested sequence of smaller modules known to stabilize at  $ic(R)$ . These modules are easily described mathematically by  $M_{i+1} := \{f \in M_i : NormalForm(f^q, \mathcal{I}) \in M_i\}$ , and easily produced by an algorithm (such as the single-variable implementation given above) which is *linear* over  $\mathbb{F}_q$ . (The competing algorithms were designed to work over characteristic 0, to work on number fields, and not with any idea of weights in mind; so they do not immediately provide an appropriate description for this type of application.)

The output of the MAGMA program above, modified to be more readable, gives  $\Delta := y_4^2$ , and  $\mathbb{F}_2$ -module bases for  $\Delta M_i$  as  $B_0 := \{1, y_6, y_7, y_7y_6\}$ ,  $B_1 := \{y_4, y_6y_4, y_7y_4, y_7y_6\}$ ,  $B_2 := \{y_4^2, y_7y_6 + y_6y_4, y_6y_4^2, y_7y_4^2\} =: B_3$ , meaning the missing function is  $y_5 := (y_7y_6 + y_6y_4)/y_4^2$ . The curve can then be properly defined in *one-point form* as  $\overline{\mathbb{F}}_2[y_7, y_6, y_5, y_4]/\mathcal{J}$  for  $\mathcal{J}$  the ideal of induced relations (also produced by the code above) as

$$\begin{aligned}
\mathcal{J} := & \langle y_7^2 + y_6y_4^2 + y_5y_4^2 + y_7y_4 + y_6y_4 + y_7, \\
& y_7y_6 + y_5y_4^2 + y_6y_4, \\
& y_6^2 + y_4^3 + y_6y_4 + y_4^2 + y_6, \\
& y_7y_5 + y_4^3 + y_7y_4 + y_6y_4 + y_5y_4 + y_4^2 + y_7, \\
& y_6y_5 + y_7y_4 + y_5y_4 + y_4^2 + y_7 + y_5 + y_4, \\
& y_5^2 + y_6y_4 + y_5y_4 + y_4^2 + y_6 + y_5 + y_4 \rangle.
\end{aligned}$$

The codes over  $\mathbb{F}_4$  are then functionally encoded or functionally decoded relative to (some of) the  $n = 13$  functions  $f_0 := 1$ ,  $f_4 := y_4$ ,  $f_5 := y_5$ ,  $f_6 := y_6$ ,  $f_7 := y_7$ ,  $f_8 := y_4^2$ ,  $f_9 := y_5y_4$ ,  $f_{10} := y_6y_4$ ,  $f_{11} := y_7y_4$ ,  $f_{12} := y_4^3$ ,  $f_{13} := y_5y_4^2$ ,  $f_{14} := y_6y_4^2$ ,  $f_{15} := y_7y_4^2$  as generator or parity-check functions respectively. The  $n = 13$

affine points rational over  $\mathbb{F}_4$  correspond to the *variety* of the above ideal  $\mathcal{J}$  (intersected with  $\langle y_4^4 - y_4, y_5^4 - y_5, y_6^4 - y_6, y_7^4 - y_7 \rangle$ ) to restrict the curve to that subfield. In particular  $(y_7(P), y_6(P), y_5(P), y_4(P))$  is a coordinatization of the affine point  $P$ .

A slightly more complicated example of producing a one-point AG code description is:

$$p_I(T) := \sum_{k=0}^m a_k T^k \in \mathbb{F}_q[T]$$

be monic and irreducible. Define inductively,  $P_k \in \mathbb{F}_q[x, y]$  by  $P_0(x, y) := 1$  and

$$P_{k+1}(x, y) := x P_k(x, y)^{q^2} + y P_k(x, y)^q \quad \text{for } k \geq 0.$$

Then let

$$F(x, y) = F_I(x, y) := \sum_{k=0}^m a_k P_k(x, y).$$

The equation  $F_I(x, y) = 0$  is an analogue of the modular equation.

As a small example, let  $q := 2$  and  $p_I(T) := 1 + T + T^3$ . Then

$$\begin{aligned} P_1(x, y) &= x + y, & P_2(x, y) &= x^5 + x^2 y + x y^4 + y^3, \\ P_3(x, y) &= x^{21} + x^{10} y + x^9 y^4 + x^5 y^{16} + x^4 y^3 + x^2 y^9 + x y^{12} + y^7, \\ F(x, y) &= x^{21} + x^{10} y + x^9 y^4 + x^5 y^{16} + x^4 y^3 + x^2 y^9 + x(y^{12} + 1) \\ &\quad + (y^7 + y + 1). \end{aligned}$$

Start by using  $x_1 := x$  and  $x_2 := y + x$  to get

$$\begin{aligned} x_1^5(x_2^{16} + x_2^8) + x_1^3(x_2^8 + x_2^4) + x_1^2(x_2^9 + x_2^5) + x_1(x_2^{12} + x_2^6) + (x_2^7 + x_2 + 1), \\ (x_1) = (-8) \cdot P_1 + (-4) \cdot P_2 + (-4) \cdot P_3 + 5 \cdot P_4 + 4 \cdot P_5 + \sum_{j=1}^7 1 \cdot Q_j, \\ (x_2) = 0 \cdot P_1 + 2 \cdot P_2 + 3 \cdot P_3 + (-1) \cdot P_4 + (-4) \cdot P_5 + \sum_{j=1}^7 0 \cdot Q_j. \end{aligned}$$

Use  $y_2 := x_1 x_2$  to get

$$\begin{aligned} x_1^{11} + x_1^{10}(y_2^4 + y_2) + x_1^8(y_2^8 + y_2^5) + x_1^6(y_2^8 + y_2^6) + x_1^4(y_2^9 + y_2^7) + (y_2^{16} + y_2^{12}), \\ (y_2) = (-8) \cdot P_1 + (-2) \cdot P_2 + (-1) \cdot P_3 + 4 \cdot P_4 + 0 \cdot P_5 + \sum_{j=1}^7 1 \cdot Q_j. \end{aligned}$$

Use  $y_1 := x_1/y_2$ , to get

$$y_1^{11} + y_1^{10}(y_2^3 + 1) + y_1^8(y_2^5 + y_2^2) + y_1^6(y_2^3 + y_2) + y_1^4(y_2^2 + 1) + (y_2^5 + y_2),$$

$$(y_1) = 0 \cdot P_1 + (-2) \cdot P_2 + (-3) \cdot P_3 + 1 \cdot P_4 + 4 \cdot P_5 + \sum_{j=1}^7 0 \cdot Q_j.$$

Use  $z_2 := y_2(y_1 + 1)^2$  to get

$$\begin{aligned} z_2^5 + z_2^3 y_1^6 + z_2^2 (y_1^6 + y_1^4) + z_2 (y_1^6 + 1) + y_1^4 (y_1 + 1)^2 (y_1^7 + y_1 + 1), \\ (z_2) = 2 \cdot P_1 + (-6) \cdot P_2 + (-7) \cdot P_3 + 4 \cdot P_4 + 0 \cdot P_5 + \sum_{j=1}^7 1 \cdot Q_j. \end{aligned}$$

Use

$$h_{21} := y_1^7 + z_2^2 y_1 + z_2 y_1^2 + z_2 + y_1^2,$$

$$h_{25} := z_2 y_1^6 + z_2^3 + z_2^2 y_1 + z_2 y_1 + y_1^5 + y_1^3 + y_1^2 + 1,$$

to get a single defining relation:

$$\begin{aligned} h_{25}^{21} + h_{25}^{20} h_{21} + h_{25}^{18} (h_{21}^3 + h_{21} + 1) + h_{25}^{17} (h_{21}^3 + 1) \\ + h_{25}^{16} (h_{21}^4 + h_{21}) + h_{25}^{15} (h_{21}^7 + h_{21}^6 + h_{21}^3 + h_{21} + 1) + h_{25}^{14} h_{21}^7 \\ + h_{25}^{13} (h_{21}^8 + h_{21}^7 + h_{21}^6 + h_{21}^4 + h_{21}^3 + 1) + h_{25}^{12} (h_{21}^9 + h_{21}^8 + h_{21}^4 + 1) \\ + h_{25}^{11} (h_{21}^{11} + h_{21}^9 + h_{21}^8 + h_{21}^5 + h_{21}^4 + h_{21}^3 + h_{21}^2) \\ + h_{25}^{10} (h_{21}^{12} + h_{21}^9 + h_{21}^8 + h_{21}^7 + h_{21}^5 + h_{21}^3 + h_{21} + 1) \\ + h_{25}^9 (h_{21}^{14} + h_{21}^{13} + h_{21}^{10} + h_{21}^9 + h_{21}^8 + h_{21}^7 + h_{21}^6 + h_{21}^3 + h_{21}^2 + 1) \\ + h_{25}^8 (h_{21}^{13} + h_{21}^9 + h_{21}^8 + h_{21}^6 + h_{21}^4 + h_{21}^3 + h_{21}) \\ + h_{25}^7 (h_{21}^{16} + h_{21}^{15} + h_{21}^{13} + h_{21}^{12} + h_{21}^{11} + h_{21}^7 + h_{21}^3 + h_{21}) \\ + h_{25}^6 (h_{21}^{17} + h_{21}^{16} + h_{21}^{13} + h_{21}^9 + h_{21}^8 + h_{21}) \\ + h_{25}^5 (h_{21}^{17} + h_{21}^{16} + h_{21}^{12} + h_{21}^7 + h_{21}^5 + h_{21}^2 + h_{21} + 1) \\ + h_{25}^4 (h_{21}^{19} + h_{21}^{16} + h_{21}^{15} + h_{21}^{12} + h_{21}^6 + h_{21}^5 + h_{21}^3 + 1) \\ + h_{25}^3 (h_{21}^{18} + h_{21}^{15} + h_{21}^{12} + h_{21}^{10} + h_{21}^9 + h_{21}^7 + h_{21}^4 + h_{21}) \\ + h_{25}^2 (h_{21}^{22} + h_{21}^{21} + h_{21}^{20} + h_{21}^{18} + h_{21}^{13} + h_{21}^{12} + h_{21}^9 + h_{21}^8 + h_{21}^7 + h_{21}^5 + h_{21}^4 + h_{21}^3) \\ + h_{25} (h_{21}^{23} + h_{21}^{22} + h_{21}^{20} + h_{21}^{17} + h_{21}^{15} + h_{21}^{14} + h_{21}^{12} + h_{21}^9) \\ + (h_{21}^{25} + h_{21}^{23} + h_{21}^{19} + h_{21}^{17} + h_{21}^{15} + h_{21}^{13} + h_{21}^{11} + h_{21}^5). \end{aligned}$$

But then the  $q$ -th power algorithm produces the integral closure:

$$\mathbb{F}_2[h_{16}, h_{15}, h_{13}, h_{12}, h_{11}, h_{10}, h_7]/\mathcal{I};$$

with  $\mathcal{I}$  having Gröbner basis consisting of

$$\begin{aligned}
& h_{10}^2 + h_{13}h_7 + h_{11}h_7 + h_{10}h_7 + h_{15} + h_{13}, \\
& h_{11}h_{10} + h_7^3 + h_{13}h_7 + h_{15} + h_{13} + h_{11} + h_{10}, \\
& h_{11}^2 + h_{15}h_7 + h_{13}h_7 + h_{12}h_7 + h_{15} + h_{13} + h_{11} + h_{10}, \\
& h_{12}h_{10} + h_{15}h_7 + h_{13}h_7 + h_{12}h_7 + h_{12}, \\
& h_{12}h_{11} + h_{16}h_7 + h_{12}h_7 + h_{10}h_7, \\
& h_{12}^2 + h_{10}h_7^2 + h_{16}h_7 + h_{15}h_7 + h_7^3 + h_{13}h_7 + h_{12}h_7 + h_{11}h_7 + h_{14} + h_{12} + h_7, \\
& h_{13}h_{10} + h_{16}h_7 + h_{15}h_7 + h_{13}h_7 + h_{12}h_7 + h_{11}h_7 + h_7^2 + h_{12} + h_7, \\
& h_{13}h_{11} + h_{10}h_7^2 + h_{16}h_7 + h_{11}h_7 + h_{13} + h_{10} + h_7 + 1, \\
& h_{13}h_{12} + h_{11}h_7^2 + h_{15}h_7 + h_{12}, \\
& h_{13}^2 + h_{12}h_7^2 + h_{10}h_7^2 + h_{15}h_7 + h_{15}h_7^2 + 1, \\
& h_{15}h_{10} + h_{11}h_7^2 + h_{13}h_7 + h_{15} + h_7, \\
& h_{15}h_{11} + h_{12}h_7^2 + h_{11}h_7^2 + h_{10}h_7^2 + h_{15}h_7 + h_{13}h_7 + h_{15} + h_7^2 + h_7, \\
& h_{15}h_{12} + h_{13}h_7^2 + h_{12}h_7^2 + h_{10}h_7^2 + h_{15}h_7, \\
& h_{15}h_{13} + h_7^4 + h_{13}h_7^2 + h_{12}h_7^2 + h_{10}h_7^2 + h_{15}h_7 + h_{10}h_7 + h_7^2 + h_7, \\
& h_{15}^2 + h_{16}h_7^2 + h_7^4 + h_{12}h_7^2 + h_{11}h_7^2 + h_{10}h_7^2 + h_{15}h_7 + h_{10}h_7 + h_{15} + h_7^2 + h_7, \\
& h_{16}h_{10} + h_{12}h_7^2 + h_{11}h_7^2 + h_{13}h_7 + h_{16} + h_{10} + h_7 + 1, \\
& h_{16}h_{11} + h_{13}h_7^2 + h_{11}h_7^2 + h_{15}h_7 + h_{13}h_7 + h_{12}h_7 + h_{16} + h_7^2 + h_{11} + h_7 + 1, \\
& h_{16}h_{12} + h_7^4 + h_{12}h_7^2 + h_{11}h_7^2 + h_{10}h_7^2 + h_{16}h_7 + h_{12}h_7 + h_{12} + h_7, \\
& h_{16}h_{13} + h_7^4 + h_{13}h_7^2 + h_{10}h_7^2 + h_{16}h_7 + h_{10}h_7 + h_{16} + h_{15} + h_7^2 + h_{13} + 1, \\
& h_{16}h_{15} + h_{10}h_7^3 + h_7^4 + h_{12}h_7^2 + h_{11}h_7^2 + h_{10}h_7^2 + h_{15}h_7 + h_7^3 + h_{10}h_7 + h_7, \\
& h_{16}^2 + h_{11}h_7^3 + h_{16}h_7^2 + h_{15}h_7^2 + h_7^4 + h_{13}h_7^2 + h_{15}h_7 + h_{12}h_7 + h_{10}h_7 \\
& \quad + h_{16} + h_7^2 + h_7.
\end{aligned}$$

As a by-product, the smallest type I representation (relative to this choice of  $P_\infty$ ) would then be in terms of the single polynomial relating  $h_{10}$  and  $h_7$ :

$$\begin{aligned}
& h_{10}^7 + h_{10}^6h_7 + h_{10}^5(h_7^2 + 1) + h_{10}^4(h_7 + 1) + h_{10}^3(h_7^5 + h_7^4 + h_7^2 + 1) \\
& \quad + h_{10}^2(h_7^7 + h_7^6 + h_7^3 + h_7^2 + 1) + h_{10}(h_7^7 + h_7^6 + h_7^5 + 1) \\
& \quad + (h_7^{10} + h_7^9 + h_7^8 + h_7^7 + h_7^4 + h_7^3 + h_7^2 + h_7 + 1).
\end{aligned}$$

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

## References

- D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- P. Beelen, A. Garcia, and H. Stichtenoth, *Towards a classification of recursive towers of function fields over finite fields*, Finite Fields Appl. **12** (2006), no. 1, 56–77.
- W. Bosma, J. Cannon, and C. Playoust, *The MAGMA algebra system. The user language*, J. Symbolic Comput. **24** (1997), nos. 3–4, 235–265.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory. Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- J. Cannon and C. Playoust, *MAGMA: a new computer algebra system*, Euromath. Bull. **2** (1996), no. 1, 113–144.
- D. Cox, J. Little, and D. O'Shea, *Ideals, varieties, and algorithms*, third ed., Undergraduate Texts in Mathematics, Springer, Berlin, 2007, An introduction to computational algebraic geometry and commutative algebra.
- A. Garcia and H. Stichtenoth, *On the asymptotic behaviour of some towers of function fields over finite fields*, J. Number Theory **61** (1996), no. 2, 248–273.
- A. Garcia and H. Stichtenoth, *Asymptotics for the genus and the number of rational places in towers of function fields over a finite field*, Finite Fields Appl. **11** (2005), no. 3, 434–450.
- O. Geil, *Algebraic geometry codes from order domains*, this volume, 2009, pp. 121–141.
- T. Høholdt, J. van Lint, and R. Pellikaan, *Algebraic geometry of codes*, Handbook of coding theory (V. S. Pless and W.C. Huffman, eds.), Elsevier, Amsterdam, 1998, pp. 871–961.
- D. A. Leonard, *Finding the defining functions for one-point algebraic-geometry codes*, IEEE Trans. on Inf. Th. **47** (2001), no. 6, 2566–2573.
- D. A. Leonard, *A weighted module view of integral closures of affine domains of type I*, Advances in Mathematics of Communication **3** (2009), no. 1, 1–11.
- D. A. Leonard, *A tutorial on AG code decoding from a Gröbner basis perspective*, this volume, 2009, pp. 187–196.
- D. A. Leonard and R. Pellikaan, *Integral closures and weight functions over finite fields*, Finite Fields Appl. **9** (2003), no. 4, 479–504.
- J. B. Little, *Automorphisms and encoding of AG and order domain codes*, this volume, 2009, pp. 107–120.
- MAGMA, J. J. Cannon, W. Bosma (eds.), *Handbook of MAGMA functions*, edition 2.15, 2008.
- G. L. Matthews, *Viewing multipoint codes as subcodes of one-point codes*, this volume, 2009, pp. 399–402.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- S. Sakata, *The BMS algorithm and decoding of AG codes*, this volume, 2009, pp. 165–185.

# Automorphisms and Encoding of AG and Order Domain Codes

John B. Little

**Abstract** We survey some encoding methods for AG codes, focusing primarily on one approach utilizing code automorphisms. If a linear code  $C$  over  $\mathbb{F}_q$  has a finite Abelian group  $H$  as a group of automorphisms, then  $C$  has the structure of a module over a polynomial ring  $\mathcal{P}$ . This structure can be used to develop systematic encoding algorithms using Gröbner bases for modules. We illustrate these observations with several examples including geometric Goppa codes and codes from order domains.

## 1 Introduction

In order for a code to be useful in practice, it should admit efficient encoding and efficient decoding. Although most of the research effort on algebraic geometric (AG) Goppa codes from curves has focused on the decoding side, several approaches have been considered for encoding as well.

In this article we will briefly survey several approaches to encoding these codes. We will then concentrate on a systematic encoding method based on the fact that codes possessing permutation automorphisms have the structure of modules over polynomial rings. This discussion is based on Heegard et al. (1995), which shows how to use module Gröbner bases (Buchberger 1965, 1985, 2006; Mora 2009) for such codes to construct encoders. This approach is a direct generalization of the commonly-used polynomial division encoding method for cyclic codes that appears in most textbook treatments of coding theory. The module Gröbner basis furnishes an analog of the generator polynomial of a cyclic code (see Theorem 2). This particular connection between encoding and Gröbner bases was well-established previously in the case of Abelian ( $m$ -dimensional cyclic) codes. See, for instance, Poli and Huguet (1992), Sect. 6.1.6, or Chap. 9 of Cox et al. (2005). Since it relies only on the presence of suitable groups of code automorphisms, it applies to many classes of codes, including cyclic (Augot et al. 2009), quasi-cyclic (Lally 2009), Abelian, and many Goppa-type evaluation codes (Leonard 2009; Sakata 2009) from curves, higher-dimensional varieties, and order domains (Geil 2009). In the last three cases, interesting code automorphisms often arise from

---

J.B. Little

Department of Mathematics and Computer Science, College of the Holy Cross,  
Worcester, USA

e-mail: [little@mathcs.holycross.edu](mailto:little@mathcs.holycross.edu)

automorphisms of the underlying algebraic variety (see Sect. 6). Some results on implementation of this approach in hardware for the case of codes from the Hermitian curves considered in Sect. 7 have been reported by Chen and Lu (2004).

To conclude this introduction, we note that codes with sufficiently large automorphism groups may also be amenable to *permutation decoding*. In this decoding method, a fixed collection of code automorphisms is applied to the received word. If the error weight is sufficiently small, at least one of the automorphisms will move the errors out of the information positions. Then the correct information symbols can be re-encoded to accomplish the decoding. Chabanne (1992) has considered this decoding method from the point of view of Gröbner bases in the case of Abelian codes, and [L1] gives an example for a Hermitian code.

## 2 Other Encoding Methods for AG Goppa Codes

The most basic encoding method for these codes simply treats them as linear codes and uses matrix multiplication of the information word with any generator matrix  $G$  to do the encoding.

Matsumoto et al. (1997) propose a faster encoding method for the one-point residue codes  $C_{\Omega}(D, mQ)$  (and also their dual codes). This method is based on the structure of a special basis

$$\{x^i \omega_j : i \geq 0, 0 \leq j \leq a-1, v_Q(x^i \omega_j) \geq m\}$$

for the vector space of differentials  $\Omega(mQ - D)$ . Here  $x$  is an element of the Riemann-Roch space  $L(aQ)$ , where  $a$  is the smallest pole order of a nonconstant function with poles only at  $Q$ . The  $\omega_j$  are differentials in  $\Omega(-\infty Q - D)$  having the maximum valuation at  $Q$  among differentials  $\omega$  such that  $v_Q(\omega) - j$  is divisible by  $a$ . The resulting generator matrix  $G$  possesses a block factorization that can be exploited to reduce the number of multiplications involved in computing a product of the form  $xG$ . This method does not make use of Gröbner bases and yields a nonsystematic encoder.

More recently, Matsui and Mita (2007), have described a method combining discrete Fourier transforms (DFT) and Gröbner bases that yields a systematic encoder for the  $C_{\Omega}(D, mQ) = C_L(D, mQ)^{\perp}$  codes from a  $C_{a,b}$  curve. Their method works as follows. A pair  $(i, j)$  with  $0 \leq i, j \leq q-1$  can represent either the monomial  $x^i y^j$  or the point  $(\alpha^i, \alpha^j)$  where  $\alpha$  is a primitive element of the field. For simplicity, assume  $D$  is supported at points in  $(\mathbb{F}_q^*)^2$ . Partition the support into two subsets: a set of information positions  $P'$  of cardinality  $k$ , and a set of parity-checks  $P$  of cardinality  $n-k = \dim L(mQ)$ , where  $Q$  is the point at infinity on the  $C_{a,b}$  curve. A Gröbner basis  $G$  for the ideal  $I(P)$  is pre-computed. For most choices of  $P$ , the monomials in the footprint or Gröbner escalier are identified with the collection of pairs  $(i, j)$  as above with  $ai + bj \leq m$ .

To encode a given information word  $a = (a_{(i,j)} : (i, j) \in P')$ , the DFT  $A$  is computed, where  $A_{(i,j)} = f(\alpha^i, \alpha^j)$  for the polynomial

$$f(x, y) = \sum_{(\alpha^k, \alpha^l) \in P'} a_{(k,l)} x^k y^l.$$

The portion of the DFT corresponding to  $(i, j)$  with  $ai + bj \leq m$  is then extended to an array  $A'$  for all  $(i, j)$  with  $0 \leq i, j \leq q - 1$  by means of the Gröbner basis  $G$ . The difference array  $A - A'$  represents the DFT of a codeword in  $C_{\Omega}(D, mQ)$  since its syndromes corresponding to  $x^i y^j$  with  $ai + bj \leq m$  are all zero. Moreover it can be seen that the inverse DFT of  $A - A'$  provides a systematic encoding of the information  $a$ .

### 3 Automorphisms and Module Structures

We now prepare for another encoding method by introducing some general information on automorphisms of codes and module structures. The symmetric group  $S_n$  acts on  $\mathbb{F}_q^n$  by permuting the entries of vectors. A permutation *automorphism* of a linear code  $C \subset \mathbb{F}_q^n$  is an element of  $S_n$  that maps the set of codewords to itself. We will only consider code automorphisms of this type in the following.

Let  $C$  be a code that has a nontrivial Abelian group  $H$  of automorphisms. For instance, the ordinary cyclic codes and  $m$ -dimensional cyclic codes (also known as Abelian codes) are well-studied examples. For simplicity of notation, we will usually restrict to the case that  $H = \langle \sigma \rangle$  is cyclic. The generalization to the product of several cyclic groups is essentially immediate. With the restriction to cyclic groups  $H$ , cyclic codes are the most basic examples. But note that we do not assume that  $H$  acts transitively on the set of codeword components. Hence, for instance, the *quasicyclic* codes of length  $n$  also have this sort of structure (by definition,  $C$  is quasicyclic if its automorphism group contains an  $m$ -fold cyclic shift for some  $m$  dividing  $n$ ).

Let  $O_i$ ,  $i = 1, \dots, r$  be the *orbits* of the components of the codewords  $c$  under the action of  $H$ . Pick any component  $c_{i,0}$  in the  $i$ th orbit and label the components in that orbit as  $c_{i,j}$  where  $j = 0, \dots, |O_i| - 1$ . With the convention that the second index is an integer modulo  $|O_i|$ , the action of  $\sigma$  can be written as  $\sigma(c_{i,j}) = c_{i,j+1}$  for all  $i = 1, \dots, r$ , and  $j = 0, \dots, |O_i| - 1$ .

For the remainder of this article,  $\mathcal{P}$  will denote the polynomial ring in one variable,  $\mathbb{F}_q[t]$ . As usual, let  $\mathbf{e}_i$  be the  $i$ th standard basis vector in the free module  $\mathcal{P}^r$ . Then the orbit structure of the components of the codewords of  $C$  determines the submodule  $\langle(t^{|O_i|} - 1)\mathbf{e}_i : i = 1, \dots, r\rangle$  of  $\mathcal{P}^r$ . We can view the code  $C$  as subset of the quotient module

$$N = \mathcal{P}^r / \langle(t^{|O_i|} - 1)\mathbf{e}_i : i = 1, \dots, r\rangle, \quad (1)$$

via the mapping

$$\phi : C \longrightarrow N$$

$$(c_{i,j}) \longmapsto \sum_{i=1}^r \left( \sum_{j=0}^{|O_i|-1} c_{i,j} t^j \right) \mathbf{e}_i \bmod \langle (t^{|O_i|} - 1) \mathbf{e}_i : i = 1, \dots, r \rangle.$$

We have the following theorem describing the structure of the image  $\phi(C)$ .

**Theorem 1** *Let  $C$  be a linear block code over  $\mathbb{F}_q$  with a cyclic group  $H$  of automorphisms and  $\phi, N$  be as above. Then  $\phi(C)$  has the structure of a  $\mathcal{P}$ -submodule of  $N$ .*

*Proof* First  $\phi$  is linear, so  $\phi(C)$  is an  $\mathbb{F}_q$ -vector subspace of  $N$ . By the definition of  $\phi$ , if  $c \in C$  is any codeword, multiplication of  $\phi(c)$  by  $t$  yields

$$t\phi(c) = \sum_{i=1}^r \left( \sum_{j=0}^{|O_i|-1} c_{i,j} t^{j+1} \right) \mathbf{e}_i \equiv \sum_{i=1}^r \left( \sum_{j=0}^{|O_i|-1} c_{i,j-1} t^j \right) \mathbf{e}_i \bmod N = \phi(\sigma^{-1}(c)).$$

By hypothesis, this is another element of  $\phi(C)$ . Hence  $\phi(C)$  is closed under multiplication by  $t$ , hence under multiplication by all polynomials in  $\mathcal{P}$ . It follows that  $\phi(C)$  is a  $\mathcal{P}$ -submodule of  $N$ .  $\square$

Note that if the theorem applies to a code  $C$ , it applies to the dual code  $C^\perp$  as well.

In Heegard et al. (1995), Little et al. (1997), and Little (1995), this essentially straightforward generalization of the usual construction showing that a cyclic code of length  $n$  over  $\mathbb{F}_q$  is an ideal in  $\mathcal{P}/\langle t^n - 1 \rangle$  was applied to some AG Goppa codes. We will present several explicit examples in Sect. 7. The article (Lally and Fitzpatrick 2001) applies the module structures described here to study quasicyclic codes (Lally 2009). The module structure is even used for some convolutional codes, as in Gluesing-Luerssen et al. (2009). Theorem 1 can also be generalized to more general finite Abelian automorphism groups. In those cases, we obtain module structures over the polynomial ring in  $s$  variables if a minimal generating set for the group  $H$  has  $s$  elements.

## 4 A Systematic Encoding Algorithm

We will now show how the theory of Gröbner bases for modules can be applied to work with these codes. Let  $M(C)$  be the submodule of  $\mathcal{P}^r$  corresponding to  $\phi(C) \subset N$  under the mapping

$$\pi : \mathcal{P}^r \longrightarrow N,$$

where  $N$  is the quotient module from (1). The key observation here is that the canonical form algorithm with respect to a Gröbner basis  $G$  for  $M(C)$  with respect to any term ordering  $\prec$  on  $\mathcal{P}^r$  can be used to produce a systematic encoder for  $C$ .

The encoding algorithm can be described succinctly using the standard and non-standard terms for  $M(C)$ . Following the general notational conventions of this volume,  $\mathbf{N}_\prec(M(C))$  will denote the *Gröbner escalier* or “footprint” of the module  $M(C)$  with respect to a term order  $\prec$ . Similarly,  $\mathbf{T}_\prec(M(C))$  will denote the leading term module of  $M(C)$ . The terms  $t^j \mathbf{e}_i \in \mathbf{N}_\prec(M(C))$  will be called the *standard terms*. The *nonstandard terms* are the  $t^j \mathbf{e}_i$  with  $j \leq |O_i| - 1$  contained in  $\mathbf{T}_\prec(M(C))$ .

In this method, the coefficients of the nonstandard terms give the information positions in the codewords, and the coefficients of the standard terms are the parity checks. The precise statement of the encoding method is given in the following theorem.

**Theorem 2** *Let  $G$  be a Gröbner basis for the module  $M(C)$  with respect to a term ordering  $\prec$  on  $\mathcal{P}^r$ . The algorithm below produces a codeword  $c$  in all cases and gives a systematic encoder for the code  $C$ .*

Input:  $G$ , the nonstandard terms  $m_i$ , information symbols  $c_i$

Output:  $c$ , a codeword

$$\begin{aligned} f &= \sum c_i m_i; \\ c &:= f - \mathbf{CanonicalForm}(f, G); \end{aligned}$$

*Proof* Since

$$\mathbf{CanonicalForm}(c) = \mathbf{CanonicalForm}(f - \mathbf{CanonicalForm}(f, G), G) = 0,$$

it follows that  $c \in M(C)$ , which means that  $c$  represents a codeword of  $C$ . The information symbols appear as coefficients of the nonstandard terms in  $f$ , but  $\mathbf{CanonicalForm}(f, G)$  is a linear combination of standard terms. The sets of non-standard and standard terms are disjoint, hence this encoder is systematic, in the sense that the information symbols appear unchanged in a subset of the codeword entries.  $\square$

Some important examples of the term orderings that can be used here are obtained as follows. First order the  $\mathbf{e}_j$  themselves; we will use

$$\mathbf{e}_1 > \mathbf{e}_2 > \cdots > \mathbf{e}_r,$$

but the opposite order is also possible and is used too. The *position over term* (or *POT*) ordering on  $\mathcal{P}^r$  is defined by

$$t^i \mathbf{e}_j \prec_{POT} t^k \mathbf{e}_\ell$$

if  $j > \ell$ , or  $j = \ell$  and  $i < k$ . Reversing the way the comparison is made, we obtain the *term over position* (or *TOP*) ordering on  $\mathcal{P}^r$ :

$$t^i \mathbf{e}_j \prec_{TOP} t^k \mathbf{e}_\ell$$

if  $i < k$ , or  $i = k$  and  $j > \ell$ .

See also Guerrini and Rimoldi (2009) for other uses of these orderings.

## 5 Complexity Comparisons

The basic encoding method described at the start of Sect. 2 requires  $kn$  products and  $(k-1)n$  sums in  $\mathbb{F}_q$  to compute the matrix product  $xG$  if  $G$  is a general, dense generator matrix. By way of comparison, the method of Matsumoto et al. (1997) described in Sect. 2 effectively reduces the storage space and the number of operations needed. However, as noted above, this encoding method is not systematic.

One potential advantage of exploiting the module structures described in Theorem 1 is that, as is true for the generator polynomial of a cyclic code, a Gröbner basis for  $M(C)$  is typically significantly smaller than a full systematic generator matrix. The exact savings in stored information (or the size of the circuit in hardware) required for the encoding depends on the particular code. However, the situation in Example 2 in Sect. 7 below is quite typical. The code  $C$  there is a [64, 44, 8] code over  $\mathbb{F}_8$ . A reduced echelon form systematic generator matrix would be a  $44 \times 64$  matrix  $G = (I|X)$  with  $X$  a  $44 \times 20$  block of potentially nonzero entries. The Gröbner basis for the module  $M(C)$  has 10 generators, which contain at most

$$5 \times 2 + 6 \times 4 + 7 \times 6 + 8 \times 7 + 9 = 141$$

nonzero, non-leading terms. The division algorithm used for encoding in Theorem 2 takes roughly the same amount of arithmetic as the matrix product  $xG$  (see Heegard et al. 1995).

The authors of Matsui and Mita (2007) conjecture that their method requires less field arithmetic than multiplication  $xG$  with a systematic generator matrix but do not prove this. The Gröbner basis for the ideal  $I(P)$  would typically be even smaller than the Gröbner basis for the module  $M(C)$  when there is a module structure.

## 6 Automorphisms of Curves and AG Goppa Codes

In Heegard et al. (1995), it was pointed out that many examples of AG Goppa codes have the module structures described in Theorem 1, hence systematic encoders as described in Theorem 2, because of the presence of automorphisms of the underlying curves. Indeed, many interesting curves with large numbers of  $\mathbb{F}_q$ -rational points also tend to have large automorphism groups.

Let  $\mathcal{X}$  be a smooth projective algebraic curve defined over  $\mathbb{F}_q$ . An automorphism of  $\mathcal{X}$  is a regular mapping from  $\mathcal{X}$  to itself with a regular inverse. An automorphism  $\sigma$  of  $\mathcal{X}$  defined over  $\mathbb{F}_q$  induces an  $\mathbb{F}_q$ -automorphism of the function field  $K = \mathbb{F}_q(\mathcal{X})$  (an isomorphism of fields from  $K$  to itself that is the identity on  $\mathbb{F}_q$ ) via  $f \mapsto f \circ \sigma^{-1}$ . The set of all automorphisms of  $\mathcal{X}$  forms a group  $\text{Aut}(\mathcal{X})$  under function composition and  $\text{Aut}(\mathcal{X})$  acts on divisors on  $\mathcal{X}$  in the obvious way:  $\sigma(\sum n_P P) = \sum n_P \sigma(P)$ .

In fact, all of the examples of automorphisms we will consider will be induced by invertible linear mappings on the ambient projective space of  $\mathcal{X}$ . If such a mapping takes the curve  $\mathcal{X}$  to itself, then it induces an automorphism of  $\mathcal{X}$ .

For instance, consider the Hermitian function fields and curves over  $\mathbb{F}_{q^2}$ . The Hermitian curve may be defined as the variety

$$\mathcal{V}(x_0^{q+1} + x_1^{q+1} + x_2^{q+1}) \subset \mathbb{P}^2,$$

where  $(x_0 : x_1 : x_2)$  is the homogeneous coordinate vector of a point in  $\mathbb{P}^2$ . In Sect. VI.3 of Stichtenoth (1993), it is shown that (in geometric language) the tangent line to this curve at an  $\mathbb{F}_{q^2}$ -rational point can be taken to the line at infinity by a linear change of coordinates in  $\mathbb{P}^2$ . When that is done, the defining equation is taken to the form given in the following

$$\mathcal{HC}_q = \mathcal{V}(x^{q+1} - y^q z - yz^q) = \{(x : y : z) \in \mathbb{P}^2 : x^{q+1} - y^q z - yz^q = 0\}. \quad (2)$$

We will use this form of the equations of the Hermitian curves. Let  $\alpha$  be a primitive element of  $\mathbb{F}_{q^2}$ . The mapping

$$\begin{aligned} \sigma : \mathbb{P}^2 &\longrightarrow \mathbb{P}^2 \\ (x : y : z) &\longmapsto (\alpha x : \alpha^{q+1} y : z) \end{aligned} \quad (3)$$

induces an automorphism of the curve  $\mathcal{HC}_q$  because it is easy to check that if the point  $(x : y : z)$  satisfies the equation in (2), the same is true of  $\sigma(x : y : z)$ .

In the construction of an AG Goppa evaluation code on a curve  $\mathcal{X}$ , recall that one begins by selecting  $\mathbb{F}_q$ -rational divisors  $D = \sum_{i=1}^n P_i$  and  $E$  with disjoint supports on  $\mathcal{X}$ . The codewords are obtained by evaluating the rational functions  $f$  in the vector space

$$L(E) = \{f : (f) + E \geq 0\} \cup \{0\}$$

at the points in  $D$ :

$$\begin{aligned} ev : L(E) &\longrightarrow \mathbb{F}_q^n \\ f &\longmapsto (f(P_1), \dots, f(P_n)). \end{aligned}$$

The image of the evaluation mapping is the AG Goppa evaluation code  $C_L(D, E)$ .

**Theorem 3** In this situation, let  $\sigma$  be an automorphism of the curve  $\mathcal{X}$  and assume the divisors  $D$  and  $E$  are fixed by  $\sigma$ . Then  $\sigma$  induces an automorphism of the code  $C_L(D, E)$ .

*Proof* Since  $\sigma$  fixes the divisor  $E$ , it follows that  $f \mapsto f \circ \sigma^{-1}$  takes  $L(E)$  to itself. Hence we can define an action of  $\sigma$  on the codewords of  $C_L(D, E)$  by

$$(f(P_1), \dots, f(P_n)) \longmapsto (f(\sigma^{-1}(P_1)), \dots, f(\sigma^{-1}(P_n))).$$

Since the divisor  $D$  is also assumed to be fixed by  $\sigma$ , this means that the points  $\{\sigma^{-1}(P_i)\}$  are a permutation of the  $\{P_i\}$ . Hence  $\sigma$  induces a permutation automorphism of the code  $C_L(D, E)$ .  $\square$

By Proposition VII.3.3 of Stichtenoth (1993), the subgroup  $\langle \sigma \rangle$  of  $\text{Aut}(\mathcal{X})$  can be viewed as a subgroup of the permutation automorphism group of  $C_L(D, E)$  whenever  $n > 2g + 2$ , where  $g$  is the genus of  $\mathcal{X}$ . Furthermore, Joyner and Ksir (2006) have given conditions under which the permutation automorphism group of  $C_L(D, E)$  is isomorphic to the subgroup of  $\text{Aut}(\mathcal{X})$  fixing  $D$  and  $E$ .

Because of these observations, Theorems 1 and 2 from Sect. 3 apply to any  $C_L(D, E)$  code from a curve  $\mathcal{X}$  with an automorphism  $\sigma$  fixing  $D$  and  $E$ , provided  $n = \deg D$  is sufficiently large. In the case of maximal length one-point codes ( $E = aQ$  for some point  $Q$ ,  $a \geq 0$ , and  $D$  the sum of the other  $\mathbb{F}_q$ -rational points), it suffices to find a  $\sigma$  defined over  $\mathbb{F}_q$  fixing  $Q$ . One usually takes  $\sigma$  with maximal order to make the number of orbits as small as possible.

## 7 Examples

In this section, we will consider a series of examples to illustrate the previous theory.

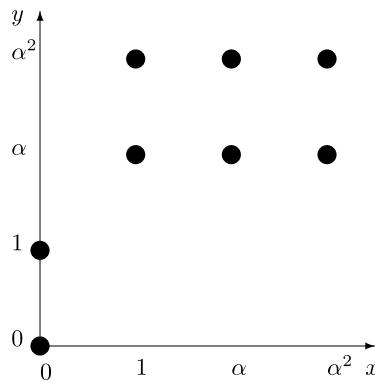
*Example 1* As shown in Sect. VII.4 of Stichtenoth (1993), the Hermitian curve  $\mathcal{HC}_q$  is a smooth plane curve of degree  $q + 1$ , hence has genus  $q(q - 1)/2$ . In addition,  $\mathcal{HC}_q$  has  $q^3 + 1$   $\mathbb{F}_{q^2}$ -rational points. There are  $q^3$  affine points. In the coordinates used in (2), there are  $q$  points on each line  $x = c$  and  $Q = (0 : 1 : 0)$  at infinity. As is well-known, this is the maximum number possible for a curve of genus  $g = q(q - 1)/2$  over  $\mathbb{F}_{q^2}$  by the Hasse–Weil bound. If  $g = g(\mathcal{X}) = q(q - 1)/2$ , then

$$|\mathcal{X}(\mathbb{F}_{q^2})| \leq 1 + q^2 + 2gq = 1 + q^2 + q(q - 1)q = q^3 + 1.$$

With  $q = 2$ , we get the picture of the  $\mathbb{F}_4$ -rational points on the Hermitian curve  $\mathcal{V}(x^3 + y^2z + yz^2)$  given below in Fig. 1.

The mapping  $\sigma$  from (3) is an automorphism of the Hermitian curve fixing  $Q$  and permuting the  $q^3$  affine  $\mathbb{F}_{q^2}$ -rational points. The subgroup of the full automorphism group generated by  $\sigma$  has order  $q^2 - 1$ . Theorem 3 and the construction from Sect. 3

**Fig. 1** The  $\mathbb{F}_4$ -rational points of the Hermitian curve with  $q = 2$



apply if we take the divisor  $E = aQ$  for any  $a \geq 0$ , and let  $D$  be the sum of the  $q^3$  affine  $\mathbb{F}_{q^2}$ -rational points, each with coefficient 1.

In the case  $q = 2$ , the automorphism  $\sigma$  is given by

$$\sigma(x : y : z) = (\alpha x : y : z)$$

(since  $\alpha^3 = 1$ ). This permutes the eight affine  $\mathbb{F}_4$ -rational points in four orbits, two of length three, and two of length one:

$$\begin{aligned} O_1 &= \{(1 : \alpha : 1), (\alpha : \alpha : 1), (\alpha^2 : \alpha : 1)\} \\ O_2 &= \{(1 : \alpha^2 : 1), (\alpha : \alpha^2 : 1), (\alpha^2 : \alpha^2 : 1)\} \\ O_3 &= \{(0 : 0 : 1)\} \\ O_4 &= \{(0 : 1 : 1)\}. \end{aligned}$$

There are similar patterns for the orbits of  $G = \langle \sigma \rangle$  on the  $\mathbb{F}_{q^2}$ -rational points in  $D$  for any  $q$ . Under  $\sigma$  there are  $q$  orbits of length  $q^2 - 1$  (all coordinates nonzero), one orbit of length  $q - 1$  (the points with  $x = 0, y \neq 0$ ), and one orbit of length 1 (a fixed point— $\{(0 : 0 : 1)\}$ ). See Heegard et al. (1995) and Little et al. (1997) for more detail on these Hermitian examples.

We next show the module structure for the code  $C = C_L(D, 3Q)$  from the Hermitian curve over  $\mathbb{F}_4$  and a Gröbner basis in detail. The affine coordinate functions  $x/z$  and  $y/z$  are elements of  $L(3Q)$ , as is  $1 = z/z$ . Hence, if we order the  $\mathbb{F}_4$ -rational points on  $\mathcal{HC}_2$  according to the orbit structure above (listing the points in  $O_1$ , then  $O_2$ , then  $O_3$ , and finally  $O_4$ ), the code  $C_L(D, 3Q)$  has generator matrix

$$\mathcal{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & 1 & \alpha & \alpha^2 & 0 & 0 \\ \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & 0 & 1 \end{pmatrix}$$

and parameters  $[n, k, d] = [8, 3, 5]$  over  $\mathbb{F}_4$  (incidentally, the best possible  $d$  for this  $n, k$  over  $\mathbb{F}_4$ ).

Under the mapping  $\phi$  from Sect. 3, the first row corresponds, for instance, to the module element

$$(1 + t + t^2, 1 + t + t^2, 1, 1).$$

With respect to the  $\prec_{POT}$  term ordering, the reduced Gröbner basis  $G$  for the submodule of  $\mathcal{P}^4$  corresponding to  $\phi(C)$  is:

$$\begin{aligned} g_1 &= (\alpha + t, \alpha + t, \alpha^2, \alpha^2) \\ g_2 &= (0, 1 + t + t^2, \alpha, \alpha^2) \\ g_3 &= (0, 0, 1 + t, 0) \\ g_4 &= (0, 0, 0, 1 + t). \end{aligned}$$

The element  $g_1$ , for instance, equals the linear combination

$$\alpha^2(1 + t + t^2, 1 + t + t^2, 1, 1) + (1 + \alpha t + \alpha^2 t^2, 1 + \alpha t + \alpha^2 t^2, 0, 0)$$

of the module elements from rows 1 and 2 of  $\mathcal{M}$  which would be computed in the course of Buchberger's algorithm. (Recall that  $\alpha^2 + \alpha + 1 = 0$  in  $\mathbb{F}_4$ .)

In the systematic encoding presented in Theorem 2, we have

- Information positions: coefficients of  $t^2 e_1, te_1, t^2 e_2$ .
- Parity checks: coefficients of  $e_1, te_2, e_2, e_3, e_4$ .

Then, to do encoding in this example, it suffices to compute remainders on division by  $G$ . For the  $\prec_{POT}$  term ordering, this amounts to ordinary polynomial divisions in each component. For example, if we want to encode

$$f = (t + \alpha t^2, \alpha^2 t^2, 0, 0),$$

it is easy to check that dividing first by  $g_1$ , then  $g_2$  yields

$$\mathbf{CanonicalForm}(f, G) = (\alpha^2, \alpha, \alpha, \alpha^2).$$

The corresponding codeword is

$$c = f - \mathbf{CanonicalForm}(f, G) = (\alpha^2 + t + \alpha t^2, \alpha + \alpha^2 t^2, \alpha, \alpha^2).$$

As indicated in the proof of Theorem 2, the information from the coefficients of  $f$  is visible immediately in the codeword  $c$ , so this is a systematic encoding method.

We note that Hermitian curves have many automorphisms besides those in the subgroup generated by  $\sigma$  above. Indeed, for some  $q$ , there are  $\sigma$  of order larger than  $q^2 - 1$  fixing  $Q$  and  $D$  (see Heegard et al. 1995). Moreover, by (Stichtenoth 1993), VII.4.6, there is also a non-Abelian subgroup  $\overline{H}$  of order  $|\overline{H}| = (q^2 - 1)q^3$  in the full automorphism group of the Hermitian curve that fixes both the point at infinity  $Q$ , and the divisor  $D$ , hence induces automorphisms of  $C_L(D, aQ)$  for all  $a$ . The elements of this subgroup can be written as the mappings

$$\tau_{\lambda, \delta, mu}(x : y : z) = (\lambda x + \delta z : \lambda^{q+1} y + \lambda \delta^q x + \mu z : z),$$

where  $\lambda \in \mathbb{F}_{q^2}^*$ , and  $(\delta : \mu : 1)$  is any affine  $\mathbb{F}_{q^2}$ -rational point on the curve. Note that

$$\tau_{\lambda, \delta, \mu}(0 : 0 : 1) = (\delta : \mu : 1).$$

This implies that  $\overline{H}$  acts *transitively* on the affine  $\mathbb{F}_{q^2}$ -rational points, or equivalently, that there is only one orbit of points under  $\overline{H}$ . Consequently, the codes  $C_L(D, aQ)$  can also be studied as *ideals* in the group algebra  $\mathbb{F}_{q^2}[\overline{H}]$ . Since  $H$  is not Abelian, however, the analogs of Gröbner bases in the group algebra do not seem to be as convenient for encoding.

AG Goppa codes from several other classes of curves with the maximal number of rational points for their genus over the given field  $\mathbb{F}_q$  can be treated in a very similar fashion.

*Example 2* Let  $q = 2^{2n+1}$ ,  $q_0 = 2^n$  and let  $\mathcal{Y}_n$  be the curve over  $\mathbb{F}_q$  defined by the affine equation

$$z^q + z = y^{q_0}(y^q + y).$$

These curves were studied by Hansen and Stichtenoth in Hansen and Stichtenoth (1990), and by a number of other authors. With  $n = 1$ , for example, the curve  $\mathcal{Y}_1$  over  $\mathbb{F}_8$  in this family has affine equation

$$z^8 + z = y^2(y^8 + y).$$

This defines a curve of degree 10, genus 14, with a single (singular) point  $Q$  at infinity. The singularity at  $Q$  is a cuspidal (unibranch) singularity—more precisely, there is only one point  $\tilde{Q}$  that lies over  $Q$  on the normalization (smooth model)  $\tilde{\mathcal{Y}}_n \rightarrow \mathcal{Y}_n$ .

From the point of view of coding theory, the curves  $\mathcal{Y}_n$  are interesting because they have as many  $\mathbb{F}_q$ -rational points as possible for a curve of their genus (however, the Hasse–Weil bound is not sharp in these cases). Indeed,  $\mathcal{Y}_n$  passes through *every point* of the affine plane over  $\mathbb{F}_q$ . For constructing AG Goppa evaluation codes from  $\mathcal{Y}_n$ , one can use  $G = a\tilde{Q}$  and  $D$  the sum of the  $q^2$   $\mathbb{F}_q$ -rational affine points, each with coefficient 1.

Letting  $\alpha$  denote a primitive element of  $\mathbb{F}_q$ , the mapping

$$\sigma(y, z) = (\alpha y, \alpha^{q_0+1} z) \tag{4}$$

restricts to an automorphism of  $\mathcal{Y}_n$ . Since  $\sigma$  fixes the divisors  $D$  and  $G = a\tilde{Q}$ ,  $\sigma$  induces an automorphism of each of the codes  $C_L(D, a\tilde{Q})$  constructed from  $\mathcal{Y}_n$ , by Theorem 3. The automorphism  $\sigma$  has order  $q - 1$  in  $\text{Aut}(\mathcal{Y}_n)$ . The following explicit example of the module structure of one of these codes comes from Little (1995).

The code  $C = C_L(D, 57\tilde{Q})$  on the Hansen–Stichtenoth curve  $\mathcal{Y}_1$  has parameters  $n = 64$ ,  $k = 44$ ,  $d = 8$  by Chen and Duursma (2003). The automorphism  $\sigma$  from (4) permutes the points of  $D$  in 10 orbits, 9 of length 7 and one of length 1.

The semigroup of pole orders at  $\tilde{Q}$  of rational functions on the curve  $\mathcal{Y}_1$  is generated by the natural numbers 8, 10, 12, 13. Moreover  $y \in L(8\tilde{Q})$ ,  $z \in L(10\tilde{Q})$ ,  $f = y^5 + z^4 \in L(12\tilde{Q})$ , and  $g = yz^4 + y^{20} + z^{16} = yz^4 + y^6 + z^2 \in L(13\tilde{Q})$  (see Hansen and Stichtenoth 1990). We use these functions to generate a basis for  $L(57\tilde{Q})$ , the normalization  $\mathbb{F}_8 = \mathbb{F}_2[\alpha]/\langle \alpha^3 + \alpha + 1 \rangle$ , and the orbit representatives  $(y, z) = (1, \alpha^6), \dots, (1, \alpha), (1, 1), (0, 1), (1, 0), (0, 0)$  (in that order). The reduced  $\prec_{POT}$  Gröbner basis  $G$  of the module  $M(C)$  has the form

$$\begin{aligned} g_1 &= (1, 0, 0, 0, 0, *, *, *, *, *) \\ g_2 &= (0, 1, 0, 0, 0, *, *, *, *, *) \\ g_3 &= (0, 0, 1, 0, 0, *, *, *, *, *) \\ g_4 &= (0, 0, 0, 1, 0, *, *, *, *, *) \\ g_5 &= (0, 0, 0, 0, 1, *, *, *, *, *) \\ g_6 &= (0, 0, 0, 0, 0, t^2 + (\alpha^2 + \alpha)t + \alpha + 1, *, *, *, *) \\ g_7 &= (0, 0, 0, 0, 0, 0, t^4 + (\alpha + 1)t^3 + (\alpha^2 + 1)t^2 + \alpha^2 + \alpha + 1, *, *, *) \\ g_8 &= (0, 0, 0, 0, 0, 0, 0, t^6 + t^5 + t^4 + t^3 + t^2 + t + 1, 0, 1) \\ g_9 &= (0, 0, 0, 0, 0, 0, 0, 0, t^7 + 1, 0) \\ g_{10} &= (0, 0, 0, 0, 0, 0, 0, 0, 0, t + 1). \end{aligned}$$

(To save space, the coefficients in the non-maximal terms are omitted. Also, it is only the maximal terms that determine the information positions and parity check positions for the code.)

Like the Hermitian curves, the Hansen–Stichtenoth curve  $\mathcal{Y}_1$  has many automorphisms besides those in the subgroup generated by the  $\sigma$  from (4) we have used. There is a (non-Abelian) subgroup  $\overline{H}$  of  $\text{Aut}(\mathcal{Y}_1)$ , of order 448 that fixes both  $a\tilde{Q}$  and  $D$ , hence induces automorphisms of each  $C_L(D, a\tilde{Q})$  code. The elements of this subgroup can be written as

$$\tau_{\lambda, \delta, \mu}(y, z) = (\lambda y + \delta, \lambda^3 z + \lambda \delta^2 y + \mu),$$

where  $\lambda \in \mathbb{F}_8^*$ , and  $(\delta, \mu)$  is any affine  $\mathbb{F}_8$ -rational point on  $\mathcal{Y}_1$ . Once again,  $\overline{H}$  acts transitively on the points of  $D$ , and the codes  $C_L(D, a\tilde{Q})$  are *ideals* in the group algebra  $\mathbb{F}_8[\overline{H}]$ .

Codes from both the Hermitian and Hansen–Stichtenoth curves can be studied with the language of order domains introduced in Høholdt et al. (1998). Higher-dimensional varieties can also be used to construct examples of order domains and generalized Goppa-type evaluation codes. See, for instance, Geil and Pellikaan (2002) and Little (2007) for a general discussion of order domains (Geil 2009), how they arise, and how the theory of Gröbner bases yields key insights about their structure and their special relevance for coding theory.

*Example 3* For instance let

$$\mathcal{HS}_q = \mathcal{V}(x_0^{q+1} + x_1^{q+1} + x_2^{q+1} + x_3^{q+1})$$

be the Hermitian surface in  $\mathbb{P}^3$ . The variety  $\mathcal{HS}_q$  has

$$(q^2 + 1)(q^3 + 1)$$

$\mathbb{F}_{q^2}$ -rational points. Changing coordinates to put a tangent plane to the surface as the plane at infinity gives the affine surface

$$\mathcal{HS}'_q = \mathcal{V}(x^{q+1} + y^{q+1} - z^q - z)$$

(whose affine coordinate ring has an order domain structure).  $\mathcal{HS}'_q$  has  $q^5$   $\mathbb{F}_{q^2}$ -rational points.  $\mathcal{HS}'_q$  also has many automorphisms, for instance

$$\sigma(x, y, z) = (\alpha x, \alpha y, \alpha^{q+1} z)$$

(of order  $= q^2 - 1$ ). This  $\sigma$  fixes the plane at infinity and permutes the  $q^5 \mathbb{F}_{q^2}$ -rational points in  $q^3 + q$  orbits of size  $q^2 - 1$ , one of size  $q - 1$ , and one of size 1. So Theorem 1 applies to all evaluation codes constructed from  $\mathcal{H}'$  and subspaces  $L \subset \mathbb{F}_{q^2}[x, y, z]$ . For instance, the code from the Hermitian surface over  $\mathbb{F}_4$  constructed by evaluating  $1, x, y, z$  has  $[n, k, d] = [32, 4, 22]$ . The minimum weight codewords come by evaluating linear polynomials that define the tangent plane at one of the  $\mathbb{F}_4$ -rational points on the surface. The minimum distance  $d = 22$  equals the best possible for a code with  $n = 32, k = 4$  over  $\mathbb{F}_4$  by Brouwer's online tables. But these codes also have Gröbner basis encoding, and good decoding algorithms because of the extra order domain structure.

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

## References

- D. Augot, E. Betti and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- H. Chabanne, *Permutation decoding of Abelian codes*, IEEE Trans. on Inf. Th. **38** (1992), no. 6, 1826–1829.

- C. Y. Chen and I. M. Duursma, *Geometric Reed–Solomon codes of length 64 and 65 over  $\mathbb{F}_8$* , IEEE Trans. on Inf. Th. **49** (2003), no. 5, 1351–1353.
- J. P. Chen and C. C. Lu, *A serial-in-serial-out hardware architecture for systematic encoding of Hermitian codes via Gröbner bases*, IEEE Trans. on Comm. **52** (2004), no. 8, 1322–1332.
- D. Cox, J. Little and D. O’Shea, *Using algebraic geometry*, second ed., Springer, Berlin, 2005.
- O. Geil, *Algebraic geometry codes from order domains*, this volume, 2009, pp. 121–141.
- O. Geil and R. Pellikaan, *On the structure of order domains*, Finite Fields Appl. **8** (2002), 369–396.
- H. Gluesing-Luerssen, B. Langfeld and W. Schmale, *A short introduction to cyclic convolutional codes*, this volume, 2009, pp. 403–408.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick’s approach to recent developments*, this volume, 2009, pp. 197–218.
- J. P. Hansen and H. Stichtenoth, *Group codes on certain algebraic curves with many rational points*, AAECC **1** (1990), no. 1, 67–77.
- C. Heegard, J. Little and K. Saints, *Systematic encoding via Gröbner bases for a class of algebraic-geometric codes*, IEEE Trans. on Inf. Th. **41** (1995), 1752–1761.
- T. Høholdt, J. H. van Lint and R. Pellikaan, *Algebraic geometry of codes*, Handbook of coding theory, vols. I, II (V. S. Pless and W.C. Huffman, eds.), North-Holland, Amsterdam, 1998, pp. 871–961.
- D. Joyner and A. Ksir, *Automorphism groups of some AG codes*, IEEE Trans. on Inf. Th. **52** (2006), no. 7, 3325–3329.
- K. Lally, *Canonical representation of quasicyclic codes using Gröbner basis theory*, this volume, 2009, pp. 351–355.
- K. Lally and P. Fitzpatrick, *Algebraic structure of quasicyclic codes*, Discrete Appl. Math. **111** (2001), nos. 1–2, 157–175.
- D. A. Leonard, *A tutorial on AG code construction from a Gröbner basis perspective*, this volume, 2009, pp. 93–106.
- J. Little, *The algebraic structure of some AG Goppa codes*, Proc. of the 33rd annual Allerton conference on communication, control, and computing, University of Illinois, Champaign, 1995, pp. 492–500.
- J. B. Little, *The ubiquity of order domains for the construction of error control codes*, Adv. Math. Commun. **1** (2007), no. 1, 151–171.
- J. Little, K. Saints and C. Heegard, *On the structure of Hermitian codes*, J. Pure Appl. Algebra **121** (1997), no. 3, 293–314.
- H. Matsui and S. Mita, *Encoding via Gröbner bases and discrete Fourier transforms for several types of algebraic codes*, Proc. of ISIT2007, 2007, pp. 2656–2660.
- R. Matsumoto, M. Oishi and K. Sakaniwa, *On the structure of Hermitian codes*, IEICE Trans. Fundamentals **121** (1997), no. 3, 293–314.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- A. Poli and L. Huguet, *Error correcting codes. Theory and applications*, Prentice Hall International, Englewood Cliffs, 1992.
- S. Sakata, *The BMS algorithm and decoding of AG codes*, this volume, 2009, pp. 165–185.
- H. Stichtenoth, *Algebraic function fields and codes*, Universitext, Springer, Berlin, 1993.

# Algebraic Geometry Codes from Order Domains

Olav Geil

**Abstract** In this tutorial we introduce order domains and study the related codes. Special attention is given to the one-point geometric Goppa codes. We show how Gröbner basis theory helps us constructing order domains as well as helps us dealing with the related codes.

## 1 Introduction

The theory of order domains is a relatively new key object in the study of algebraic geometry codes. It provides us with a simplified understanding of more well established results and gives rise to new findings and constructions. By use of the theory one can give a simple proof of the usual bounds from algebraic geometry on the minimum distance of one-point geometric Goppa codes  $C_{\mathcal{L}}(D, m\mathcal{P})$  and  $C_{\Omega}(D, m\mathcal{P})$ . In a Feng–Rao type manner it is often possible to improve on the above bounds and using the improved information one can then construct improved codes. Furthermore, order domain theory gives us an easy way of generalizing the concept of one-point geometric Goppa codes (Leonard 2009; Little 2009) to algebraic structures of higher transcendence degree. The very definition of order domains finally implies that the Berlekamp–Massey–Sakata decoding algorithm Sakata (2009a, 2009b) can be easily applied to any of the above codes for which a parity check matrix description is given.

Order domain codes can be viewed as generalizations of Reed–Solomon codes (Augot et al. 2009). Recall that Reed–Solomon codes are defined from the polynomial ring  $R = \mathbb{F}_q[X]$ . Denoting  $\mathbb{F}_q = \{P_1, \dots, P_q\}$  the Reed–Solomon code with parameters  $[n = q, k, d = n - k + 1]$  (here of course  $k \leq q$  must hold) is

$$\{(F(P_1), \dots, F(P_q)) \mid \deg(F) < k\} = \{(F(P_1), \dots, F(P_q)) \mid \deg(F) < n - k\}^{\perp}.$$

The parameters of the Reed–Solomon code are easily demonstrated by using the fact that a polynomial of degree  $t$  can have at most  $t$  zeros. From an order domain perspective  $R = \mathbb{F}_q[X]$  is an order domain and the degree function  $\rho : \mathbb{F}_q[X] \rightarrow \mathbb{N}_0 \cup \{-\infty\}$ ,  $\rho(F(X)) = \deg(F)$  is a weight function. The codes are defined by using the map  $\varphi : \mathbb{F}_q[X] \rightarrow (\mathbb{F}_q)^n = \mathbb{F}_q^n$ ,  $\varphi(F(X)) = (F(P_1), \dots, F(P_n))$  and the parameters can be found by studying the properties of  $\rho$  and  $\varphi$ .

---

O. Geil

Department of Mathematical Sciences, Aalborg University, Aalborg, Denmark

e-mail: [olav@math.aau.dk](mailto:olav@math.aau.dk)

The next most simple examples of order domain codes are the Hermitian codes, the improved Hermitian codes, the generalized Reed–Muller codes and the improved generalized Reed–Muller codes known as hyperbolic codes. Throughout the paper we will investigate the behavior of these codes and the nature of the algebraic structures used for their construction. The idea is to make it easier for the reader to grasp the general theory of order domains.

The paper is organized as follows. In Sect. 2 we explain what is an order domain with a weight function. Then in Sect. 3 we introduce codes defined from order domains and estimate their parameters. Section 4 is concerned with one-point geometric Goppa codes. In Sect. 5 we show how to easily construct order domains by use of Gröbner basis methods (Buchberger 1965, 1985, 2006), and in Sect. 6 we see how Gröbner basis theory helps us construct the corresponding codes. Finally, in Sect. 7 we briefly discuss the connection to valuation theory. Being a tutorial the paper contains only known results. Our presentation mostly relies on Høholdt et al. (1998), O’Sullivan (2001), Geil and Pellikaan (2002) and Andersen and Geil (2008).

## 2 Order Domains with Weight Functions

We start our treatment of weight functions by considering in detail the Hermitian order domain.

*Example 1* Consider the Hermitian polynomial  $X^{q+1} - Y^q - Y$  and let  $I$  be the ideal  $I = \langle X^{q+1} - Y^q - Y \rangle \subseteq \mathbb{F}_{q^2}[X, Y]$ . The Hermitian order domain is  $R = \mathbb{F}_{q^2}[X, Y]/I$ . As will be shown in Example 9 of Sect. 5 one possible basis for  $R$  as a vector space over  $\mathbb{F}_{q^2}$  is  $\mathcal{B} = \{X^i Y^j + I \mid 0 \leq i, 0 \leq j < q\}$ . Denote by  $\mathcal{M}(X, Y)$  the monomials in  $X$  and  $Y$  and define a function  $w : \mathcal{M}(X, Y) \rightarrow \mathbb{N}_0$  by  $w(X^i Y^j) = iq + j(q + 1)$ . The value  $w(X^i Y^j)$  will be called the weight of  $X^i Y^j$ . We observe that the restriction of  $w$  to  $\{X^i Y^j \mid 0 \leq i, 0 \leq j < q\}$  is injective. Therefore  $w$  induces a bijective map  $\rho : \mathcal{B} \rightarrow \langle q, q + 1 \rangle$  by  $\rho(X^i Y^j + I) = w(X^i Y^j)$ . Here,  $\langle q, q + 1 \rangle$  denotes the numerical semigroup generated by  $q$  and  $q + 1$ . As  $\mathcal{B}$  is a basis it is clear that any element  $f \in R$  can be uniquely described as  $f = F(X, Y) + I$  where every monomial  $X^i Y^j$  in the support of  $F(X, Y)$  satisfies  $0 \leq i, 0 \leq j < q$ . This unique polynomial will be called the canonical representative of  $f$ . Given a description  $F'(X, Y) + I$  not of this form we can substitute repeatedly any occurrences of  $Y^q$  with  $X^{q+1} - Y$  and thereby eventually get a description  $F(X, Y) + I$  of the desired form. We can now extend  $\rho$  to a function on  $R$ . Let  $F(X, Y)$  be the canonical representative of  $f$ , we define  $\rho(0 + I) = -\infty$  and

$$\rho(f) = \max\{w(M) \mid M \text{ is in the support of } F(X, Y)\}$$

when  $F(X, Y) \neq 0$ . Observe, that  $F(X, Y)$  either is 0 or has precisely one monomial of highest weight in its support. Given two nonzero elements  $f_1 = F_1(X, Y) + I$ ,  $f_2 = F_2(X, Y) + I$  where  $F_1(X, Y)$  respectively  $F_2(X, Y)$  is the canonical representative of  $f_1$  respectively  $f_2$  we conclude that there will be exactly one monomial

in  $G'(X, Y) = F_1(X, Y)F_2(X, Y)$  of highest weight and this highest weight equals  $\rho(f_1) + \rho(f_2)$ . Substituting in  $G'(X, Y)$  repeatedly any occurrences of  $Y^q$  with  $X^{q+1} - Y$  will as mentioned above eventually give a description  $G(X, Y) + I$  of  $G'(X, Y) + I$  such that every monomial  $X^i Y^j$  in  $G(X, Y)$  satisfies  $0 \leq i, 0 \leq j < q$ . The crucial observation now is that by induction at any step in this reduction the derived polynomial will have exactly one monomial in its support of highest weight and this weight equals

$$\max\{w(M) \mid M \text{ is in the support of } F_1(X, Y)F_2(X, Y)\} = \rho(f_1) + \rho(f_2).$$

The above observation follows from the fact that the polynomial  $X^{q+1} - Y$  replacing  $Y^q$  has exactly one monomial in its support of highest weight and from the fact that this weight equals  $w(Y^q)$ . As a consequence of the above observation we get the nice result  $\rho(f_1 f_2) = \rho(f_1) + \rho(f_2)$ .

The function  $\rho : R \rightarrow \langle q, q+1 \rangle \cup \{-\infty\}$  described in Example 1 is an instance of a weight function. Keeping this in mind should make it easier to understand the general definition of a weight function. We will need the concept of a well-behaving basis.

**Definition 1** Let  $\mathbb{F}$  be a field and let  $R$  be a  $\mathbb{F}$ -algebra. Let  $\Gamma \subseteq \mathbb{N}_0^r$  be a semigroup and assume  $<_{\mathbb{N}_0^r}$  is a term ordering on  $\mathbb{N}_0^r$ . Given a basis  $\mathcal{B}$  for  $R$  and a bijective map  $\rho : \mathcal{B} \rightarrow \Gamma$  we will write  $\mathcal{B} = \{f_\lambda \mid \lambda \in \Gamma\}$  (with the underlying assumption that  $\rho(f_\lambda) = \lambda$ ) and for all  $\lambda \in \Gamma$  define  $R_\lambda = \text{Span}_{\mathbb{F}}\{f_\gamma \mid \gamma \leq_{\mathbb{N}_0^r} \lambda\}$ . We also define  $R_{-\infty} = \{0\}$ . The ordered basis  $\mathcal{B}$  is called a well-behaving basis if for all  $\lambda, \gamma \in \Gamma$  we have  $f_\lambda f_\gamma \in R_{\lambda+\gamma}$  but  $f_\lambda f_\gamma \notin R_\delta$  for any  $\delta <_{\mathbb{N}_0^r} \lambda + \gamma$ .

The basis  $\mathcal{B}$  from Example 1 clearly satisfies the conditions of Definition 1 and is therefore a well-behaving basis. Just as was the case in Example 1 the ordered basis from Definition 1 induces a map  $\rho : R \rightarrow \Gamma \cup \{-\infty\}$ . We have

**Definition 2** Let  $\mathcal{B} = \{f_\lambda \mid \lambda \in \Gamma\}$  be a well-behaving basis. If  $f = 0$  we define  $\rho(f) = -\infty$ . For nonzero  $f$  we consider the expansion  $f = \sum_{i=1}^t k_i f_{\lambda_i}$ ,  $k_i \in \mathbb{F} \setminus \{0\}$  for  $i = 1, \dots, t$  and  $\lambda_i \neq \lambda_j$  for  $i \neq j$ . We then define  $\rho(f) = \max\{\lambda_i \mid i = 1, \dots, t\}$ . A function  $\rho$  defined in this way is called a weight function.

*Remark 1* It is not hard to show that Definition 2 is equivalent to the following characterization. Let  $<_{\mathbb{N}_0^r}$  be a term ordering on  $\mathbb{N}_0^r$  and let  $\Gamma, \Gamma \subseteq \mathbb{N}_0^r$  be a semigroup. For all  $\lambda \in \Gamma$  define  $\lambda + (-\infty) = -\infty$ . A surjective map  $\rho : R \rightarrow \Gamma \cup \{-\infty\}$  is called a weight function if for all  $f, g, h \in R$  we have

- (W.0)  $\rho(f) = -\infty \Leftrightarrow f = 0$
- (W.1)  $\rho(af) = \rho(f)$  for all  $a \in \mathbb{F} \setminus \{0\}$
- (W.2)  $\rho(f+g) \leq_{\mathbb{N}_0^r} \max\{\rho(f), \rho(g)\}$
- (W.3)  $\rho(fg) = \rho(f) + \rho(g)$

- (W.4) If  $f$  and  $g$  are nonzero and  $\rho(f) = \rho(g)$  then there exists a nonzero  $a \in \mathbb{F}$  such that  $\rho(f - ag) <_{\mathbb{N}_0^r} \rho(g)$

*Remark 2* Weight functions are special cases of order functions. The general definition of order functions (Geil and Pellikaan 2002, Definition 2.1) calls for the following changes in the characterization in Remark 1. We start by replacing  $(\Gamma \subseteq \mathbb{N}_0^r, <_{\mathbb{N}_0^r})$  by any well-order  $(\Gamma, <_\Gamma)$ . Then we replace (W.3) with

- (O.3) If  $\rho(f) <_\Gamma \rho(g)$  and  $h \neq 0$  then  $\rho(fh) <_\Gamma \rho(gh)$ .

A  $\mathbb{F}$ -algebra with an order function is called an order domain (over  $\mathbb{F}$ ).

*Remark 3* The original definition of an order function in Høholdt et al. (1998, Definition 3.4) is a little less general than the definition in Geil and Pellikaan (2002). More precisely it is in Høholdt et al. (1998) required that  $\Gamma \subseteq \mathbb{N}_0$  and therefore automatically  $<_\Gamma$  becomes the usual ordering  $<$  on  $\mathbb{N}_0$ . For a weight function to be an order function under this description one must require that the ordering  $<_{\mathbb{N}_0^r}$  on  $\mathbb{N}_0^r$  is isomorphic to the ordering  $<$  on  $\mathbb{N}_0$ . We will see later in the paper that mapping to  $\mathbb{N}_0^r$  rather than just to  $\mathbb{N}_0$  gives us a method for dealing with order domains of transcendence degree more than one.

In Sect. 5 we will observe that all order functions relevant in coding theory are actually weight functions. Although Definitions 1 and 2 are not very involved they will be general enough to help us construct quite a large class of algebraic geometry codes. The following example describes the algebraic structure needed in the construction of one-point geometric Goppa codes.

*Example 2* Let  $\mathcal{P}$  be a rational place in an algebraic function field of one variable and let  $v_{\mathcal{P}}$  be the valuation corresponding to  $\mathcal{P}$ . Then  $R = \bigcup_{m=0}^{\infty} \mathcal{L}(m\mathcal{P})$  is an order domain with a weight function given by  $\rho(x) = -v_{\mathcal{P}}(x)$  for any  $x \in R$ .

The next example describes the algebraic structures needed in the construction of generalized Reed–Muller codes and hyperbolic codes.

*Example 3* Let  $R = \mathbb{F}_q[X_1, \dots, X_m]$  and fix any term ordering  $<_{\mathbb{N}_0^m}$  on  $\mathbb{N}_0^m$ . Define a weight function  $\rho : R \rightarrow \mathbb{N}_0^m \cup \{-\infty\}$  as follows. We have  $\rho(0) = -\infty$  and for nonzero  $F(X_1, \dots, X_m)$  we have

$$\begin{aligned} \rho(F(X_1, \dots, X_m)) = \max\{(\alpha_1, \dots, \alpha_m) \mid X_1^{\alpha_1} \cdots X_m^{\alpha_m} \text{ is in} \\ \text{the support of } F(X_1, \dots, X_m)\}. \end{aligned}$$

Here max is taken with respect to the ordering  $<_{\mathbb{N}_0^m}$ . An obvious choice of a well-behaving basis is  $\mathcal{B} = \{X_1^{i_1} \cdots X_m^{i_m} \mid 0 \leq i_1, \dots, 0 \leq i_m\}$ . For  $m \geq 2$  there are infinitely many term orderings on  $\mathbb{N}_0^m$  and therefore a polynomial ring in more variables possesses infinitely many weight functions.

The construction of order domains and weight functions in Examples 1 and 3 was not very involved whereas the construction in Example 2 relies on algebraic function field theory or algebraic geometry. Later in this paper we present a method for constructing order domains and weight functions by use of only simple Gröbner basis theoretical methods. The construction is very similar to the one in Example 1 and deals with any weight function for which the semigroup  $\Gamma$  is finitely generated. The Gröbner basis construction provides us in particular with much more sophisticated examples of order domains of higher transcendence degree than the one in Example 3. However, before continuing our study of order domains we should get involved with the codes. This is done in the next section.

### 3 Codes from Order Domains

With a reference to Høholdt et al. (1998, p. 873) by an algebraic geometry code we mean a code that is defined from an algebraic geometry structure by use of some kind of evaluation map.<sup>1</sup> We now consider algebraic geometry codes related to order domains over finite fields  $\mathbb{F}_q$ . As we will see this set of codes contains in particular one-point geometric Goppa codes (Leonard 2009; Little 2009; Sakata 2009b).

We start by recalling that the component-wise product in  $\mathbb{F}_q^n$  is given by  $(a_1, \dots, a_n) * (b_1, \dots, b_n) = (a_1 b_1, \dots, a_n b_n)$ . With this product  $\mathbb{F}_q^n$  becomes an  $\mathbb{F}_q$ -algebra. For the code constructions we consider any map  $\varphi$  of the following type.

**Definition 3** Let  $R$  be an  $\mathbb{F}_q$ -algebra. A surjective map  $\varphi : R \rightarrow \mathbb{F}_q^n$  is called a morphism of  $\mathbb{F}_q$ -algebras if  $\varphi$  is  $\mathbb{F}_q$ -linear and  $\varphi(fg) = \varphi(f) * \varphi(g)$  holds for all  $f, g \in R$ .

*Example 4* By using the fact that  $\alpha^{q+1}$  is the norm map from  $\mathbb{F}_{q^2}$  to  $\mathbb{F}_q$  and by using the fact that  $\alpha^q + \alpha$  is the trace map from  $\mathbb{F}_{q^2}$  to  $\mathbb{F}_q$  the zeros of the Hermitian polynomial  $X^{q+1} - Y^q - Y$  can be determined. There are  $n = q^3$  of them. Hence, if  $I = \langle X^{q+1} - Y^q - Y \rangle \subseteq \mathbb{F}_{q^2}(X, Y)$  then the variety  $\mathcal{V}_{\mathbb{F}_{q^2}}(I)$  consists of  $q^3$  points, say  $P_1, \dots, P_{q^3}$ . The evaluation map  $\varphi : R = \mathbb{F}_{q^2}[X, Y]/I \rightarrow \mathbb{F}_{q^2}^n$  given by  $\varphi(F(X, Y) + I) = (F(P_1), \dots, F(P_{q^3}))$  is a morphism of  $\mathbb{F}_{q^2}$ -algebras. Recall from the previous section, that  $\mathcal{B} = \{X^i Y^j + I \mid 0 \leq i, 0 \leq j < q\}$  is a well-behaving basis for the Hermitian order domain. The most natural codes from the Hermitian order domains are the one-point geometric Goppa codes

$$\begin{aligned} E(s) &= \varphi(R_s) = \text{Span}_{\mathbb{F}_{q^2}} \{ \varphi(X^i Y^j + I) \mid 0 \leq i, 0 \leq j < q, w(X^i Y^j) \leq s \} \\ C(s) &= (E(s))^\perp \end{aligned}$$

---

<sup>1</sup>In recent literature the name geometric Goppa code is often replaced with the name algebraic geometric code. Hence, algebraic geometry codes and algebraic geometric codes are not the same and the word AG-code should be used with caution.

In the remainder of this section let  $R$  be an order domain with a weight function  $\rho : R \rightarrow \Gamma \cup \{-\infty\}$ ,  $\Gamma \subseteq \mathbb{N}_0^r$  and let  $\varphi : R \rightarrow \mathbb{F}_q^n$  be a morphism between  $\mathbb{F}_q$  algebras. Consider the following very general problem. Let  $\mathcal{B} = \{f_\lambda \mid \lambda \in \Gamma\}$  be a well-behaving basis and consider a subset  $\mathcal{B}' \subseteq \mathcal{B}$ . We would like to know what is the minimum distance of  $\text{Span}_{\mathbb{F}_q}\{\varphi(f) \mid f \in \mathcal{B}'\}$  and what is the minimum distance of  $(\text{Span}_{\mathbb{F}_q}\{\varphi(f) \mid f \in \mathcal{B}'\})^\perp$ . As we shall demonstrate the semigroup  $\Gamma$  holds a lot of information about these questions. The information even suggests clever ways of choosing  $\mathcal{B}'$ . We start our investigation by stating some useful lemmas and propositions.

**Definition 4** For  $\lambda \in \Gamma$  define

$$N(\lambda) = \{\eta \in \Gamma \mid \exists \beta \in \Gamma \text{ with } \eta + \beta = \lambda\} \quad \text{and} \quad \mu(\lambda) = \#N(\lambda).$$

*Remark 4* In Høholdt et al. (1998, Definition 4.8)  $N(\lambda)$  is defined slightly differently and  $\#N(\lambda)$  is called  $\nu(\lambda)$ . To apply the definition in Høholdt et al. (1998) to the weight functions described in the present paper we must require that the well-order  $(\mathbb{N}_0^r, <_{\mathbb{N}_0^r})$  is isomorphic to the well-order  $(\mathbb{N}_0, <)$ . In other words, for every nonzero  $\lambda \in \Gamma$  there exists a maximal element  $\gamma \in \Gamma$  for which  $\gamma <_{\mathbb{N}_0^r} \lambda$  holds. We then have  $\nu(\gamma) = \mu(\lambda)$ . The main motivation for using Definition 4 rather than Høholdt et al. (1998, Definition 4.8) is that in this way  $N(\lambda)$  and therefore also the size of it becomes independent on the term ordering on  $\mathbb{N}_0^r$ . Given  $r > 1$  and two different term orderings using Høholdt et al. (1998, Definition 4.8) we would have to keep track of two different functions  $\nu$ .

**Lemma 1** *Given a nonzero word  $\mathbf{c} \in \mathbb{F}_q^n$  let  $\lambda \in \Gamma$  be the (unique) element such that  $\mathbf{c} \cdot \varphi(f_\lambda) \neq 0$  but  $\mathbf{c} \cdot \varphi(f_\gamma) = 0$  for all  $\gamma <_{\mathbb{N}_0^r} \lambda$ . Then  $\mathbf{c} \cdot \varphi(f) \neq 0$  for all  $f$  with  $\rho(f) = \lambda$  and  $\mathbf{c} \cdot \varphi(f) = 0$  for all  $f$  with  $\rho(f) <_{\mathbb{N}_0^r} \lambda$ .*

*Proof* The lemma follows by linearity of  $\varphi$ . □

**Proposition 1** *Given a nonzero word  $\mathbf{c} \in \mathbb{F}_q^n$  let  $\lambda$  be as in Lemma 1. The Hamming weight of  $\mathbf{c}$  satisfies  $w_H(\mathbf{c}) \geq \mu(\lambda)$ .*

*Proof* Let  $N(\lambda) = \{i_1, \dots, i_\mu\}$ . By the definition of  $N(\lambda)$  for every  $i_s$ ,  $s = 1, \dots, \mu$  there exists a  $j_s \in \Gamma$  with  $i_s + j_s = \lambda$ . Consider any nonzero linear combination of  $f_{i_1}, \dots, f_{i_\mu}$  over  $\mathbb{F}_q$ ,

$$r = \sum_{s=1}^{\mu} k_s f_{i_s}.$$

Let  $t \in \{1, \dots, \mu\}$  be the maximal value such that  $k_t \neq 0$ . That is,  $\rho(r) = i_t$ . From (W.3) in Remark 1 we conclude that  $\rho(r f_{j_t}) = \lambda$  and therefore by Lemma 1  $\mathbf{c} \cdot \varphi(r f_{j_t}) \neq 0$  must hold. Using the fact that  $\varphi$  is a morphism, we get

$$\begin{aligned}
\mathbf{c} \cdot (\varphi(r) * \varphi(f_{j_i})) \neq 0 &\Rightarrow (\mathbf{c} * \varphi(r)) \cdot \varphi(f_{j_i}) \neq 0 \\
&\Rightarrow \mathbf{c} * \varphi(r) \neq \mathbf{0} \\
&\Rightarrow \mathbf{c} * \left( \sum_{s=1}^{\mu} k_s \varphi(f_{i_s}) \right) \neq \mathbf{0}.
\end{aligned} \tag{1}$$

Aiming for a contradiction, assume

$$w_H(\mathbf{c}) < \mu. \tag{2}$$

Without loss of generality we assume that the nonzero entries of  $\mathbf{c}$  are among the first  $\mu - 1$  entries. Recall, that (1) holds for any choice of  $k_1, \dots, k_\mu$  not all zero. We will choose  $k_1, \dots, k_\mu$  not all zero in such a way that the first  $\mu - 1$  entries of  $\sum_{s=1}^{\mu} k_s \varphi(f_{i_s})$  are zero. This is possible due to a standard linear algebra result. But then (1) can not be true and therefore the assumption (2) was wrong.  $\square$

To state the next lemma we will need two definitions.

**Definition 5** Let  $\alpha(1) = \mathbf{0}$ . For  $i = 2, 3, \dots, n$  recursively define  $\alpha(i)$  to be the smallest element in  $\Gamma$  that is greater than  $\alpha(1), \alpha(2), \dots, \alpha(i-1)$  and satisfies  $\varphi(R_\gamma) \subsetneq \varphi(R_{\alpha(i)})$  for all  $\gamma <_{\mathbb{N}_0^r} \alpha(i)$ . Write  $\Delta(R, \rho, \varphi) = \{\alpha(1), \dots, \alpha(n)\}$ .

It is clear that the set  $\{\varphi(f_{\alpha(1)}), \dots, \varphi(f_{\alpha(n)})\}$  constitutes a basis for  $\mathbb{F}_q^n$  as a vector space over  $\mathbb{F}_q$ . Before proceeding we illustrate the definition with two examples.

*Example 5* This is a continuation of Example 4. Recall, that  $\mathcal{B} = \{X^i Y^j + I \mid 0 \leq i, 0 \leq j < q\}$  constitutes a well-behaving basis for the Hermitian order domain. Clearly,  $\varphi(X^i Y^j + I) = \varphi(X^{i+q^2-1} Y^j + I)$  for all  $i > 0$  and therefore the elements in  $\Delta(R, \rho, \varphi)$  need to be of the form  $iq + j(q+1)$  with  $0 \leq i < q^2$  and  $0 \leq j < q$ . But there are exactly  $n = \dim(\varphi(R)) = q^3$  such numbers and therefore  $\Delta(R, \rho, \varphi) = \{iq + j(q+1) \mid 0 \leq i < q^2, 0 \leq j < q\}$ .

*Example 6* This is a continuation of Example 3 where we considered a family of weight functions on the order domain  $R = \mathbb{F}_q[X_1, \dots, X_m]$ . Denote  $\{P_1, \dots, P_{q^m}\} = \mathbb{F}_q^m$  and write  $n = q^m$ . Define  $\varphi : R \rightarrow \mathbb{F}_q^n$  by  $\varphi(F(X_1, \dots, X_m)) = (F(P_1), \dots, F(P_n))$ . Recall, that for any of the described weight functions  $\mathcal{B} = \{X_1^{i_1} \cdots X_m^{i_m} \mid 0 \leq i_1, \dots, 0 \leq i_m\}$  is a well-behaving basis. Clearly,  $\varphi(X_1^{i_1} \cdots X_m^{i_m}) = \varphi(X_1^{i_1+q} \cdots X_m^{i_m}) = \cdots = \varphi(X_1^{i_1} \cdots X_m^{i_m+q})$  and therefore the elements in  $\Delta(R, \rho, \varphi)$  need to be of the form  $(i_1, \dots, i_m)$  with  $0 \leq i_1 < q, \dots, 0 \leq i_m < q$ . But there are exactly  $n = \dim(\varphi(R)) = q^m$  such values and therefore  $\Delta(R, \rho, \varphi) = \{(i_1, \dots, i_m) \mid 0 \leq i_1 < q, \dots, 0 \leq i_m < q\}$ .

**Definition 6** For  $\alpha \in \Delta(R, \rho, \varphi)$  let

$$M(\alpha) = \{\lambda \in \Delta(R, \rho, \varphi) \mid \exists \beta \in \Gamma \text{ such that } \alpha + \beta = \lambda\} \quad \text{and} \quad \sigma(\alpha) = \#M(\alpha).$$

**Proposition 2** Let  $\{\alpha(1), \dots, \alpha(n)\}$  be as in Definition 5. Given a nonzero word  $\mathbf{c}$  expand it as follows

$$\mathbf{c} = \sum_{s=1}^n k_s \varphi(f_{\alpha(s)}), \quad k_1, \dots, k_n \in \mathbb{F}_q.$$

Let  $t$  be the maximal value such that  $k_t$  is nonzero. The Hamming weight of  $\mathbf{c}$  satisfies  $w_H(\mathbf{c}) \geq \sigma(\alpha(t))$ .

*Proof* Write  $\sigma(\alpha(t)) = \sigma$ ,  $M(\alpha(t)) = \{\lambda_1, \dots, \lambda_\sigma\}$  and let  $\beta_1, \dots, \beta_\sigma$  be such that  $\alpha(t) + \beta_1 = \lambda_1, \dots, \alpha(t) + \beta_\sigma = \lambda_\sigma$ . Writing  $f = \sum_{s=1}^t k_s f_{\alpha(s)}$  we get  $\mathbf{c} = \varphi(f)$ . Now by assumption  $k_t$  is nonzero but  $k_s = 0$  for  $t < s$  and therefore  $\rho(f) = \alpha(t)$  follows. We get  $\rho(f f \beta_1) = \lambda_1, \dots, \rho(f f \beta_\sigma) = \lambda_\sigma$ . But then from the definition of  $\Delta(R, \rho, \varphi)$  we conclude that  $\varphi(f f \beta_1), \dots, \varphi(f f \beta_\sigma)$  are linearly independent. In other words  $\mathbf{c} * \varphi(f \beta_1), \dots, \mathbf{c} * \varphi(f \beta_\sigma)$  are linearly independent. However, the vector space  $\{\mathbf{b} \mid \text{there exists an } \mathbf{a} \text{ such that } \mathbf{b} = \mathbf{c} * \mathbf{a}\}$  is clearly of dimension exactly  $w_H(\mathbf{c})$  and therefore  $\sigma \leq w_H(\mathbf{c})$  must hold.  $\square$

With Propositions 1 and 2 in hand we are now able to deal with some very large classes of codes. We start by stating a very general theorem.

**Theorem 1** Given numbers  $i_1, \dots, i_t$  with  $1 \leq i_1 < \dots < i_t \leq n$  consider the corresponding elements  $f_{\alpha(i_1)}, \dots, f_{\alpha(i_t)} \in \mathcal{B}$ . The code

$$\text{Span}_{\mathbb{F}_q} \{\varphi(f_{\alpha(i_1)}), \dots, \varphi(f_{\alpha(i_t)})\}$$

is of dimension  $t$  and has minimum distance at least

$$\min\{\sigma(\alpha(i_s)) \mid s = 1, \dots, t\}.$$

The dual code is of dimension  $n - t$  and has minimum distance at least

$$\min\{\mu(\alpha(i)) \mid i \in \{1, \dots, n\} \setminus \{i_1, \dots, i_t\}\} \tag{3}$$

$$\geq \min\{\mu(\lambda) \mid \lambda \in \Gamma \setminus \{\alpha(i_1), \dots, \alpha(i_t)\}\}. \tag{4}$$

*Proof* The result concerning the first code follows immediately from Proposition 2. Concerning the dual code we observe that since  $\{\varphi(f_{\alpha(1)}), \dots, \varphi(f_{\alpha(n)})\}$  constitutes a basis for  $\mathbb{F}_q^n$ , a nonzero word  $\mathbf{c}$  will have to satisfy  $\mathbf{c} \cdot \varphi(f_\alpha) \neq 0$  for some  $\alpha \in \Delta(R, \rho, \varphi)$ . Combining Lemma 1 with the definition of  $\Delta(R, \rho, \varphi)$  we see that the smallest value  $\lambda \in \Gamma$  such that  $\mathbf{c} \cdot \varphi(f_\lambda) \neq 0$  is an element in  $\Delta(R, \rho, \varphi)$ . But by construction of the dual code  $\mathbf{c} \cdot \varphi(f_{\alpha(i_1)}) = \dots = \mathbf{c} \cdot \varphi(f_{\alpha(i_t)}) = 0$  holds. Hence, the smallest possible  $\lambda \in \Gamma$  such that  $\mathbf{c} \cdot \varphi(f_\lambda) \neq 0$  must be contained in  $\Delta(R, \rho, \varphi) \setminus \{\alpha(i_1), \dots, \alpha(i_t)\}$ . The estimate (3) of the minimum distance of the dual code now follows immediately from Proposition 1. The number in (3) is clearly larger than or equal to the number in (4).  $\square$

The estimates (3) and (4) of the minimum distance of the dual code are known as the order bound. They are instance of the Feng–Rao bound. Consider the following particular classes of codes.

### Definition 7

$$\begin{aligned} E(\lambda) &= \varphi(R_\lambda) = \text{Span}_{\mathbb{F}_q}\{\varphi(f_\gamma) \mid \gamma \leq_{\mathbb{N}_0^r} \lambda\} \\ \tilde{E}_\varphi(s) &= \text{Span}_{\mathbb{F}_q}\{\varphi(f_\lambda) \mid \lambda \in \Delta(R, \rho, \varphi) \text{ and } \sigma(\lambda) \geq s\} \\ C(\lambda) &= (E(\lambda))^\perp = \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{c} \cdot \varphi(f_\gamma) = 0 \text{ for all } \gamma \leq_{\mathbb{N}_0^r} \lambda\} \\ \tilde{C}(s) &= \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{c} \cdot \varphi(f_\gamma) = 0 \text{ for all } \gamma \text{ with } \mu(\gamma) < s\} \\ \tilde{C}_\varphi(s) &= \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{c} \cdot \varphi(f_\gamma) = 0 \text{ for all } \gamma \in \Delta(R, \rho, \varphi) \text{ with } \mu(\gamma) < s\} \end{aligned}$$

Note that the codes in Example 4 were of the type  $E(\lambda)$  and  $C(\lambda)$ . In larger generality we see from Example 2 that one-point geometric Goppa codes  $C_{\mathcal{L}}(D, m\mathcal{P})$  are codes  $E(\lambda)$  from order domains with a numerical semigroup. Similarly one-point geometric Goppa codes  $C_{\Omega}(D, m\mathcal{P})$  are codes  $C(\lambda)$  from order domains with a numerical semigroup. The codes  $\tilde{E}_\varphi(s)$ ,  $\tilde{C}(s)$  and  $\tilde{C}_\varphi(s)$  are said to be improved codes. This name is justified by the following theorem.

**Theorem 2** *We have*

$$\begin{aligned} E(\lambda) &= \text{Span}_{\mathbb{F}_q}\{\varphi(f_\gamma) \mid \gamma \in \Delta(R, \rho, \varphi) \text{ and } \gamma \leq_{\mathbb{N}_0^r} \lambda\} \\ C(\lambda) &= \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{c} \cdot \varphi(f_\gamma) = 0 \text{ for all } \gamma \in \Delta(R, \rho, \varphi) \text{ with } \gamma \leq_{\mathbb{N}_0^r} \lambda\} \end{aligned}$$

*The minimum distances of the codes in Definition 7 satisfy*

$$d(E(\lambda)) \geq \min\{\sigma(\gamma) \mid \gamma \in \Delta(R, \rho, \varphi) \text{ and } \gamma \leq_{\mathbb{N}_0^r} \lambda\} \quad (5)$$

$$d(\tilde{E}_\varphi(s)) \geq s$$

$$d(C(\lambda)) \geq \min\{\mu(\eta) \mid \eta \in \Delta(R, \rho, \varphi), \lambda <_{\mathbb{N}_0^r} \eta\} \quad (6)$$

$$\geq \min\{\mu(\eta) \mid \eta \in \Gamma, \lambda <_{\mathbb{N}_0^r} \eta\} \quad (7)$$

$$d(\tilde{C}(s)) \geq s$$

$$d(\tilde{C}_\varphi(s)) \geq s$$

*Proof* The description of  $E(\lambda)$  and  $C(\lambda)$  is an immediate consequence of the definition of  $\Delta(R, \rho, \varphi)$ . The estimates of the minimum distances of all codes but  $\tilde{C}(s)$  follow from Theorem 1. Finally,  $\tilde{C}(s) \subseteq \tilde{C}_\varphi(s)$  and therefore  $d(\tilde{C}(s)) \geq d(\tilde{C}_\varphi(s))$  holds.  $\square$

To illustrate the theorem we consider two examples.

**Table 1** Parameters from Example 7

$i$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\alpha(i)$	0	3	4	6	7	8	9	10	11	12	13	14	15	16
$\mu(\alpha(i))$	1	2	2	3	4	3	4	6	6	7	8	9	10	11
$\sigma(\alpha(i))$	27	24	23	21	20	19	18	17	16	15	14	13	12	11

$i$	15	16	17	18	19	20	21	22	23	24	25	26	27
$\alpha(i)$	17	18	19	20	21	22	23	24	25	26	28	29	32
$\mu(\alpha(i))$	12	13	14	15	16	17	18	19	20	21	23	24	27
$\sigma(\alpha(i))$	10	9	8	7	6	6	4	3	4	3	2	2	1

*Example 7* This is a continuation of Example 5. Consider the Hermitian polynomial  $X^4 - Y^3 - Y$  over  $\mathbb{F}_9$ . The set  $\Delta(R, \rho, \varphi) = \{\alpha(1), \dots, \alpha(27)\}$  was established in Example 5. We now list the corresponding values of  $\mu(\alpha(i))$  and  $\sigma(\alpha(i))$ . For  $i = 1, \dots, 19$   $\sigma(\alpha(i)) = n - \alpha(i)$  and therefore for  $i = 1, \dots, 19$  the code  $E(\alpha(i))$  has minimum distance at least  $n - \alpha(i) = 27 - \alpha(i)$  and dimension  $k = i$ . For larger values of  $i$  the picture is a little more complicated. For instance the minimum distance of  $E(22)$  is at least 6. The minimum distances of  $E(24)$ ,  $E(25)$  and  $E(26)$  are all estimated to 3. The corresponding dimensions are  $k = 22, 23, 24$ . The code  $\tilde{E}_\varphi(4)$  however has minimum distance at least 4 and dimension  $k = 22$ . For  $i = 8, \dots, 23$   $\min\{\mu(\alpha(s)) \mid i < s\} = i - 2$  and therefore for  $i = 8, \dots, 23$  the code  $C(\alpha(i))$  has minimum distance at least  $i - 2$  and dimension  $k = 27 - i$ . For smaller or larger values of  $i$  the picture is a little more complicated. For instance the minimum distance of  $C(4)$ ,  $C(6)$  and  $C(7)$  are estimated by 3. The corresponding dimensions are  $k = 24, 23, 22$ . The code  $\tilde{C}_\varphi(4)$  however has minimum distance at least 4 and dimension 22. The codes  $C(26)$ ,  $C(28)$  and  $C(29)$  have minimum distances at least 23, 24 respectively 27. In this example we considered the particular case  $q^2 = 9$ . The general case where  $q^2$  is arbitrary is treated in Geil (2003). It is shown that all estimates on the minimum distances are tight. The class of codes  $E(\lambda)$  equals the class of codes  $C(\lambda)$ . Similarly, the class of codes  $\tilde{E}_\varphi(s)$  equals the class of codes  $\tilde{C}_\varphi(s)$ .

*Example 8* This is a continuation of Examples 3 and 6. Consider the polynomial ring  $\mathbb{F}_5[X, Y]$ . For any of the described weight functions the values of  $\Delta(R, \rho, \varphi)$  are

$$\begin{aligned}
 &(0, 4) (1, 4) (2, 4) (3, 4) (4, 4) \\
 &(0, 3) (1, 3) (2, 3) (3, 3) (4, 3) \\
 &(0, 2) (1, 2) (2, 2) (3, 2) (4, 2) \\
 &(0, 1) (1, 1) (2, 1) (3, 1) (4, 1) \\
 &(0, 0) (1, 0) (2, 0) (3, 0) (4, 0)
 \end{aligned}$$

with corresponding  $\sigma$ -values respectively  $\mu$ -values

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & \\ 10 & 8 & 6 & 4 & 2 & \\ 15 & 12 & 9 & 6 & 3 & \text{respectively} \\ 20 & 16 & 12 & 8 & 4 & \\ 25 & 20 & 15 & 10 & 5 & \end{array} \quad \begin{array}{cccccc} 5 & 10 & 15 & 20 & 25 & \\ 4 & 8 & 12 & 16 & 20 & \\ 3 & 6 & 9 & 12 & 15 & \\ 2 & 4 & 6 & 8 & 10 & \\ 1 & 2 & 3 & 4 & 5 & \end{array}$$

To choose a particular weight function among the ones described in Example 3 we need to fix the term ordering  $<_{\mathbb{N}_0^2}$  on  $\mathbb{N}_0^2$ . Let the term ordering  $<_{\mathbb{N}_0^2}$  be the graded lexicographic ordering given by  $(a, b) <_{\mathbb{N}_0^2} (c, d)$  if either  $a + b < c + d$  holds or  $a + b = c + d$  holds with  $b < d$ . The generalized Reed–Muller code

$$\text{RM}_5(s, 2) = \{\varphi(F(X, Y)) \mid \deg(F) \leq s\}$$

is then seen to be equal to  $E((0, s))$ . By the first part of Theorem 2 we see that

$$E((0, s)) = \text{Span}_{\mathbb{F}_5}\{\varphi(X^i Y^j) \mid 0 \leq i < 5, 0 \leq j < 5, i + j \leq s\}.$$

We list the performance of a few of the codes. We have  $d(\tilde{E}_\varphi(5)) \geq 5$  and  $k(\tilde{E}_\varphi(5)) = 17$  whereas  $d(E((0, 4))) \geq 5$  and  $k(E((0, 4))) = 15$ . We have  $d(\tilde{E}_\varphi(4)) \geq 4$  and  $k(\tilde{E}_\varphi(4)) = 20$  whereas  $d(E((0, 5))) \geq 4$  and  $k(E((0, 5))) = 19$ . The study of  $\mu$  gives a similar picture of the codes  $C(\lambda)$  and  $\tilde{C}_\varphi(s)$ . In this example we considered the particular case  $\mathbb{F}_5[X, Y]$ . The general case  $\mathbb{F}_q[X_1, \dots, X_m]$  was treated in Geil and Høholdt (2001). It is shown that the estimates on the minimum distances are always tight. The class of codes  $E(\lambda)$  equals the class of codes  $C(\lambda)$ . Similarly, the class of codes  $\tilde{E}_\varphi(s)$  equals the class of codes  $\tilde{C}_\varphi(s)$ . The improved codes coming from the order domain  $\mathbb{F}_q[X_1, \dots, X_m]$  are known as hyperbolic codes or Massey–Costello–Justesen codes.

The above example illustrates the fact that using weights in  $\mathbb{N}_0^r$  with  $r > 1$  one can often construct rather long and still relatively good codes in a very simple way. We will see one more example of this in Sect. 5. In an unpublished work the author of the present paper has constructed asymptotically good concatenated codes that beats the performance of the Justesen codes for small rates by using as outer codes hyperbolic codes instead of Reed–Solomon codes.

We conclude the section by mentioning briefly some other interesting results from the literature. First we note that the bounds in Theorems 1 and 2 can be easily extended to deal not only with the minimum distance but with any generalized Hamming weight. Details can be found in Heijnen and Pellikaan (1998), Geil and Thommesen (2006) and Andersen and Geil (2008). Next we note that a modification of the order bound was made in Beelen (2007) to make the bound applicable to general geometric Goppa codes. Another result we would like to mention is that it is possible to modify Sudan’s list decoding without multiplicity so that it works for any evaluation code from order domain theory. Details can be found in Geil and

Matsumoto (2007). Finally, we note that there exists another improved code construction besides the ones described here, namely the improved generic evaluation codes. They were introduced in Bras-Amorós and O’Sullivan (2006) and allow for correction of so-called generic errors of high weight. The minimum distances of these codes however are not in general very high. In the next section we relate the bounds in Theorem 2 to the usual bounds from algebraic geometry on codes defined from curves.

## 4 One-Point Geometric Goppa Codes

In this section we treat weight functions with a numerical semigroup. We will always assume that  $\mathbb{N}_0 \setminus \Gamma$  is finite which is not really a restriction as any numerical semigroup will be isomorphic to a (unique) numerical semigroup such that the requirement holds. We observed in Example 2 that if  $\mathcal{P}$  is a rational place in an algebraic function field of one variable and  $v_{\mathcal{P}}$  is the corresponding valuation then  $R = \bigcup_{m=0}^{\infty} \mathcal{L}(m\mathcal{P})$  is an order domain with a weight function given by  $\rho(x) = -v_{\mathcal{P}}(x)$ . Clearly, any subring of such an order domain will again be an order domain and if the subring is non trivial then the corresponding semigroup  $\Gamma \subseteq \mathbb{N}_0$  will be non trivial. It is an obvious question if there are other examples of order domains with numerical weight functions than the ones coming from  $R = \bigcup_{m=0}^{\infty} \mathcal{L}(m\mathcal{P})$ . This question was settled in Matsumoto (1999, Theorem 1). The answer is no. Hence, if we restrict to algebraic structures over  $\mathbb{F}_q$  then what we are discussing in the present section are nothing but the algebraic structures giving us one-point geometric Goppa codes. Let  $\mathcal{Q}_1, \dots, \mathcal{Q}_n$  be pairwise different rational places not equal to  $\mathcal{P}$ . Then the map  $\varphi : \bigcup_{m=0}^{\infty} \mathcal{L}(m\mathcal{P}) \rightarrow \mathbb{F}_q^n$  is clearly a surjective morphism between  $\mathbb{F}_q$ -algebras and therefore one-point geometric Goppa codes  $C_{\mathcal{L}}(\mathcal{Q}_1 + \dots + \mathcal{Q}_n, s\mathcal{P})$  respectively  $C_{\Omega}(\mathcal{Q}_1 + \dots + \mathcal{Q}_n, s\mathcal{P})$  are codes of the form  $E(s)$  respectively  $C(s)$  coming from order domains with a weight function with a numerical semigroup. As we shall see we will be able to establish the usual bounds from algebraic geometry on their minimum distances using only a little effort.

We start by introducing some notation. Write  $\Gamma = \{\lambda_1 = 0, \lambda_2, \dots\}$  where  $\lambda_i < \lambda_{i+1}$  for  $i = 1, 2, \dots$ . We define  $g(i) = \#\{\lambda \in \mathbb{N}_0 \setminus \Gamma \mid \lambda < \lambda_i\}$  and  $g = \#\mathbb{N}_0 \setminus \Gamma$ . According to the discussion above  $\Gamma$  is the Weirstrass semigroup of a rational place and therefore by the Weirstrass Gap Theorem  $g$  equals the genus of the function field under consideration. The following results from Høholdt et al. (1998, Lemma 5.15 and Theorem 5.24) are easily proven.

**Lemma 2** *For any  $i \in \mathbb{N}_0$  we have  $\Gamma \setminus (\lambda_i + \Gamma) = \lambda_i$  and  $\mu(\lambda_i) = i - g(i) + \#D(i)$  where  $D(i) = \{(x, y) \mid x, y \in \mathbb{N}_0 \setminus \Gamma \text{ and } x + y = \lambda_i\}$ .*

Applied to the special case of one-point geometric Goppa codes the Goppa bounds from algebraic geometry states.

**Theorem 3** Assume  $\Gamma$  is numerical with  $\mathbb{N}_0 \setminus \Gamma$  finite. We have

$$d(E(s)) \geq n - s \quad (8)$$

$$d(C(\lambda_t)) \geq t + 1 - g \quad (9)$$

The usual proof of the Goppa bounds requires the use of the Riemann-Roch theorem. However, in the case of one-point geometric Goppa codes order domain theory provides an alternative proof by combining Theorem 2 and Lemma 2.

**Theorem 4** Assume  $\Gamma$  is numerical with  $\mathbb{N}_0 \setminus \Gamma$  finite. The bound in (5) is at least as good as the bound (8) and sometimes better. The bound in (7) is at least as good as the bound (9) and sometimes better.

*Proof* To prove the first claim we need only consider numbers  $i \in \Delta(R, \rho, \varphi)$  with  $i \leq s$ . We have  $\sigma(i) = \#(\Delta(R, \rho, \varphi) \cap (i + \Gamma))$ . By Lemma 2 the number of elements in  $\Delta(R, \rho, \varphi)$  that are not in  $i + \Gamma$  is at most  $i$  and therefore  $\sigma(i) \geq n - i$  holds. Equality holds only when  $\Gamma \setminus (i + \Gamma) \subseteq \Delta(R, \rho, \varphi)$ . We conclude  $\min\{\sigma(i) \mid i \in \Delta(R, \rho, \varphi), i \leq s\} \geq n - s$ . Concerning the last claim we have

$$\min\{\mu(\eta) \mid \eta \in \Gamma \text{ and } \lambda_t < \eta\} = \min\{i - g(i) + \#D(i) \mid t < i\} \geq t + 1 - g$$

with equality if and only if  $\lambda_{t+1} = \lambda_t + 1$ ,  $g(t+1) = g$  and  $\#D(t+1) = 0$  holds.  $\square$

## 5 Gröbner Basis Theoretical Tools for the Construction of Order Domains

In this section we will see how to construct order domains by use of only simple Gröbner basis theoretical tools. The method to be described can be viewed as a generalization of the Hermitian order domain construction from Example 1. We start by recalling some basic facts from Gröbner basis theory. In the following let  $\mathbb{F}$  be any field.

**Definition 8** Denote by  $\mathcal{M}(X_1, \dots, X_m)$  the set of monomials in  $X_1, \dots, X_m$ . Given a term ordering  $<_{\mathcal{M}}$  on  $\mathcal{M}(X_1, \dots, X_m)$  and an ideal  $I \subseteq \mathbb{F}[X_1, \dots, X_m]$  the footprint (or the Gröbner éscalier) of  $I$  is the set

$$\begin{aligned} \Delta_{<_{\mathcal{M}}}(I) = \{M \in \mathcal{M}(X_1, \dots, X_m) \mid M \text{ is not} \\ \text{a leading monomial of any polynomial in } I\}. \end{aligned}$$

**Theorem 5** Let  $I \subseteq \mathbb{F}[X_1, \dots, X_m]$  be an ideal. Then  $\{M + I \mid M \in \Delta_{<_{\mathcal{M}}}(I)\}$  is a basis for  $\mathbb{F}[X_1, \dots, X_m]/I$  as a vector space over  $\mathbb{F}$ .

*Example 9* This is a continuation of Example 1 where we considered the Hermitian order domain  $R = \mathbb{F}_{q^2}[X, Y]/I$ , with  $I = \langle X^{q+1} - Y^q - Y \rangle$ . Let a weighted degree lexicographic ordering  $<_w$  on  $\mathcal{M}(X, Y)$  be given as follows. We have  $X^\alpha Y^\beta <_w X^\gamma Y^\delta$  if either  $\alpha q + \beta(q+1) < \gamma q + \delta(q+1)$  holds or  $\alpha q + \beta(q+1) = \gamma q + \delta(q+1)$  but  $\beta < \delta$  holds. Clearly,  $\Delta_{<_w}(I) = \{X^i Y^j \mid 0 \leq i, 0 \leq j < q\}$  and therefore by Theorem 5  $\mathcal{B} = \{X^i Y^j + I \mid 0 \leq i, 0 \leq j < q\}$  is a basis for  $R$ .

For the construction of order domains we will need generalized weighted degree orderings. These are defined as follows

**Definition 9** Given weights  $w(X_1), \dots, w(X_m) \in \mathbb{N}_0^r \setminus \{\mathbf{0}\}$  let  $\mathbb{N}_0^r$  be ordered by some fixed term ordering  $<_{\mathbb{N}_0^r}$ . Let  $<_{\mathcal{M}}$  be a fixed term ordering on  $\mathcal{M}(X_1, \dots, X_m)$ . The weights extend to a monomial function  $w : \mathcal{M}(X_1, \dots, X_m) \rightarrow \mathbb{N}_0^r$  by  $w(X_1^{\alpha_1} \cdots X_m^{\alpha_m}) = \sum_{i=1}^m \alpha_i w(X_i)$ . For a monomial  $M$  we call  $w(M)$  the weight of  $M$ . Now the generalized weighted degree ordering  $<_w$  induced by  $w$ ,  $<_{\mathbb{N}_0^r}$  and  $<_{\mathcal{M}}$  is the term ordering defined as follows. Given  $M_1, M_2 \in \mathcal{M}(X_1, \dots, X_m)$  then  $M_1 <_w M_2$  if and only if one of the following two conditions holds

- (GWD.1)  $w(M_1) <_{\mathbb{N}_0^r} w(M_2)$
- (GWD.2)  $w(M_1) = w(M_2)$  and  $M_1 <_{\mathcal{M}} M_2$ .

We observe, that if the weights are numerical then we do not need to define the ordering  $<_{\mathbb{N}_0^{r-1}}$  as there exists only one term ordering on  $\mathbb{N}_0$ . In this case the generalized weighted degree ordering simplifies to the usual weighted degree ordering. We can now describe the main result of this section.

**Theorem 6** Let  $I$  be an ideal in  $\mathbb{F}[X_1, \dots, X_m]$  and assume  $\mathcal{G}$  is a Gröbner basis for  $I$  with respect to a generalized weighted degree ordering  $<_w$ . Suppose that the elements of the corresponding footprint  $\Delta_{<_w}(I)$  have mutually distinct weights and that every element of  $\mathcal{G}$  has exactly two monomials of highest weight in its support. Write  $\Gamma = \{w(M) \mid M \in \Delta_{<_w}(I)\} \subseteq \mathbb{N}_0^r$ . For  $f \in \mathbb{F}[X_1, \dots, X_m]/I$  denote by  $F$  the (unique) remainder of any polynomial in  $f$  after division with  $\mathcal{G}$ . Then  $R = \mathbb{F}[X_1, \dots, X_m]/I$  is an order domain with a weight function  $\rho : R \rightarrow \Gamma \cup \{-\infty\}$  defined by  $\rho(0) = -\infty$  and  $\rho(f) = \max_{<_{\mathbb{N}_0^r}} \{w(M) \mid M \in \text{Supp}(F)\}$  for  $f \neq 0$ .

*Proof* Theorem 5 tells us that  $\mathcal{B} = \{M + I \mid M \in \Delta_{<_w}(I)\}$  is a basis for  $R$  as a vector space over  $\mathbb{F}$ . For  $f \in \mathcal{B}$  let  $F \in \Delta_{<_w}(I)$  be the unique monomial such that  $f = F + I$ . The map  $\rho : \mathcal{B} \rightarrow \Gamma$  given by  $\rho(f) = w(F)$  is well-defined and by assumption it is bijective. If we can show that  $\mathcal{B} = \{f_\lambda \mid \lambda \in \Gamma\}$  is a well-behaving basis then from Definition 2 it follows that a weight function is given just as described in the theorem above. For  $\gamma \in \Gamma$  we denote by  $f_\gamma$  the element in  $\mathcal{B}$  with  $\rho(f_\gamma) = \gamma$  and write  $f_\gamma = F_\gamma + I$  where  $F_\gamma \in \Delta_{<_w}(I)$ . It follows by the definition of  $\rho$  that  $w(F_\gamma) = \gamma$  holds. Recall from Definition 1, that  $R_{-\infty} = \{0\}$  and that for  $\lambda \in \Gamma$  we define  $R_\lambda = \text{Span}_{\mathbb{F}}\{f_\gamma \mid \gamma \leq_{\mathbb{N}_0^r} \lambda\}$ . We must show that if  $\alpha, \beta \in \Gamma$  then  $f_\alpha f_\beta \in R_{\alpha+\beta}$  but

$f_\alpha f_\beta \notin R_\delta$  for any  $\delta$  with  $\delta <_{\mathbb{N}_0} \alpha + \beta$ . By the definition of  $\rho$  this corresponds to showing that if  $f_\alpha f_\beta$  is written as

$$\sum_{\eta \in \Gamma, k_\eta \in \mathbb{F}} k_\eta F_\eta + I$$

then

$$\max_{<_{\mathbb{N}_0}} \left\{ w(M) \mid M \text{ is in the support of } \sum_{\eta \in \Gamma, k_\eta \in \mathbb{F}} k_\eta F_\eta \right\} = \alpha + \beta$$

holds. Multiplying  $f_\alpha = F_\alpha + I$  with  $f_\beta = F_\beta + I$  we get  $F_\alpha F_\beta + I$ . Clearly,  $F_\alpha F_\beta$  is a monomial but it need not be an element in  $\Delta_{<_w}(I)$ . To find

$$\sum_{\eta \in \Gamma, k_\eta \in \mathbb{F}} k_\eta F_\eta$$

we reduce  $F_\alpha F_\beta$  modulo  $\mathcal{G}$ . At every stage of this reduction by induction the derived polynomial will have exactly one monomial of highest weight in its support and the weight of this monomial equals  $w(F_\alpha F_\beta) = \alpha + \beta$ .  $\square$

*Example 10* This is a continuation of Examples 1 and 9 where we considered the Hermitian order domain. Clearly,  $\mathcal{G} = \{X^{q+1} - Y^q - Y\}$  is a Gröbner basis for  $I$  with respect to  $<_w$ . We have  $w(X^{q+1}) = w(Y^q) = q(q+1) > w(Y) = q$  and therefore all polynomials in  $\mathcal{G}$  contains exactly two monomials of highest weight. It is easily verified that no two different monomials in  $\Delta_{<_w}(I)$  are of the same weight and therefore all the conditions in Theorem 6 are satisfied.

*Example 11* This is a continuation of Example 3 where we considered a family of weight functions on the order domain  $R = \mathbb{F}[X_1, \dots, X_m]$ . Using the convention that  $\mathcal{G} = \emptyset$  is a Gröbner basis for the ideal  $\langle 0 \rangle$  the description in Example 3 can be viewed as an instance of Theorem 6.

Theorem 6 actually captures all order domains relevant in coding theory including the spaces  $R = \bigcup_{m=0}^{\infty} \mathcal{L}(m\mathcal{P})$  used in the construction of one-point geometric Goppa codes (Example 2 and Sect. 4). This not too obvious result is the content of Geil and Pellikaan (2002). To explain precisely what Geil and Pellikaan (2002, Theorem 10.4) is saying we consider the general definition of an order function in Remark 2. We observe that although the well-order  $(\Gamma, <_\Gamma)$  is not born with a binary operation the order function induces one. More precisely, one can define an operation  $\oplus$  on  $\Gamma$  by the rule  $\rho(f) \oplus \rho(g) = \rho(fg)$ . Denoting by 0 the minimal element of  $\Gamma$ ,  $(\Gamma, \oplus, 0)$  becomes a semigroup. Now Geil and Pellikaan (2002, Theorem 10.4) deals with the case where an order domain and an order function is given for which the semigroup  $(\Gamma, \oplus, 0)$  is finitely generated. Under this condition the following three things hold. Firstly,  $(\Gamma, \oplus, 0)$  is isomorphic to a sub semigroup

of  $\mathbb{N}_0^r$  for some  $r$ . Secondly, under the isomorphism  $<_{\Gamma}$  is the restriction of a term ordering on  $\mathbb{N}_0^r$  to  $\Gamma$ . Finally and most importantly, up to isomorphism the order domain and the order function can be described as in Theorem 6. In particular if an order function  $\rho$  has a finitely generated semigroup  $(\Gamma, \oplus, 0)$  then  $\rho$  is isomorphic to a weight function. Furthermore, Geil and Pellikaan (2002, Theorem 11.9) states that if the transcendence degree of  $R$  is  $r$  and  $(\Gamma, \oplus, 0)$  is finitely generated then it is possible to embed  $(\Gamma, \oplus, 0)$  into  $(\mathbb{N}_0^r, +, 0)$  but impossible to embed it into  $(\mathbb{N}_0^{r-1}, +, 0)$ . We next observe that every numerical semigroup is finitely generated and therefore in theory the algebraic structure  $R = \bigcup_{m=0}^{\infty} \mathcal{L}(m\mathcal{P})$  used in the construction of one-point geometric Goppa codes can be described as in Theorem 6. One main advantage of Theorem 6 is that it allows us to construct in a very easy way order domains of higher transcendence degree. We now give such an example.

*Example 12* Let

$$\begin{aligned} H_1(X, Y, Z, U) &= X^q + YZ^q - Y^q Z - X, \\ H_2(X, Y, Z, U) &= U^q - Z^{q+1} + aX^q - aY^q Z + bY^{q+1} + U \end{aligned}$$

where  $a, b \in \mathbb{F}_q$ . Consider  $I = \langle H_1(X, Y, Z, U), H_2(X, Y, Z, U) \rangle \subseteq \mathbb{F}_{q^2}[X, Y, Z, U]$  and define the generalized weighted degree ordering  $<_w$  on  $\mathcal{M}(X, Y, Z, U)$  as follows. Consider weights  $w(X) = (q, 1), w(Y) = (0, q), w(Z) = (q, 0), w(U) = (q+1, 0) \in \mathbb{N}_0^2$  and let  $<_{\mathbb{N}_0^2}$  be any fixed term ordering on  $\mathbb{N}_0^2$  that satisfies  $(q^2, q), (q, q^2), (0, q^2 + q) <_{\mathbb{N}_0^2} (q^2 + q, 0)$  and  $(q, q^2) <_{\mathbb{N}_0^2} (q^2, q)$ . Finally let  $<_{\mathcal{M}}$  be any fixed term ordering on  $\mathcal{M}(X, Y, Z, U)$  that satisfies  $YZ^q <_{\mathcal{M}} X^q$  and  $Z^{q+1} <_{\mathcal{M}} U^q$ . The leading monomial of  $H_1$  is  $X^q$  and the leading monomial of  $H_2$  is  $U^q$ . Hence, the two leading monomials are relatively prime. By a standard result in Gröbner basis theory this implies that  $\{H_1(X, Y, Z, U), H_2(X, Y, Z, U)\}$  constitutes a Gröbner basis. It is easily shown that the remaining conditions in Theorem 6 are satisfied. From Theorem 6 we get a weight function

$$\rho : R = \mathbb{F}_{q^2}[X, Y, Z, U]/I \rightarrow \{(q, 1), (0, q), (q, 0), (q+1, 0)\} \cup \{-\infty\}.$$

The particular choice of terms not of highest weight in  $H_1$  and  $H_2$  will be important in a later example where we derive codes from the above order domain.

Consider any finitely generated semigroup  $\Gamma = \langle \lambda_1, \dots, \lambda_m \rangle \subseteq \mathbb{N}_0^r$  and a term ordering  $<_{\mathbb{N}_0^r}$ . Let an ordering  $<_w$  on  $\mathcal{M}(X_1, \dots, X_m)$  be defined by  $w(X_1) = \lambda_1, \dots, w(X_m) = \lambda_m$  and some term ordering  $<_{\mathcal{M}}$ . The ideal

$$\begin{aligned} I_{\Gamma} &= \langle M - N \mid M, N \in \mathcal{M}(X_1, \dots, X_m), w(M) = w(N) \rangle \\ &\subseteq \mathbb{F}[X_1, \dots, X_m] \end{aligned} \tag{10}$$

is called a toric ideal.  $I_{\Gamma}$  has a Gröbner basis with respect to  $<_w$  that consists of a collection of binomials of the form from (10). In fact, the Gröbner basis can be

found by use of elimination theory (see Geil and Pellikaan 2002, Proposition. 10.6). By (10) no two different monomials of the same weight can be simultaneously members of  $\Delta_{<w}(I_\Gamma)$  and therefore the conditions in Theorem 6 are satisfied. That is, we have a weight function

$$\rho : R_\Gamma = \mathbb{F}[X_1, \dots, X_m]/I_\Gamma \rightarrow \Gamma \cup \{-\infty\}.$$

This sort of a trivial order domain plays a special role in order domain theory. Namely, it was shown in Little (2007) that the conditions in Theorem 6 regarding the defining polynomials of a  $\mathbb{F}$ -algebra  $R = \mathbb{F}[X_1, \dots, X_m]/I$  and a semigroup  $\Gamma \subseteq \mathbb{N}_0^n$  are equivalent to saying that  $R$  has a flat deformation to  $R_\Gamma$ . This result has proved very useful. As an example it is used in Little (2007, Sect. 6) in combination with some results on deformation of Grassmannians to derive weight functions on all Grassmannians.

## 6 Gröbner Basis Theoretical Tools for the Code Construction

In this section we shall see that not only is Gröbner basis theory an important tool for the construction of order domains—it is also an important tool for the construction of the corresponding codes. Recall, that for the code construction we need an order domain  $R$  over  $\mathbb{F}_q$  and a surjective  $\mathbb{F}_q$ -linear map  $\varphi : R \rightarrow \mathbb{F}_q^n$  satisfying  $\varphi(fg) = \varphi(f) * \varphi(g)$ . Recall, that such a map is called a morphism between  $\mathbb{F}_q$ -algebras. Given an order domain  $R = \mathbb{F}_q[X_1, \dots, X_m]/I$  as in Theorem 6 the most obvious choice of  $\varphi$  would be  $\varphi(F + I) = (F(P_1), \dots, F(P_n))$  where  $\{P_1, \dots, P_n\} \subseteq \mathcal{V}_{\mathbb{F}_q}(I)$ . Here,  $\mathcal{V}_{\mathbb{F}_q}(I)$  denotes the variety of  $I$ . Theorem 7 tells us that there are no maps beside this that have the desired properties.

**Theorem 7** *Let  $\varphi : \mathbb{F}_q[X_1, \dots, X_m]/I \rightarrow \mathbb{F}_q^n$  be a surjective  $\mathbb{F}_q$ -linear map satisfying  $\varphi(fg) = \varphi(f) * \varphi(g)$  for all  $f, g \in \mathbb{F}_q[X_1, \dots, X_m]/I$ . Then there exists a set  $\{P_1, \dots, P_n\} \subseteq \mathcal{V}_{\mathbb{F}_q}(I)$ ,  $P_i \neq P_j$  for  $i \neq j$  such that  $\varphi(F(X_1, \dots, X_m) + I) = (F(P_1), \dots, F(P_n))$  holds for all  $F(X_1, \dots, X_m) \in \mathbb{F}_q[X_1, \dots, X_m]$ .*

*Proof* We will use the notation  $\varphi(f) = (\varphi_1(f), \dots, \varphi_n(f))$ . The assumption that  $\varphi$  is surjective implies that  $\varphi_i : \mathbb{F}_q[X_1, \dots, X_m]/I \rightarrow \mathbb{F}_q$ ,  $i = 1, \dots, n$  are pairwise different surjective maps. The remaining assumptions imply that  $\varphi_i : \mathbb{F}_q[X_1, \dots, X_m]/I \rightarrow \mathbb{F}_q$  is a ring homomorphism with  $\varphi_i(c + I) = c$  for all  $c \in \mathbb{F}_q$ . Writing  $x_1 = X_1 + I, \dots, x_m = X_m + I$  and identifying  $c + I$  with  $c$  for all  $c \in \mathbb{F}_q$  we get  $F(X_1, \dots, X_m) + I = F(x_1, \dots, x_m)$ . This is nothing but the usual way of doing arithmetic on residue classes. Now let  $P_i^{(1)} = \varphi_i(x_1), \dots, P_i^{(m)} = \varphi_i(x_m) \in \mathbb{F}_q$ . The fact that  $\varphi_i$  is a ring homomorphism with  $\varphi_i(c + I) = c$  for all  $c \in \mathbb{F}_q$  now implies that  $\varphi_i(F(x_1, \dots, x_m)) = F(P_i^{(1)}, \dots, P_i^{(m)})$  holds. That is,  $\varphi_i(F(X_1, \dots, X_m) + I) = F(P_i^{(1)}, \dots, P_i^{(m)})$ . For every  $F(X_1, \dots, X_m) \in I$  we have  $\varphi_i(F(x_1, \dots, x_m)) = \varphi_i(0 + I) = 0$  and therefore  $P_i = (P_i^{(1)}, \dots, P_i^{(n)})$  is a zero of  $F(X_1, \dots, X_m)$ . In other words  $P_i \in \mathcal{V}_{\mathbb{F}_q}(I)$ .  $\square$

In conclusion we see that a very large class of algebraic geometry codes including one-point geometric Goppa codes can be described by use of only simple Gröbner basis theoretical tools. Unfortunately, given a general order domain  $R$  then it is not at all obvious how to derive the description in Theorem 6. However, it is still possible to apply the simple Gröbner basis theoretical tools when dealing with the codes at a theoretical level. As an example we note that one can reprove the result in Sect. 4 regarding the minimum distance of the codes  $E(\lambda)$  and  $\tilde{E}_\varphi(s)$  in a pure Gröbner basis theoretical setting.

The remainder of the present section is about the case where a description as in Theorem 6 is known. As we are normally interested in large codes we concentrate mostly on the case where  $\varphi$  is defined by evaluating in all the points of the variety  $\mathcal{V}_{\mathbb{F}_q}(I)$ . Recall, that for the actual code construction we would like to know for which  $\lambda \in \Gamma$  we have  $\varphi(R_\lambda) \neq \varphi(R_\gamma)$  for all  $\gamma <_{\mathbb{N}_0^r} \lambda$ . The set of such  $\lambda$ s was denoted  $\Delta(R, \rho, \varphi)$  in Sect. 3. The following not too surprising theorem explains the choice of notation.

**Theorem 8** *Consider an order domain  $R$  and a weight function  $\rho : R \rightarrow \Gamma \cup \{-\infty\}$  described as in Theorem 6. Let  $\varphi$  be the morphism  $\varphi : R \rightarrow \mathbb{F}_q^n$  given by  $\varphi(F + I) = (F(P_1), \dots, F(P_n))$  where  $\mathcal{V}_{\mathbb{F}_q}(I) = \{P_1, \dots, P_n\}$ ,  $P_i \neq P_j$  for  $i \neq j$ . We have  $\Delta(R, \rho, \varphi) = \{w(M) \mid M \in \Delta_{<_w}(I_q)\}$  where  $I_q = I + \langle X_1^q - X_1, \dots, X_m^q - X_m \rangle$ .*

*Remark 5* It is possible to generalize Theorem 8 to deal with the situation where  $\{P_1, \dots, P_n\}$  ( $P_i \neq P_j$  for  $i \neq j$ ) is not necessarily the entire variety  $\mathcal{V}_{\mathbb{F}_q}(I)$  but is any subset. From the fact that every finite set of points constitutes a variety we conclude that there exist polynomials  $G_1(X_1, \dots, X_m), \dots, G_s(X_1, \dots, X_m)$  such that  $\{P_1, \dots, P_n\} = \mathcal{V}_{\mathbb{F}_q}(I + \langle G_1, \dots, G_s \rangle)$ . But then we can apply Andersen (2007, Proposition 20) which states that if  $\{P_1, \dots, P_n\} = \mathcal{V}_{\mathbb{F}_q}(I + \langle G_1, \dots, G_s \rangle)$  then the result in Theorem 8 holds again if we replace  $I_q = I + \langle X_1^q - X_1, \dots, X_m^q - X_m \rangle$  with  $I + \langle G_1, \dots, G_s \rangle + \langle X_1^q - X_1, \dots, X_m^q - X_m \rangle$ .

*Remark 6* The concept of affine variety codes was coined in Fitzgerald and Lax (1998). The construction is based on an ideal  $I \subseteq \mathbb{F}_q[X_1, \dots, X_m]$  from which we define  $I_q$  just as in Theorem 8. We write  $\{P_1, \dots, P_n\} = \mathcal{V}_{\mathbb{F}_q}(I_q)$ ,  $R = \mathbb{F}_q[X_1, \dots, X_m]/I_q$  and let  $L$  be any subspace of the vector space  $R$ . Defining  $\varphi : R \rightarrow \mathbb{F}_q^n$  by  $\varphi(F(X_1, \dots, X_m) + I) = (F(P_1), \dots, F(P_n))$  the code  $C(I, L) = \{\varphi(f) \mid f \in L\}$  and its dual are called affine variety codes. We observe that Theorems 7, 8 and Remark 5 provide us with a way of interpreting order domain codes as affine variety codes.

Theorem 8 immediately applies to the Hermitian order domain and the polynomial ring  $\mathbb{F}_q[X_1, \dots, X_m]$ . However, we already derived the corresponding set  $\Delta(R, \rho, \varphi)$  in Examples 5 and 6 so we will not treat them again. Instead we apply Theorem 8 to the order domain in Example 12.

*Example 13* In Example 12 we considered  $R = \mathbb{F}_{q^2}[X, Y, Z, U]/I$  where  $I = \langle H_1(X, Y, Z, U), H_2(X, Y, Z, U) \rangle$  and  $H_1(X, Y, Z, U) = X^q + YZ^q - Y^q Z - X$ ,

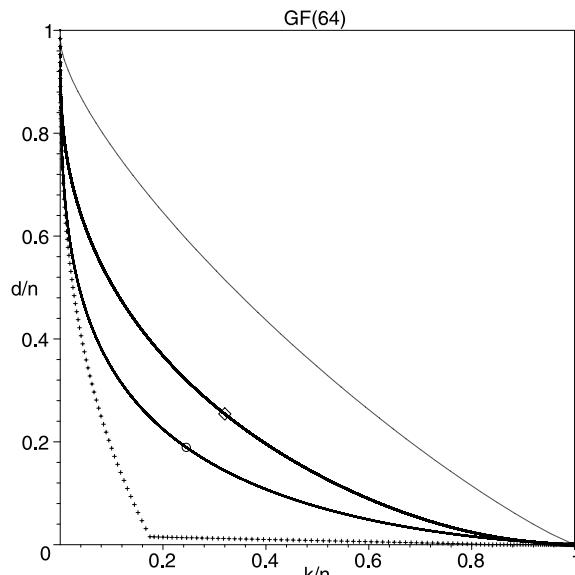
$H_2(X, Y, Z, U) = U^q - Z^{q+1} + aX^q - aY^qZ + bY^{q+1} + U$  with  $a, b \in \mathbb{F}_q$ . We defined weights  $w(X) = (q, 1)$ ,  $w(Y) = (0, q)$ ,  $w(Z) = (q, 0)$ ,  $w(U) = (q+1, 0) \in \mathbb{N}_0^2$  and chose as term ordering  $<_{\mathbb{N}_0^2}$  any term ordering satisfying  $(q^2, q), (q, q^2), (0, q^2+q) <_{\mathbb{N}_0^2} (q^2+q, 0)$  and  $(q, q^2) <_{\mathbb{N}_0^2} (q^2, q)$ . As ordering  $<_{\mathcal{M}}$  we chose any term ordering on  $\mathcal{M}(X, Y, Z, U)$  that satisfies  $YZ^q <_{\mathcal{M}} X^q$  and  $Z^{q+1} <_{\mathcal{M}} U^q$ . Defining  $<_w$  accordingly we showed that  $R$  is an order domain satisfying the conditions in Theorem 6. By applying Buchberger's first criterion we now see that

$$\mathcal{G}' = \{H_1(X, Y, Z, U), H_2(X, Y, Z, U), X^{q^2} - X, Y^{q^2} - Y, Z^{q^2} - Z, U^{q^2} - U\}$$

constitutes a Gröbner basis for  $I_{q^2}$ . Hence, we get

$$\Delta_{<_w}(I_q) = \{X^\alpha Y^\beta Z^\gamma U^\delta \mid \alpha, \delta < q \text{ and } \beta, \gamma < q^2\}.$$

The footprint is of size  $q^6$  and we therefore get codes of length  $n = q^6$ . The footprint  $\Delta_{<_w}(I_q)$  has the form of a box. From this observation it is not difficult to show that the dimension of  $\tilde{C}_\varphi(s)$  equals the dimension of  $\tilde{E}_\varphi(s)$  for all  $s = 1, 2, \dots, q^6$ . In Fig. 1 we plot the estimated performances of the codes  $\tilde{E}_\varphi(\delta)$  and  $\tilde{C}_\varphi(\delta)$  from the present example in the case  $\mathbb{F}_{q^2} = \mathbb{F}_{64}$ . These codes are of length  $n = 262144$  and are marked with a  $\diamond$ . The hyperbolic codes and the generalized Reed–Muller codes from  $\mathbb{F}_{64}[X_1, X_2, X_3]$  are of the same length. For comparison we also plot their performances. The performances of the hyperbolic codes are given by the graph marked with a  $\circ$  and the performances of the generalized Reed–Muller codes are marked with  $+$ 's. The last graph is the asymptotic Gilbert–Varshamov bound.



**Fig. 1** Code performance

## 7 The Connection to Valuation Theory

The theory of order domains has grown too large in its almost ten years lifetime for us to be able to cover all interesting aspects in the present paper. One of the aspects that we have not treated is the connection to valuation theory. We now give a brief discussion of the subject and refer the reader to the literature for more details. In Example 2 and Sect. 4 we demonstrated the close connection between weight functions with  $\Gamma \subseteq \mathbb{N}_0 \cup \{-\infty\}$  and valuations on curves. It should come as no surprise that every weight function corresponds to a valuation on an extension of the order domain. In Remark 2 we defined a more general class of functions called order functions which maps to a well-order  $(\Gamma, <_\Gamma)$ . As mentioned in Sect. 5 we can make  $\Gamma$  into a semigroup by defining the binary operation  $\oplus$  on  $\Gamma$  by  $\rho(f) \oplus \rho(g) = \rho(fg)$ . It was shown in O’Sullivan (2001, Theorem 2.1) and Geil and Pellikaan (2002, Proposition 6.1) that the above observation regarding a connection to valuation theory applies to order functions in general, in that an order function  $\rho : R \rightarrow \Gamma \cup \{-\infty\}$  defines a valuation  $\tilde{\rho} : \text{QF}(R) \rightarrow D(\Gamma) \cup \{\infty\}$  by  $\tilde{\rho}(0) = \infty$  and  $\tilde{\rho}(f/g) = \rho(g) - \rho(f)$ . Here,  $\text{QF}(R)$  denotes the field of fractions of  $R$  and  $D(\Gamma)$  is the totally ordered semigroup of differences of  $\Gamma$ . We have seen in Sect. 4 that every valuation related to a rational place of a function field in one variable defines an order function. For function fields in more variables the picture is more complicated as for these there are classes of valuations that do not define order functions. A thoroughly treatment of the problem in the case of a function field in two variables can be found in O’Sullivan (2001). In Little (2007) order functions are constructed on the basis of projective varieties that have a flag of subvarieties satisfying certain mild conditions. Using this method order functions on all Hermitian hypersurfaces are described.

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

The author wishes to express a special thank to Professor Ryutaroh Matsumoto who drew the author’s attention to Theorem 7. Also the author would like to thank the anonymous referees as well as Professor Massimiliano Sala and Professor Shojiro Sakata for their helpful comments and suggestions.

## References

- H. E. Andersen, *On puncturing of codes from norm-trace curves*, Finite Fields Appl. **13** (2007), no. 1, 136–157.
- H. E. Andersen and O. Geil, *Evaluation codes from order domain theory*, Finite Fields Appl. **14** (2008), 92–123.
- D. Augot, E. Betti and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- P. Beelen, *The order bound for general algebraic geometric codes*, Finite Fields Appl. **13** (2007), no. 3, 655–680.
- M. Bras-Amorós and M. E. O’Sullivan, *The correction capability of the Berlekamp–Massey–Sakata algorithm with majority voting*, AAECC **17** (2006), no. 5, 315–335.

- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- J. Fitzgerald and R. F. Lax, *Decoding affine variety codes using Gröbner bases*, Des. Codes Cryptogr. **13** (1998), no. 2, 147–158.
- O. Geil, *On codes from norm-trace curves*, Finite Fields Appl. **9** (2003), 351–371.
- O. Geil and T. Høholdt, *On hyperbolic codes*, LNCS, vol. **2227**, Springer, Berlin, 2001, pp. 159–171.
- O. Geil and R. Matsumoto, *Generalized Sudan's list decoding for order domain codes*, Proc. of AAECC2007, 2007, pp. 50–59.
- O. Geil and R. Pellikaan, *On the structure of order domains*, Finite Fields Appl. **8** (2002), 369–396.
- O. Geil and C. Thommesen, *On the Feng–Rao bound for generalized Hamming weights*, LNCS, vol. **3857**, Springer, Berlin, 2006, pp. 295–306.
- P. Heijnen and R. Pellikaan, *Generalized Hamming weights of  $q$ -ary Reed–Muller codes*, IEEE Trans. on Inf. Th. **44** (1998), no. 1, 181–196.
- T. Høholdt, J. van Lint and R. Pellikaan, *Algebraic geometry of codes*, Handbook of coding theory (V. S. Pless and W.C. Huffman, eds.), Elsevier, Amsterdam, 1998, pp. 871–961.
- D. A. Leonard, *A tutorial on AG code construction from a Gröbner basis perspective*, this volume, 2009, pp. 93–106.
- J. B. Little, *The ubiquity of order domains for the construction of error control codes*, Adv. Math. Commun. **1** (2007), no. 1, 151–171.
- J. B. Little, *Automorphisms and encoding of AG and order domain codes*, this volume, 2009, pp. 107–120.
- R. Matsumoto, *Miura's generalization of one-point AG codes is equivalent to Høholdt, van Lint and Pellikaan's generalization*, IEICE Trans. Fund. **E82-A** (1999), no. 10, 2007–2010.
- M. E. O'Sullivan, *New codes for the Berlekamp–Massey–Sakata algorithm*, Finite Fields Appl. **7** (2001), no. 2, 293–317.
- S. Sakata, *The BMS algorithm*, this volume, 2009a, pp. 143–163.
- S. Sakata, *The BMS algorithm and decoding of AG codes*, this volume, 2009b, pp. 165–185.

# The BMS Algorithm

Shojiro Sakata

**Abstract** We present a sketch of the  $n$ -dimensional ( $n$ -D) Berlekamp–Massey algorithm (alias Berlekamp–Massey–Sakata or BMS algorithm) w.r.t.  $n$ -D arrays. That is: (1) How is it related to Gröbner basis? (2) What problem can it solve? (3) How does it work? (4) Its variations. First we discuss another problem closely related to our main problem, and introduce some concepts about  $n$ -D linear recurrences and modules of  $n$ -D arrays as their general solutions. These two problems are just the inverse (or rather dual) to each other, which can be solved by the Buchberger algorithm (Buchberger in Ein Algorithmus zum Auffinden der Basislemente des Restklassenringes nach einem nulldimensionalen Polynomideal, Ph.D. thesis, Innsbruck, 1965; J. Symb. Comput. 41(3–4):475–511, 2006; Multidimensional systems theory, Reidel, Dordrecht, pp. 184–232, 1985; Mora in Gröbner technology, this volume, pp. 11–25, 2009b), and the BMS algorithm, respectively. Furthermore, we discuss some properties of BMS algorithm and its outputs, including its computational complexity, as well as several variations of the BMS algorithm.

## 1 Introduction

In this paper, we present a sketch of the multidimensional Berlekamp–Massey algorithm (alias Berlekamp–Massey–Sakata algorithm or BMS algorithm) from (Sakata 1988, 1990). It is a generalization of the Berlekamp–Massey algorithm (Berlekamp 1968; Massey 1969) from one-dimensional (1-D) arrays to  $n$ -dimensional ( $n$ -D) arrays for  $n \geq 1$ . We discuss:

- (1) How is it related to the Gröbner basis theory?
- (2) What problem can it solve?
- (3) How does it work?
- (4) its several variations.

In another paper (Sakata 2009) of this issue we present its applications to decoding of algebraic error-correcting codes. In most part of this paper we restrict ourselves to treating finite fields although the contents remain valid in any field  $\mathbb{F}$  provided that we have exact computations over  $\mathbb{F}$ .

Before we introduce our main theme, i.e. our main problem and its solution by the BMS algorithm, we discuss another problem closely related to it as well as some concepts which are important in this paper, where we call these problems

---

S. Sakata

The University of Electro-Communications, Chofu-shi, Tokyo 182-8585, Japan  
e-mail: [sakata@ice.uec.ac.jp](mailto:sakata@ice.uec.ac.jp)

*primal* and *dual*, respectively. The duality of the two problems are similar to the duality discussed by Mora (2009a). About these details we give several remarks in the following. Although the concept of “functional” given in *ibid.* is mathematically natural and more general, we use the terminology of “array” instead of that of functional for the convenience of our discussions. The descriptions of this chapter are rather elementary and intuitive, which are not necessarily refined mathematically as in *ibid.* The topic also turns to be a history of Gröbner basis in the world of Coding Theory.

Now we start to consider sequences (or 1-D arrays) and linear recurrences satisfied by them. The following is a linear recurrence over the real number field  $\mathbb{R}$ , which is satisfied by the famous Fibonacci sequence:

$$s_{j+2} - s_{j+1} - s_j = 0, \quad j \geq 0$$

When we start with the initial values  $s_0 = 1, s_1 = 1$ , we have not only the 1-D array  $(s_j) = (1, 1, 2, 3, 5, 8, \dots)$  but also an explicit form of the  $j$ -th element  $s_j$  for any  $j \in \mathbb{N}$  (over a finite field we have another array, of course). Well, we generalize such 1-D arrays and 1-D linear recurrences to multidimensional arrays and multi-dimensional linear recurrences. For example, we consider the following system of two-dimensional (2-D) linear recurrences over  $\mathbb{F}$ :

$$\begin{cases} u_{i+2,j} + u_{i,j} = 0 \\ u_{i+1,j+1} + u_{i,j} = 0, \quad (i, j) \in \mathbb{N}^2 \\ u_{i,j+2} + u_{i,j} = 0 \end{cases}$$

In general such a condition as above is called a system of constant-coefficient *linear recurrences* or *(partial) finite difference equations*. Given a system of  $n$ -Dimensional ( $n$ -D) linear recurrences over any field  $\mathbb{F}$ , we want to find all  $n$ -D arrays satisfying them. It is just a *digital* version of finding the general solutions of a system of (homogeneous) constant-coefficient linear partial differential equations. We want to obtain not only a special solution but also the general solutions (the whole set of solutions). We treat multiple recurrences satisfied by  $n$ -D arrays. To discuss our problem in general we need some notation as follows.

An  $n$ -D array over a field  $\mathbb{F}$  is a mapping  $u$  from  $\mathbb{N}^n$  into  $\mathbb{F}$ . An array can be extended to a mapping (functional) from the  $n$ -variate polynomial ring  $\mathcal{P} = \mathbb{F}[X_1, \dots, X_n]$  into  $\mathbb{F}$  so that a refinement of our discussions on duality can be given as in Mora (2009a). Since an  $n$ -D array  $u$  identifies a field element in  $\mathbb{F}$  to an  $n$ -dimensional vector  $a = (a_1, \dots, a_n) \in \mathbb{N}^n$  of nonnegative integers (and hence to any term  $X^a := X_1^{a_1} \cdots X_n^{a_n}$ ), we can associate to  $u$  the unique functional  $L$  taking the values of  $u$  on the corresponding term (and vice versa). If we know the values of a functional  $L$  on all terms, then by linearity we know all of its values. Moreover, the value of  $L$  on one term is completely independent from its value on any other term. We persist in using the terminology of *array*, which has been used in most literature on the (applications of) BMS algorithm. We denote an array  $u$  also as  $u = (u_a)_{a \in \mathbb{N}^n}$ , where  $u_a := u(a) \in \mathbb{K}$  and we consider these elements to be arranged on the whole  $n$ -D integral lattice which is identified with  $\mathbb{N}^n$ . Let  $\mathcal{A}$  be the set of all  $n$ -D arrays

over  $\mathbb{F}$  defined on  $\mathbf{N}^n$ , and introduce basic operations upon arrays  $u \in \mathcal{A}$ . Naturally, we have the sum of two arrays  $u = (u_a), v = (v_a) \in \mathcal{A}$  as  $u + v = (u_a + v_a) \in \mathcal{A}$ , and the scalar product of  $u$  by an element  $c$  of the coefficient field  $\mathbb{F}$  as  $cu = (cu_a) \in \mathcal{A}$ . Furthermore, we consider polynomials  $f = \sum_{a \in \text{supp}(f)} c(f, a)X^a \in \mathcal{P}$ . In this paper we often refer to exponents (integer vectors)  $a = (a_1, \dots, a_n) \in \mathbf{N}^n$  as basic entities instead of terms  $\tau = X^a$  so that we denote the coefficient of  $\tau = X^a$  as  $c(f, a)$  instead of  $c(f, X^a)$ . We call the (finite) set of exponents  $a$  ( $\in \mathbf{N}^n$ ) of its nonzero monomials  $c(f, a)X^a$  (having the nonzero coefficient  $c(f, a) \in \mathbb{K} \setminus \{0\}$ ) by the name of the *support*<sup>1</sup> of  $f$  and denote it as  $\text{supp}(f)$  ( $\subset \mathbf{N}^n$ ). A polynomial  $f \in \mathcal{P}$  is operated on an array  $u \in \mathcal{A}$  (it is the same as multiplying a functional  $L$  by  $f$ ) so that the following array  $v$  is obtained:

$$v = f \circ u := (v_b) \in \mathcal{A}, \quad v_b := \sum_{a \in \text{supp}(f)} c(f, a)u_{a+b}, \quad b \in \mathbf{N}^n$$

This *polynomial operation* ‘ $f \circ$ ’ is just a transformation of an array  $u$  to another array  $v$ . In particular, the operation by the monomial  $f = X_j$ ,  $1 \leq j \leq n$  is a unit shift along the  $X_j$ -axis (to the negative direction), where  $v = X_j \circ u = (v_a)$ ,  $a \in \mathbf{N}^n$  has the elements  $v_{a_1, \dots, a_j, \dots, a_n} = u_{a_1, \dots, a_j+1, \dots, a_n}$  (the elements of  $u$  which are put out of the domain  $\mathbf{N}^n$  are pruned away). For example, in case of  $n = 1$ , for  $X := X_1$  and  $u = (u_i)$ , the unit-shifted array  $v = X \circ u = (v_i)$  has the elements  $v_i = u_{i+1}$ ,  $i \in \mathbf{N}$ , and the double-shifted array  $w = X^2 \circ u = (w_i)$  has  $w_i = u_{i+2}$ ,  $i \in \mathbf{N}$ , etc.<sup>2</sup> Consequently, the module  $\mathcal{A}$  is a  $\mathcal{P}$ -module, i.e. a module with the ring  $\mathcal{P}$  of operators. By using this notation, we can write any linear recurrences with the *characteristic polynomials*  $F = \{f^{(1)}, \dots, f^{(\mu)}\}$  ( $\subset \mathcal{P}$ ) as follows,

$$f^{(i)} \circ u = 0, \quad 1 \leq i \leq \mu, \tag{1}$$

where 0 is the all-zero array. From now on, we do not distinguish between linear recurrences and the corresponding characteristic polynomials, identifying them. That is, for simplicity, provided that the formula (1) holds, we often say that the array  $u$  satisfies the polynomial  $f^{(i)}$ , and that ‘the polynomial  $f^{(i)}$  is valid for the array  $u$ ,’ etc. For a given  $F \subset \mathcal{P}$ , it is easy to see that the set  $\mathcal{A}(F)$  of solutions  $u$  of (1) is a  $\mathcal{P}$ -submodule of the  $\mathcal{P}$ -module  $\mathcal{A}$ , since  $f \circ (g \circ u) = (fg) \circ u$  for  $f, g \in \mathcal{P}$ . For example, for a univariate polynomial  $f = x^2 - x - 1$  over  $\mathbb{R}$ ,  $\mathcal{A}(f)$  is the set of 1-D arrays (Fibonacci sequences)  $u = (u_i)$  which are obtained by setting any initial values  $u_0, u_1$  and then uniquely by determining the other values  $u_i$ ,  $i \geq 2$  iteratively with the linear recurrence  $f \circ u = 0$ . In general, for any polynomial set  $F \subset \mathcal{P}$  and the ideal  $\mathbf{I}(F) := \langle F \rangle_{\mathcal{P}}$  ( $\subset \mathcal{P}$ ) generated by  $F$ ,  $\mathcal{A}(F) = \mathcal{A}(\mathbf{I}(F)) := \{u \in \mathcal{A} \mid f \circ u = 0, f \in \mathbf{I}(F)\}$ .

<sup>1</sup>In Mora (2009a) the support is defined as a subset of the whole set  $\mathcal{T}$  of terms  $X^a$ ,  $a \in \mathbf{N}^n$ .

<sup>2</sup>Trivially, by multiplying a polynomial  $g = \sum_{0 \leq i \leq d} g_i X^i$  with  $X$ , one gets the polynomial  $\bar{g} = Xg = \sum_{0 \leq i \leq d} g_i X^{i+1} = \sum_{1 \leq i \leq d+1} g_{i-1} X^i$ , where the array of coefficients of its monomials is obtained by shifting to the positive direction:  $(g_i) \rightarrow (g_{i-1})$  in contrast with the above shift (to the negative direction) by operation  $X$ .

As is seen, in case of  $n = 1$ , we can easily obtain  $\mathcal{P}$ -submodules  $\mathcal{A}(f)$  and  $\mathcal{A}(F)$  of 1-D arrays. In particular, for  $F = \{f^{(1)}, \dots, f^{(\mu)}\} (\subset \mathbb{F}[X])$ ,  $\mathcal{A}(F) = \mathcal{A}(g)$ , where  $g = \gcd(F)$  ( $\gcd$  = greatest common divisor). However, it is not so easy to obtain  $\mathcal{A}(F)$  in case of  $n \geq 2$  as in case of  $n = 1$ . In case of  $n \geq 2$ , it is difficult not only to give the solutions of a system of homogeneous linear recurrences (1) but also to specify even the positions of initial values. In fact, the Buchberger algorithm gives the solution of the present problem, which we will mention in the next section. For our discussions, we need some variants of basic notion from the Gröbner basis theory. In this paper we consider any term ordering  $<$  over  $\mathbf{N}^n$ , although it usually is defined over the set  $\mathcal{T} = \{X^a \mid a \in \mathbf{N}^n\}$  of terms in the Gröbner basis theory. Furthermore, we often consider the leading exponent  $\text{le}(f) := \max_{<} \text{supp}(f) (\in \mathbf{N}^n)$  of  $f$  instead of the corresponding leading term  $\mathbf{T}(f) = X^{\text{le}(f)} \in \mathcal{T}$ .

## 2 Generating Arrays

We consider the problem of initial value positions via the following example in case of  $n = 2$ . Now we assume for a set of polynomials  $F = \{f^{(1)}, \dots, f^{(\mu)}\} (\subset \mathbb{F}[X_1, X_2])$  that the leading exponents  $\text{le}(f^{(i)}) = d^{(i)} = (d_1^{(i)}, d_2^{(i)}) \in \mathbf{N}^2$ ,  $1 \leq i \leq \mu$  of its elements satisfy

$$d_1^{(1)} > d_1^{(2)} > \dots > d_1^{(\mu-1)} > d_1^{(\mu)} = 0, \quad d_2^{(1)} = 0 < d_2^{(2)} < \dots < d_2^{(\mu-1)} < d_2^{(\mu)}$$

Then,  $\mathbf{N}^2$  can be split into two parts:

$$\Sigma(F) := \{a \in \mathbf{N}^2 \mid a \geq_P d^{(i)}, 1 \leq^3 i \leq \mu\}, \quad \Delta(F) := \mathbf{N}^2 \setminus \Sigma(F),$$

where  $\geq_P$  is the natural partial ordering over  $\mathbf{N}^n$ . These subsets have the following properties and are called *stable* sets (sometimes the former and latter sets are called *upper* and *lower* sets, respectively),

$$\begin{aligned} a \in \Sigma(F), \quad b \in \mathbf{N}^2, \quad a \leq_P b &\Rightarrow b \in \Sigma(F); \\ a \in \Delta(F), \quad b \in \mathbf{N}^2, \quad a \geq_P b &\Rightarrow b \in \Delta(F). \end{aligned}$$

If  $F$  is a Gröbner basis (w.r.t.  $<$ ) of an ideal  $\mathbf{I}$ , there are complementary subsets  $\mathbf{T}(\mathbf{I})$  and  $\mathbf{N}(\mathbf{I}) (\subset \mathcal{T})$  which are determined uniquely by  $\mathbf{I}$  (Mora 2009a). Now we consider any stable subsets without assuming any knowledge of Gröbner basis. The latter set  $\Delta(F)$  called *delta-set* or *footprint* seemingly can be used as the initial value positions. That is, after having specified any values  $u_a \in \mathbb{F}$ ,  $a \in \Delta(F)$  as the initial values, we proceed to find each of the remaining values  $u_b$ ,  $b \in \Sigma(F)$  iteratively by using the following (pseudo)algorithm of generating an array up to an prescribed position  $r \in \mathbf{N}^2$ . In the following we denote the next greater point (w.r.t. the term ordering  $<$ ) of any point  $a \in \mathbf{N}^2$  as  $a \oplus 1$ , and define  $\overline{\text{supp}}(f) := \text{supp}(f) \setminus \{\text{le}(f)\} (\subset \mathbf{N}^2)$ ,  $\Sigma^r := \{b \in \mathbf{N}^2 \mid b < r\}$ ;  $\min_{<} \Sigma(F)$  is the minimum element of  $\Sigma(F)$  w.r.t.  $<$ .

**Table 1** Initial values  $u_a$ ,  $a \in \Delta(F)$  and partial array  $u_a, a \in \Sigma^{(1,2)}$

$a_1 \setminus a_2$	0	1	2	3	$a_1 \setminus a_2$	0	1	2	3
0	1	0			0	1	0	1	
1		1			1		1	1	
2					2		1	0*	
3					3		1		
4					4				

**Algorithm 1** Generating an array  $u_b$ ,  $b \in \Sigma^r$ ;

Step 1 (initialization):  $b := \min_{<} \Sigma(F)$ ;

Step 2 (computation): if  $b \in \Sigma(F)$  then

begin let  $i$  be any  $i$ ,  $1 \leq i \leq \mu$  s.t.  $d^{(i)} \leq_P b$ ;

$$u_b := \frac{1}{\text{lc}(f^{(i)})} \left( - \sum_{a \in \overline{\text{supp}}(f^{(i)})} c(f^{(i)}, a) u_{a+b-d^{(i)}} \right)$$

end;

Step 3 (termination):  $b := b \oplus 1$ ; if  $b < r$  then go to Step 2 else stop.

Although it seems that we could find an array  $u$  before the terminal point  $r$  by using this algorithm, it will turn out naturally that we do not always succeed in getting a proper array having the specified initial values and satisfying all of the given linear recurrences. In Step 2, the value  $u_b$  is determined by using the polynomial  $f^{(i)}$  so that one of the desired conditions

$$f^{(i)}[u]_b := \sum_{a \in \text{supp}(f^{(i)})} c(f^{(i)}, a) u_{a+b-d^{(i)}} = 0$$

is satisfied, but some conditions corresponding to other polynomials  $f^{(j)}$ ,  $j \neq i$  with  $d^{(j)} \leq_P b$  might not always be satisfied. Consider the previous example (1) over  $\mathbb{F}_2$ . The linear recurrences are specified by  $F = \{f^{(1)} := X_1^2 + 1, f^{(2)} := X_1 X_2 + 1, f^{(3)} := X_2^2 + 1\}$ , and the delta-set is  $\Delta(F) = \{(0, 0), (1, 0), (0, 1)\}$ . Starting with the initial values shown in the left half of Table 1 we proceed to find the other values of  $u$  iteratively w.r.t. the graded reverse lexicographic ordering  $<$  (i.e. the term ordering with the weight  $w = (1, 1)$  associated with the reverse lexicographic ordering  $X_1 <_L X_2$ ). Then, we have the intermediate result shown in the right half of Table 1.

The value  $u_{2,1} = 0$  with signature \* is found by using  $f^{(1)}$ , but it does not satisfy the linear recurrence of  $f^{(2)}$ . Thus, there exists no array  $u$  which has the initial values and is a solution of the linear recurrences (1) over  $\mathbb{F}_2$ . In other words, the above delta-set is not appropriate as a set of positions for initial values. On taking into consideration the properties of Gröbner basis, it might be hit upon that the polynomial

set  $F$  is not a Gröbner basis and that the corresponding delta-set is too large to be a set of positions for initial values so that the algorithm<sup>3</sup> for generating arrays fails to find proper arrays. In fact, it is easy to see that the reduced Gröbner basis (w.r.t. the term ordering  $<$  with  $w = (1, 1)$ ) of the ideal  $I = \langle F \rangle_{\mathcal{P}}$  is  $\{X_1^2 + 1, X_2 + X_1\}$ , and that the proper set of positions for initial values is  $\{(0, 0), (1, 0)\}$ . As we have seen, the problem of finding the proper set of positions for initial values, given a set of linear recurrences or its characteristic polynomials, is just identical with that of finding a Gröbner basis of the ideal  $\mathbf{I}(F)$ . Furthermore, iteratively from  $F$ , we can find a polynomial  $f^{(b)} := X^b - \sum_{a \in \Delta(F)} c(f^{(b)}, a)X^a \in \mathbf{I}(F)$ , by which we can get the value  $u_b$  for any  $b \in \Sigma(F)$  directly from any given values  $u_a$ ,  $a \in \Delta(F)$ , so that the so-called S-polynomial at any outer corner point outside  $\Delta(F)$  can be obtained. By repeating reductions of such S-polynomials modulo  $F$  and consequent modifications of  $F$  and  $\Delta(F)$ , we finally get a Gröbner basis. Of course, it is just the Buchberger algorithm. Let us illustrate it with the previous example. We get  $X_2 f^{(1)} = X_1^2 X_2 + X_2 \in \mathbf{I}(F)$  from  $f^{(1)} = X_1^2 + 1$ , and  $X_1 f^{(2)} = X_1^2 X_2 + X_1$  from  $f^{(2)} = X_1 X_2 + 1$ . Then, we obtain the S-polynomial  $f^{(3)} := X_2 f^{(1)} - X_1 f^{(2)} = X_2 + X_1 \in \mathbf{I}(F)$  at the corner point  $(2, 1)$ , which is not reducible further modulo  $F$ . Finally,  $\{f^{(1)}, f^{(3)}\}$  turns out to be the reduced Gröbner basis.

As a summary, we have that the problem of finding a module of linear recurrences is equivalent to that of finding a Gröbner basis of its characteristic polynomials, and thus these problems can be solved by the same algorithm. In the world of Coding Theory, it is a historical fact that the concept (Ikai et al. 1976) equivalent to Gröbner basis and an algorithm (Sakata 1981) equivalent to the Buchberger (1965, 1985, 2006) algorithm were introduced in the process of solving a certain problem of constructing a kind of multidimensional codes independently. In general, as it is shown in Lemma 4 of Mora (2009a), the general solution of the system of linear recurrences (1) is just the  $\mathcal{P}$ -module  $L$  which is dual (in the sense of Mora 2009a) to the ideal  $\mathbf{I}$  generated by  $F = \{f^{(1)}, \dots, f^{(\mu)}\}$  and  $\dim_{\mathbb{F}} L = \#\mathbf{N}(\mathbf{I})$ ,<sup>4</sup> where  $\mathbf{N}(\mathbf{I})$  is the delta-set of the Gröbner basis of  $\mathbf{I}$  which is called *Gröbner escalier* in Mora (2009a), provided  $\dim_{\mathbb{F}} L < \infty$ , i.e.  $\mathbf{I}$  is a zero-dimensional ideal.

### 3 BMS Algorithm

Our main problem is just the inverse (or rather dual from the viewpoint of duality in Mora 2009a) to the problem of finding the general solution of a given system of linear recurrences which we have discussed in the previous section. Now, for an integer  $\mu$ , we are given a pair of sets  $U := \{u^{(i)} \mid 1 \leq i \leq \mu\}$  and  $V := \{v^{(i)} \mid 1 \leq$

<sup>3</sup>As it is seen from these considerations, Algorithm 1 can be an actual algorithm if and only if  $F$  is a Gröbner basis.

<sup>4</sup>In Sakata (1978), this fact is described in the terminology of array, which is a little bit different from Mora (2009a).

$i \leq \mu\}$  of infinite (periodic)  $n$ -D arrays, i.e.  $U, V \subset \mathcal{A}$ , and consider the following linear recurrences corresponding to (unknown) polynomials  $f \in \mathcal{P}$ :

$$f \circ u^{(i)} = v^{(i)}, \quad 1 \leq i \leq \mu \quad (2)$$

We might be required to find a valid polynomial  $f$  having a (unknown) minimal leading exponent  $\text{le}(f)$ . In most part of this paper we concentrate upon the homogeneous problem with the right-hand side arrays  $v^{(i)} = 0$ ,  $1 \leq i \leq \mu$ , leaving the non-homogeneous problem (see Sect. 4).

It is easy to see that the following set of polynomials is an ideal of  $\mathcal{P} = \mathbb{F}[X_1, \dots, X_n]$ , which we call the *characteristic ideal* of the given set  $U$  of arrays  $u^{(i)}$ ,  $1 \leq i \leq \mu$ :

$$\mathbf{I}(U) := \{f \in \mathcal{P} \mid f \circ u^{(i)} = 0, 1 \leq i \leq \mu\}$$

This is the ideal dual to a  $\mathcal{P}$ -module generated by  $U$  in the terminology of Mora (2009a).

In this section, we treat only the case of a single array,<sup>5</sup> i.e.  $U = \{u\} \subset \mathcal{A}$ , and we will give a method of finding a Gröbner basis of the characteristic ideal  $\mathbf{I}(u) := \{f \in \mathcal{P} \mid f \circ u = 0\}$ . We want to find a set of polynomials  $f = \sum_{a \in \text{supp}(f)} c(f, a)X^a$  with a minimal leading exponent  $d = \text{le}(f)$  which satisfy  $f \circ u = 0$ , i.e.

$$\sum_{a \in \text{supp}(f)} c(f, a)u_{a+b} = 0, \quad b \in \mathbb{N}^n \quad (3)$$

We try to find a set of polynomials with a minimal leading exponent satisfying a (partial) condition specified by a finite part of a given infinite array  $u$ .

To be more precise, we introduce some notations. According to a specified term ordering  $<$  over  $\mathbb{N}^n$ , we arrange the points  $a \in \mathbb{N}^n$  so that we have  $\mathbb{N}^n = \{a^{(0)} = 0, a^{(1)}, a^{(2)}, \dots, a^{(i)}, \dots \mid a^{(i+1)} = a^{(i)} \oplus 1, i \in \mathbb{N}\}$ , and a partial array  $u^b := (u_a)$ ,  $a < b$  for any point  $b \in \mathbb{N}^n$ . If a polynomial  $f = \sum_{a \in \text{supp}(f)} c(f, a)X^a$  ( $\in \mathcal{P}$ ) satisfies for a certain  $r \in \mathbb{N}^n$

$$f[u]_b := \sum_{a \in \text{supp}(f)} c(f, a)u_{a+b-d} = 0, \quad d = \text{le}(f) \leq_P b < r, \quad (4)$$

we say that  $f$  is *valid* (w.r.t.  $u$ ) *before*  $r$ . From now on, having fixed an array  $u$ , we often omit the phrase “w.r.t.  $u$ .” Furthermore, if the condition (4) holds and  $f[u]_r \neq 0$ , then we say that  $f$  is *not valid at the point  $r$  for the first time*. A monic (i.e.  $\text{lc}(f) = 1$ ) polynomial  $f$  which is valid before  $a$  and whose leading exponent  $\text{le}(f)$  is minimal w.r.t. the partial ordering  $\leq_P$  is called a *minimal polynomial of the partial array  $u^a$* . Since there exist in general plural minimal polynomials  $f$  with distinct leading exponents  $\text{le}(f)$  of a given partial array  $u^a$ , we can define a minimal polynomial set  $F(a)$  (or simply,  $F$ ) of  $u^a$  associated with a finite set of

---

<sup>5</sup>For the general case of multiple arrays, see Sect. 4.

points  $D(a) = \{\text{le}(f) \mid f \in F(a)\} \subset \mathbf{N}^n$  s.t. there is a single element  $f \in F(a)$  with  $\text{le}(f) = d$  and  $\text{lc}(f) = 1$  for each  $d \in D(a)$  and there exists no polynomial  $h$  with  $\text{le}(h) \in \Delta(a)$  which is valid (w.r.t.  $u$ ) before  $a$ , where for  $\Sigma_d := \{b \in \mathbf{N}^n \mid b \geq_P d\}$ , we have a complementary pair of subsets  $\subset \mathbf{N}^n$ :

$$\Sigma(a) := \bigcup_{d \in D(a)} \Sigma_d, \quad \Delta(a) := \mathbf{N}^n \setminus \Sigma(a)$$

In addition to  $D(a)$ , letting  $\Gamma_c := \{b \in \mathbf{N}^n \mid b \leq_P c\}$  for  $c \in \mathbf{N}^n$ , we have a finite subset  $C(a) (\subset \mathbf{N}^n)$  s.t.  $\Delta(a) = \bigcup_{c \in C(a)} \Gamma_c$ . We call  $\Delta(a)$  the *delta-set* of  $F(a)$ , which is, roughly speaking, in form of a stack of multidimensional building blocks and whose apices (corner points) are  $c \in C(a)$ . As above-mentioned, there exists no polynomial  $h$  with  $\text{le}(h) \in \Delta(a)$  which is valid before  $a$ . These subsets  $D(a)$ ,  $C(a)$ ,  $\Sigma(a)$  and  $\Delta(a)$  are unique for the given array  $u^a$ , but a minimal polynomial set  $F(a)$  is not necessarily unique for  $u^a$ . In view of the definition of minimal polynomial set  $F(a)$ ,  $\Delta(a) \subseteq \Delta(a \oplus 1)$ . Similar notations can be used for an infinite array  $u$ , e.g. a minimal polynomial set  $F(\subset \mathcal{P})$  of  $u$ , the delta-set  $\Delta(\subset \mathbf{N}^n)$  of  $u$ , etc. if they exist.

The BMS algorithm is just to find a minimal polynomial set  $F(a)$  of a given partial array  $u^a$  for a fixed point  $a \in \mathbf{N}^n$ . Starting with the origin  $0 \in \mathbf{N}^n$ , we proceed to find a minimal polynomial set  $F(b)$  of the partial array  $u^b$  iteratively at each point  $b \leq a$  accordingly to the term ordering  $<$ . If  $f \in F(b)$  is valid still at  $b \oplus 1$ , then  $f \in F(b \oplus 1)$ . However, if some  $f \in F(b)$  is not valid at  $b$ , then we must update these invalid  $f$ . Whether  $\Delta(a \oplus 1) = \Delta(a)$  or not depends on certain relations among  $a$ ,  $D(a)$  and  $C(a)$ . The following basic lemma (Sakata 1990) describing this fact stipulates the main procedure of the BMS algorithm.

**Lemma 1** *If a polynomial  $f$  is not valid (w.r.t.  $u$ ) for the first time at  $a$ , i.e.*

$$f[u]_b = 0, \quad d = \text{le}(f) \leq_P b < a; \quad f[u]_a \neq 0,$$

*then there exists no polynomial  $g$  with  $\text{le}(g) = d' \leq_P a - d$  satisfying the following condition:*

$$g[u]_b = 0, \quad d' \leq_P b \leq a$$

Lemma 1 is very important because it determines the delta-set  $\Delta(a \oplus 1)$  and its complement  $\Sigma(a \oplus 1)$ , where the minimal (w.r.t. the partial ordering  $<_P$ ) points of  $\Sigma(a \oplus 1)$  are just identical with the set  $\{\text{le}(f) \mid f \in F(a \oplus 1)\}$  of leading exponents of all elements of a minimal polynomial set  $F(a \oplus 1)$ .

Based on Lemma 1 we define the *discrepancy*, *fail* and *span*<sup>6</sup> of  $f$ , respectively, as

$$\text{dis}(f) := f[u]_a (\neq 0), \quad \text{fail}(f) := a, \quad \text{span}(f) := a - d (= \text{fail}(f) - \text{le}(f))$$

---

<sup>6</sup>This is not the usual ‘linear span.’

In addition we introduce the following notation for later convenience.

$$\begin{aligned}\text{Val}(u^b) &:= \{f \in \mathcal{P} \mid f[u]_a = 0, \text{le}(f) \leq_P a < b\} \\ \text{mVal}(u^b) &:= \{f \in \text{Val}(u^b) \mid \text{le}(f) : \text{minimal}\} \\ \text{Aux}(u; c) &:= \{g \in \mathcal{P} \mid \text{span}(g) = c\} \\ \text{mAux}(u; c) &:= \{g \in \text{Aux}(u; c) \mid \text{le}(g) : \text{minimal}\}\end{aligned}$$

Associated with the finite subset  $C(a)$  related to the delta-set  $\Delta(a)$ , we have a finite set of polynomials  $G(a) := \{g \mid \text{span}(g) \in C(a)\}$ , which we call an *auxiliary polynomial set* of  $u^a$ . An auxiliary polynomial  $g \in G(a)$  is characterized by the property that it has a maximal (w.r.t. the partial ordering  $\leq_P$ )  $\text{span}(g)$  among the polynomials s.t.  $\text{fail}(g) < a$ . If a minimal polynomial  $f \in F(a)$  fails to be valid at  $a$ , minimal polynomial(s)  $f' \in F(a \oplus 1)$  at  $a \oplus 1$  can be obtained by using appropriate auxiliary polynomial(s)  $g \in G(a)$  (if it exists) as shown in Lemma 2 or without any  $g \in G(a)$ . First we have in view of Lemma 1 that, if there exists a polynomial  $f \in F(a)$  with  $d = \text{le}(f)$  which is not valid at  $a$  and  $a - d \notin \Delta(a)$ , then  $\Delta(a \oplus 1) \neq \Delta(a)$ . Thus, we define  $F_{\text{fail}} := \{f \in F(a) \mid \text{fail}(f) = a\}$ ,  $F_{\text{fall}} := \{f \in F_{\text{fail}} \mid a - d \notin \Delta(a)\}$ ,  $D_{\text{fall}} := \{\text{le}(f) \mid f \in F_{\text{fall}}\}$ . Furthermore, for  $c = (c_i)_{1 \leq i \leq n} (\in C(a))$ , let  $\max(d, a - c) := (\max\{d_i, a_i - c_i\})_{1 \leq i \leq n} (\in \mathbb{N}^n)$ , and let  $D'$  be the set of minimal elements  $d'$  in  $D'' := \{d' := \max(d, a - c) \mid d \in D_{\text{fall}}, c \in C(a)\} (\subset \mathbb{N}^n)$ , and let  $\hat{D}$  be the set of minimal elements in  $\Sigma(a) \setminus \Gamma_a$ . Now we have the following lemma about how to update  $F$  (Sakata 1990).

**Lemma 2** (1) For  $f \in F_{\text{fail}} \setminus F_{\text{fall}}$ , there exists  $c \in C(a)$  s.t.  $d \geq_P a - c$ . In this case, by using an auxiliary polynomial  $g \in G(a)$  s.t.  $\text{span}(g) = c$ , we obtain

$$h := f - \frac{\text{dis}(f)}{\text{dis}(g)} X^{d-(a-c)} g \in F(a \oplus 1)$$

(2) For a pair  $(f, g) \in F_{\text{fall}} \times G(a)$  with  $d = \text{le}(f)$ ,  $c = \text{span}(g)$ , respectively, if it holds that  $d' := \max(d, a - c) \in D'$ , then we obtain

$$h := X^{(a-c)-dd'-d} f - \frac{\text{dis}(f)}{\text{dis}(g)} g \in F(a \oplus 1)$$

(3) For  $\hat{d} \in \hat{D}$ , if there exists no  $d' \in D'$  s.t.  $\hat{d} \geq_P d'$ , then, by using  $f \in F_{\text{fail}}$  s.t.  $\hat{d} \geq_P d = \text{le}(f)$ , we obtain

$$h := X^{\hat{d}-d} f \in F(a \oplus 1)$$

Based on the above observations we have the following form of the BMS algorithm, whose validity can be proven based on Lemmas 1, 2, where some notational simplicities are used, i.e. minimal polynomial set  $F(b)$  and auxiliary polynomial set  $G(b)$  at each point  $b$  are denoted simply as  $F$  and  $G$ , respectively. For  $f \in F$ ,

$g \in G$ , let  $d := \text{le}(f)$ ,  $c := \text{span}(g)$ ,  $d_f := \text{dis}(f)$ ,  $d_g := \text{dis}(g)$ ,

$$D = \{d = \text{le}(f) \mid f \in F\}, \quad C = \{c = \text{span}(g) \mid g \in G\},$$

and  $\Sigma = \Sigma(b)$ ,  $\Delta = \Delta(b)$  at the beginning of Step 2. A simple example of its computation is shown in Appendix A.

**Algorithm 2** (BMS algorithm) Finding a minimal polynomial set of a finite  $n$ -D array  $u^r$  over  $\mathbb{F}$  (Sakata 1988, 1990);<sup>7</sup>

Step 1 (initialization):  $b := 0$ ;  $F := \{1\}$ ;  $D := \{0\}$ ;  $\Sigma := \mathbf{N}^n$ ;  
 $G := \emptyset$ ;  $C := \emptyset$ ;  $(\Delta := \emptyset)$ ;

Step 2 (discrepancy): for each  $f \in F$ ,  $d_f := f[u]_b$ ;

$$F_{\text{fail}} := \{f \in F \mid d_f \neq 0\};$$

$$F_{\text{fall}} := \{f \in F_{\text{fail}} \mid \exists c \in C \text{ s.t. } d \geq_P b - c\};$$

$$D_{\text{fall}} := \{d = \text{le}(f) \in D \mid f \in F_{\text{fall}}\}; \hat{D} := \{\text{minimal } \hat{d} \in \Sigma \setminus \Gamma_b\};$$

$$D'' := \{\max(d, b - c) \mid d \in D_{\text{fall}}, c \in C\}; D' := \{\text{minimal } d' \in D''\};$$

Step 3 (updating): (1) for each  $f \in F_{\text{fail}} \setminus F_{\text{fall}}$

$$\begin{aligned} & \text{begin } h := f - d_f X^{d-(b-c)} g \text{ (for } g \in G \text{ s.t. } d \geq_P b - c\}; \\ & F' := F \cup \{h\} \text{ end;} \end{aligned}$$

for each  $(f, g) \in F_{\text{fall}} \times G$  s.t.  $d' := \max(d, b - c) \in D'$

$$\begin{aligned} & \text{begin } h := X^{\hat{d}-dd'-d} f - d_f g; F' := F \cup \{h\} \text{ end;} \\ & \text{for each } \hat{d} \in \hat{D} \text{ if } \nexists d' \in D' \text{ s.t. } \hat{d} \geq_P d' \text{ then} \end{aligned}$$

$$\text{for } f \in F_{\text{fall}} \text{ s.t. } d \leq_P \hat{d}$$

$$\begin{aligned} & \text{begin } h := X^{d-d} f; F' := F \cup \{h\} \text{ end;} \\ & (2) \quad F := F' \setminus F_{\text{fail}}; G'' := \{g \in G \mid \exists f \in F_{\text{fall}} \text{ s.t. } c <_P b - d\}; \\ & G := (G \cup \{\frac{1}{d_f} f \mid f \in F_{\text{fall}}\}) \setminus G''; \\ & D := \{\text{le}(f) \mid f \in F' \setminus F_{\text{fall}}\}; \Sigma := \bigcup_{d \in D} \Sigma_d; (\Delta := \bigcup_{c \in C} \Gamma_c); \\ & C := (C \cup \{b - d \mid \exists f \in F_{\text{fall}} \text{ s.t. } b - d >_P c\}) \\ & \quad \setminus \{c \in C \mid \exists f \in F_{\text{fall}} \text{ s.t. } b - d >_P c\}; \end{aligned}$$

Step 4 (termination):  $b := b \oplus 1$ ; if  $b < r$  then go to Step 2 else stop.

A minimal polynomial set  $F(b)$  is not necessarily unique for the given array  $u$ . Let  $\mathcal{F}$  be the class of all reduced minimal polynomial sets  $F = F(b)$  of  $u^b$ , where  $F \in \mathcal{F}$  is said to be *reduced* iff any  $f \in F$  has support  $\text{supp}(f)$  s.t.  $\overline{\text{supp}}(f) := \text{supp}(f) \setminus \{\text{le}(f)\}$  is contained in the delta-set  $\Delta := \Delta(a)$ . Let  $\text{le}(f) + \Delta := \{\text{le}(f) + a \mid a \in \Delta\}$ . Now, we have the following theorems (Sakata 1990) about the complete class  $\mathcal{F} = \mathcal{F}(b)$  which consists of all minimal polynomial sets  $F = F(b)$  of  $u^b$  and the condition of uniqueness of  $F$ , i.e.  $\#\mathcal{F} = 1$ .

**Theorem 1** (Complete class) *Let  $F \in \mathcal{F}$  ( $= \mathcal{F}(b)$ ) and  $G = G(b)$  be a minimal polynomial set and an auxiliary polynomial set of  $u^b$  with  $D = D(b)$ ,  $C = C(b)$*

---

<sup>7</sup>The 1-D case of this algorithm is reduced to a refined version of the well-known Berlekamp–Massey (BM) algorithm (Berlekamp 1968; Massey 1969).

s.t.  $\Sigma = \Sigma(b) = \bigcup_{d \in D} \Sigma_d$  and  $\Delta = \Delta(b) = \bigcup_{c \in C} \Gamma_c$ . Take a minimal polynomial  $f \in F$  and an  $F' \in \mathcal{F}$ . Then, any  $f' \in F'$  with  $\text{le}(f') = \text{le}(f) = d \in D$  is of the form:

$$f' = f + \sum_{g \in G_d} h_g g,$$

where  $h_g \in \mathcal{P}$ ,  $\text{le}(h_g) \leq d + \text{span}(g) - b$ , and  $G_d := \{g \in G \mid d + \text{span}(g) \geq_P b\}$ .

**Theorem 2** (Uniqueness) *Let  $F \in \mathcal{F}$ . Then, we have that  $\#\mathcal{F} = 1$  iff*

$$\bigcup_{f \in F} (\text{le}(f) + \Delta) \subseteq \Sigma^b,$$

or in other words,

$$\max_< \{\text{le}(f) + \text{fail}(g) - \text{le}(g) \mid f \in F, g \in G\} < b,$$

where  $G$  is an auxiliary polynomial set of  $u^b$ , and  $\max_< \{\dots\}$  is the maximum (w.r.t.  $<$ ) element of the set  $\{\dots\}$ .

**Theorem 3** (Gröbner basis of  $\mathbf{I}(u)$ ) *Let  $P = \{\sum_{1 \leq i \leq n} c_i a^{(i)} \in \mathbf{N}^n \mid 0 \leq c_i \leq 1, c_i \in \mathbf{Q}, 1 \leq i \leq n\}$  be a fundamental period parallelopiped  $P \subset \mathbf{N}^n$  of an infinite  $n$ -D periodic array  $u$  s.t.  $u_{b+a^{(i)}} = u_b$  for any  $b \in \mathbf{N}^n$ , and let  $2P := \{a + c \mid a, c \in P\}$ . Then, if the subset  $\Sigma^b = \{a \in \mathbf{N}^n \mid a < b\}$  contains  $2P$ , a minimal polynomial set  $F$  of the partial array  $u^b$  is a Gröbner basis of  $\mathbf{I}(u)$ .*

In real applications of the BMS algorithm, we usually know in advance the approximate size  $\#\Delta(F)$  for the delta-set  $\Delta(F)$  of the Gröbner basis  $F$  (without any other knowledge of  $F$  itself). In such cases, in view of Theorem 2, we can terminate the iterations of the BMS algorithm much earlier than described in Theorem 3. We assume the term ordering  $<$  with the weight  $w = (w_i)_{1 \leq i \leq n}$  whose elements  $w_i$  are almost equal to each other, i.e.  $w_1 \sim w_2 \sim \dots \sim w_n$ . This is just the case that the Buchberger algorithm has the least complexity. Let  $m := \#\Delta$  for the delta-set  $\Delta$  of the Gröbner basis which is the minimal polynomial set  $F$  at the termination point, and let  $\mu := \#F$  ( $\sim \#G$ ). Then, if the computational complexity<sup>8</sup> of the BMS algorithm is measured as the total number of arithmetic operations over the finite field  $\mathbb{F}$ , in view of  $\mu \sim m^{1-\frac{1}{n}}$ , it is  $\mathcal{O}(\mu m^2) \sim m^{3-\frac{1}{n}}$  when  $n$  is fixed. This complexity is somewhat better than  $\mathcal{O}(m^3)$  of any relevant algorithm based on the usual Gaussian elimination if  $n$  is not large. We should remark that we can have various modifications of the original BMS algorithm and that the computational complexities of these versions are reduced considerably when they are applied to various practical

---

<sup>8</sup>To determine the set of  $D' \cup \hat{D}$  of  $\text{le}(f')$ ,  $f' \in F(b \oplus 1)$  we need to have some combinatorial manipulation, particularly finding minimal (w.r.t.  $<_P$ ) elements  $d'$  of  $D''$ . We omit complexity of such integer operations which are independent from finite field arithmetic.

problems including decoding of algebraic error-correcting codes because they cleverly can make use of the structures or properties of the given input data (i.e. arrays) which depend on each individual problem (Sakata 2009, 1989; Sakata et al. 1995).

## 4 Variations

In this section we present several versions of the BMS algorithm, each of which solves a distinct extension or generalization of the original BMS problem, respectively. The following are a list of these problems.

### (1) Multiarray BMS problem (Sakata 1989; Feng and Tzeng 1989, 1991)

Given a finite set  $U = \{u^{(i)} \mid 1 \leq i \leq \mu\}$  of finite  $n$ -D arrays over  $\mathbb{F}$ , where all component arrays  $u^{(i)}$ ,  $1 \leq i \leq \mu$  are defined over  $\Sigma^r \subset \mathbf{N}^n$  (w.r.t. a fixed term ordering  $<$ ) for a certain point  $r \in \mathbf{N}^n$ ;

Find a minimal polynomial set  $F$  composed of polynomials  $f$  which are valid (w.r.t. every  $u^{(i)}$ ,  $1 \leq i \leq \mu$ ), i.e. satisfy the following conditions

$$\begin{aligned} f[u^{(i)}]_b := \sum_{a \in \text{supp}(f)} c(f, a) u_{a+b-d}^{(i)} &= 0, \\ d = \text{le}(f) \leq_P b < r, \quad 1 \leq i \leq \mu \end{aligned} \tag{5}$$

and have a distinct minimal leading exponent  $d = \text{le}(f)$  among the valid polynomials s.t. ( $\Sigma(F) = \bigcup_{f \in F} \Sigma_{\text{le}(f)}$ ,  $\Delta(F)$ ) is a separation of  $\mathbf{N}^n$  and there exists no valid polynomial  $g \in \mathcal{P}$  with  $\text{le}(g) \in \Delta(F)$ .

This is a multidimensional extension of the multisequence shift-register synthesis problem treated by Feng and Tzeng (1989, 1991).

### (2) Vectorial BMS problem (Sakata 1991)

Similarly to the Gröbner basis theory of modules we define the leading exponent, leading position and leading coefficient of a polynomial vector  $\mathbf{f} = (f^{(1)}, \dots, f^{(m)}) \in \mathcal{P}^m$  ( $= (\mathbb{F}[X_1, \dots, X_n])^m$ ) as follows:

$$\begin{aligned} \text{le}(\mathbf{f}) &:= \max_{<} \{\text{le}(f^{(i)}) \in \mathbf{N}^n \mid 1 \leq i \leq m\} \\ \text{lp}(\mathbf{f}) &:= \max \{i \in [1, m] \mid \text{le}(f^{(i)}) = \text{le}(\mathbf{f})\} \\ \text{lc}(\mathbf{f}) &:= c(f^{(i)}, \text{le}(f^{(i)})) \in \mathbb{K} \quad \text{for } i = \text{lp}(\mathbf{f}), \end{aligned}$$

where  $[1, m] := \{1, \dots, m\} \subset \mathbf{N}$ , and we use the pair  $\text{le}(\mathbf{f}) \in \mathbf{N}^n$ ,  $\text{lp}(\mathbf{f}) \in \mathbf{N}$  instead of  $\mathbf{T}(f) \in \mathcal{T}^m$  in Mora (2009a).

Given an array vector  $\mathbf{u} = (u^{(1)}, \dots, u^{(m)})$  whose components are finite  $n$ -D arrays  $u^{(i)}$  over  $\mathbb{F}$ ,  $1 \leq i \leq m$ , defined over  $\Sigma^r \subset \mathbf{N}^n$  (w.r.t. a fixed term ordering  $<$ ) for a certain point  $r \in \mathbf{N}^n$ ;

Find a minimal polynomial vector set  $\mathbf{F}$  of  $\mathbf{u}$  which is a union of  $m$  subsets  $\mathbf{F}^{(i)} (\subset \mathcal{P}^m)$ ,  $1 \leq i \leq m$ , where each  $\mathbf{F}^{(i)}$  is composed of polynomial vectors

$\mathbf{f} = (f^{(1)}, \dots, f^{(m)}) \in \mathcal{P}^m$  with leading position  $\text{lp}(\mathbf{f}) = i$ ,  $1 \leq i \leq m$ , which are valid w.r.t.  $\mathbf{u}$ , i.e. satisfy the following condition:

$$\mathbf{f}[\mathbf{u}]_b := \sum_{i=1}^m \sum_{a \in \text{supp}(f^{(i)})} c(f^{(i)}, a) u_{a+b-d}^{(i)} = 0, \quad d = \text{le}(\mathbf{f}) \leq_P b < r \quad (6)$$

s.t. there exists no valid polynomial vector  $\mathbf{g}$  with  $\text{lp}(\mathbf{g}) = i$  and  $\text{le}(\mathbf{g}) <_P \text{le}(\mathbf{f})$  for any  $\mathbf{f} \in \mathbf{F}^{(i)}$ ,  $1 \leq i \leq m$ .

Naturally this problem can be generalized to finding a minimal polynomial vector set of a given finite set  $\mathbf{U} = \{\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(L)}\}$  of array vectors. For the set  $\mathbf{U}$  composed of infinite array vectors, it amounts to find a Gröbner basis of the *characteristic module* of  $\mathbf{U}$  which is defined as

$$\mathbf{M}(\mathbf{U}) := \{\mathbf{f} \in \mathcal{P}^m \mid \mathbf{f}[\mathbf{u}^{(l)}]_b = 0, b \in \mathbf{N}^n, 1 \leq l \leq L\}$$

similarly to the characteristic ideal  $\mathbf{I}(U) \subset \mathcal{P}$  of a set  $U$  of infinite  $n$ -D arrays introduced in Sect. 3 (if it exists).

As far as we know, neither any 1-D version of this problem nor its solution had been published before (Sakata 1991).

### (3) Non-homogeneous BMS problem (Sakata 2003)

At the beginning of Sect. 3, we introduced the nonhomogeneous BMS problem (2) for a given pair of sets  $U := \{u^{(i)} \mid 1 \leq i \leq \mu\}$  and  $V := \{v^{(i)} \mid 1 \leq i \leq \mu\}$ . Now we consider the simplest case of  $\mu = 1$  as follows.

Given a pair of finite  $n$ -D arrays  $u, v$  over  $\mathbb{F}$ , where  $u$  and  $v$  are defined over  $2\Sigma^r := \{a + b \mid a, b \in \Sigma^r\}$  and  $\Sigma^r (\subset \mathbf{N}^n)$ , respectively, w.r.t. a fixed term ordering  $<$  for a certain point  $r \in \mathbf{N}^n$ .

Find a set  $F$  of polynomials  $f$  which are valid w.r.t. the given pair  $(u, v^r)$ , i.e. satisfy the following condition

$$f \langle u \rangle_b := \sum_{a \in \text{supp}(f)} c(f, a) u_{a+b} = v_b, \quad 0 \leq b < r, \quad (7)$$

and have a distinct minimal leading exponent  $d = \text{le}(f)$  among the valid polynomials s.t.  $(\Sigma(F) = \bigcup_{f \in F} \Sigma_{\text{le}(f)}, \Delta(F))$  is a separation of  $\mathbf{N}^n$  and there exists no valid polynomial  $g \in \mathcal{P}$  with  $\text{le}(g) \in \Delta(F)$ .

In addition to  $\text{Val}(u^b)$ ,  $\text{mVal}(u^b)$ ,  $\text{Aux}(u; c)$ , and  $\text{mAux}(u; c)$  introduced before, we define

$$\begin{aligned} \text{Val}(u; v^b) &:= \{f \in \mathcal{P} \mid f \langle u \rangle_a = v_a, 0 \leq a < b\} \\ \text{mVal}(u; v^b) &:= \{f \in \text{Val}(u; v^b) \mid \text{le}(f) : \text{minimal}\} \end{aligned}$$

For the 1-D case Sugiyama (1986) gave a solution based on the Euclidean algorithm.

### (4) Submodule BMS problem (Sakata 2007)

For a fixed pair  $(\bar{\Sigma}, \bar{\Delta})$  of stable subsets of  $\mathbf{N}^n$  s.t.  $\bar{\Sigma} = \bigcup_{d \in \bar{D}} \Sigma_d$  and  $\bar{\Delta} =$

$\mathbf{N}^n \setminus \bar{\Sigma} = \bigcup_{c \in \bar{C}} \Gamma_c$ , where  $\bar{D}$  and  $\bar{C}$  are given a priori consistently, we consider a module  $\mathcal{P}(\bar{\Sigma})$  over  $\mathcal{P}$  which is defined to be the set of all polynomials  $f$  with  $\text{supp}(f) \subset \bar{\Sigma}$ . In such a specified module  $\mathcal{P}(\bar{\Sigma})$  we have  $\mathcal{P}$ -submodules and their Gröbner bases, and correspondingly several similar concepts extended from the original BMS problem to the present case. Particularly, given a finite array  $u^r = (u_a)$ ,  $a < r$  (w.r.t. a term ordering  $<$ ) defined over the subset  $\bar{\Sigma}^r = \{a \in \bar{\Sigma} \mid a < r\}$ , a polynomial  $f \in \mathcal{P}(\bar{\Sigma})$  is said to be *valid* for the array  $u^r$  iff the identity of the same form holds as (4). And, a minimal polynomial set  $F \subset \mathcal{P}(\bar{\Sigma})$  of  $u^b$ ,  $b \in \bar{\Sigma}$  is defined similarly together with  $\Sigma(b) = \bigcup_{d \in D} \Sigma_d \subset \bar{\Sigma}$  and  $\Delta(b) = \bar{\Sigma} \setminus \Sigma(b)$ , where there exists no valid polynomial  $g$  (for  $u^b$ ) with  $\text{le}(g) \in \Delta(b)$  and there exists a valid polynomial  $f$  with  $\text{le}(f) = d$  for each  $d \in D$ , etc. In this case we have the following problem:

Given a finite array  $u^r$  (defined over  $\bar{\Sigma}^r \subset \bar{\Sigma}$ );

Find a minimal polynomial set  $F \subset \mathcal{P}(\bar{\Sigma})$  of  $u^r$ .

We have such a problem in decoding two-point codes from curves (Sakata 2007).

#### (5) Semigroup BMS problem (Sakata 1995)

Instead of the usual integral lattice  $\mathbf{N}^n$  and polynomial ring  $\mathcal{P}$  we consider a semigroup  $\bar{\Sigma}$  of the additive group  $\mathbf{Z}^n$  (or an  $n$ -D *convex cone* in geometrical terms) and the corresponding ring  $\bar{\mathcal{P}}$  which are define by a given unimodular matrix  $W = (w_{ij}) \in \mathbf{Z}^{n \times n}$  as follows:

$$\begin{aligned} \bar{\Sigma} &:= \{a \in \mathbf{Z}^n \mid aW \in \mathbf{N}^n\} \\ \bar{\mathcal{P}} &:= \left\{ f = \sum_{a \in \text{supp}(f)} c(f, a)X^a \in \mathbb{F}[X_1, \dots, X_n, X_1^{-1}, \dots, X_n^{-1}] \mid \right. \\ &\quad \left. \text{supp}(f) \subset \bar{\Sigma} \right\} \end{aligned}$$

Over  $\bar{\Sigma}$  we have a special partial ordering  $<_{\bar{P}}$  as follows:

$$a \leq_{\bar{P}} b \Leftrightarrow b - a \in \bar{\Sigma}.$$

We can consider not only ideals of this special ring  $\bar{\mathcal{P}}$  and their Gröbner bases (w.r.t. a specified term ordering  $<$  over  $\bar{\Sigma}$ ) but also a minimal polynomial set  $F(\subset \bar{\mathcal{P}})$  of a (finite or infinite) array  $u$  defined over  $\bar{\Sigma}$ , where any  $f \in F$  satisfies the condition:

$$f[u]_b := \sum_{a \in \text{supp}(f)} c(f, a)u_{a+b-d} = 0, \quad d = (f) \leq_{\bar{P}} b. \quad (8)$$

Thus we have the present problem:

Given a finite array  $u^r$  (defined over  $\bar{\Sigma}^r$  ( $:= \{a \in \bar{\Sigma} \mid a < r\}$ ));

Find a minimal polynomial set  $F(\subset \bar{\mathcal{P}})$  of  $u^r$ .

We have such a problem in decoding codes from Klein curves (Sakata 1995).

In the following subsections we present a series of extended BMS algorithms for these problems, where the validity of these algorithms can be proven similarly to the

original BMS algorithm. The theorems about complete class, uniqueness condition, and relevant Gröbner bases also can be given. All of them can be applied to fast decoding of certain algebraic codes (see Sakata 2009).

## 4.1 Multiarray BMS Algorithm

In this subsection we present the extended BMS algorithm (Sakata 1989) for solving the multiarray BMS problem (5), which is a multidimensional extension of the Feng-Tzeng algorithms (Feng and Tzeng 1989, 1991). As above-mentioned, given a finite set  $U = \{u^{(i)} \mid 1 \leq i \leq \mu\}$  of (finite or infinite)  $n$ -D arrays over  $\mathbb{F}$ , we want to find a minimal polynomial set of  $U$ . To discuss this problem, we introduce some additional notation.

For a given pair  $(j, r) \in [1, \mu] \times \mathbf{N}^n$ , where  $[1, \mu] := \{1, \dots, \mu\} \subset \mathbf{N}$ , let

$$\begin{aligned}\Sigma^{(j,r)} &:= \left( \bigcup_{i < j} \{(i, a) \in [1, \mu] \times \mathbf{N}^n \mid a \in \Sigma^{r \oplus 1}\} \right) \\ &\cup \left( \bigcup_{i \geq j} \{(i, a) \in [1, \mu] \times \mathbf{N}^n \mid a \in \Sigma^r\} \right)\end{aligned}$$

and consider the corresponding set of partial arrays defined over  $\Sigma^{(j,r)}$

$$U^{(j,r)} := \left( \bigcup_{i < j} \{u^{(i,r \oplus 1)} := (u_a^{(i)}), a \leq r\} \right) \cup \left( \bigcup_{i \geq j} \{u^{(i,r)} := (u_a^{(i)}), a < r\} \right)$$

If a polynomial  $f = \sum_{a \in \text{supp}(f)} c(f, a)X^a$  satisfies

$$f[u^{(i)}]_b := \sum_{a \in \text{supp}(f)} c(f, a)u_{a+b-d}^{(i)} = 0 \quad (9)$$

for any  $(i, b) \in \Sigma^{(j,r)}$  s.t.  $d = \text{le}(f) \leq_P b$ , we say that  $f$  is valid (w.r.t.  $U$ ) before  $(j, r)$ . As in case of a single array, i.e.  $\mu = 1$ , we can define a minimal polynomial set  $F(j, r)$  (or simply  $F$ )  $\subset \mathcal{P}$  of  $U^{(j,r)}$  for a pair  $(j, r) \in [1, \mu] \times \mathbf{N}^n$  associated with a finite set of points  $D(j, r) := \{\text{le}(f) \mid f \in F(j, r)\} \subset \mathbf{N}^n$  s.t. there is a single element  $f \in F(j, r)$  with  $\text{le}(f) = d$  and  $\text{lc}(f) = 1$  for each  $d \in D(j, r)$  and there exists no polynomial  $h$  with  $\text{le}(h) \in \Delta(j, r)$  which is valid (w.r.t.  $U$ ) before  $(j, r)$ , where we have a complementary pair of subsets  $\subset \mathbf{N}^n$ :

$$\Sigma(j, r) := \bigcup_{d \in D(j, r)} \Sigma_d, \quad \Delta(j, r) := \mathbf{N}^n \setminus \Sigma(j, r)$$

We call  $\Delta(j, r)$  the *delta-set* of  $F(j, r)$ . These subsets  $D(j, r)$ ,  $\Sigma(j, r)$  and  $\Delta(j, r)$  are unique for the given set  $U^{(j,r)}$  of partial arrays, but a minimal polynomial set  $F(j, r)$  is not necessarily unique for  $U^{(j,r)}$ . In view of the definition

of minimal polynomial set  $F(j, r)$ ,  $\Delta(j, r) \subseteq \Delta(j, r \oplus 1)$ ,  $\Delta(j, r) \subseteq \Delta(j + 1, r)$ ,  $1 \leq j < \mu$  and  $\Delta(\mu, r) \subseteq \Delta(1, r \oplus 1)$ . Similar notations can be used for any set of infinite arrays  $U$ , e.g. a minimal polynomial set  $F(\subset \mathcal{P})$  of  $U$ , the delta-set  $\Delta(\subset \mathbb{N}^n)$  of  $U$ , etc. if they exist. In the present problem we do not have a single auxiliary polynomial set of  $U$ , which is associated directly to the minimal polynomial set  $F$  of  $U$  and its delta-set  $\Delta$ , but  $\mu$  distinct auxiliary polynomial sets  $G^{(i)} = G^{(i)}(j, r)$  with  $C^{(i)} = C^{(i)}(j, r) = \{\text{span}(g) \mid g \in G^{(i)}\} \subset \mathbb{N}^n$  and  $\Delta^{(i)} = \Delta^{(i)}(j, r) = \bigcup_{c \in C^{(i)}} \Gamma_c$ ,  $1 \leq i \leq \mu$  s.t.  $\#\Delta(j, r) = \sum_{1 \leq i \leq \mu} \#\Delta^{(i)}$ . Below we show the multiarray BMS algorithm, which is almost the same as the original BMS algorithm (Algorithm 2) except for appearance of the additional loop w.r.t.  $j$  and  $C^{(i)}$  (and  $G^{(i)}$ )  $1 \leq i \leq \mu$  instead of  $C$ .

**Algorithm 3** (Multiarray BMS algorithm) Finding a minimal polynomial set of  $U^{(\mu, r)}$  for  $U = \{u^{(i)} \mid 1 \leq i \leq \mu\}$  of finite  $n$ -D arrays over  $\mathbb{F}$  (Sakata 1989);

Step 1 (initialization):  $j := 1$ ;  $b := 0$ ;  $F := \{1\}$ ;  $D := \{0\}$ ;  $\Sigma := \mathbb{N}^n$ ;  
 $G^{(i)} := \emptyset$ ,  $1 \leq i \leq \mu$ ;  $C^{(i)} := \emptyset$ ,  $1 \leq i \leq \mu$ ;

Step 2 (discrepancy): for each  $f \in F$   $d_f := f[u]_b$ ;

$$F_{\text{fail}} := \{f \in F \mid d_f \neq 0\};$$

$$F_{\text{fall}} := \{f \in F_{\text{fail}} \mid \exists c \in C^{(j)} \text{ s.t. } d \geq_P b - c\};$$

$$D_{\text{fall}} := \{d \in D \mid f \in F_{\text{fall}}\}; \hat{D} := \{\text{minimal } \hat{d} \in \Sigma \setminus \Gamma_b\};$$

$$D'' := \{\max(d, b - c) \mid d \in D_{\text{fall}}, c \in C^{(j)}\};$$

$$D' := \{\text{minimal } d' \in D''\};$$

Step 3 (updating): (1) for each  $f \in F_{\text{fail}} \setminus F_{\text{fall}}$

$$\begin{aligned} &\text{begin } h := f - d_f X^{d-(b-c)} g \text{ (for } g \in G^{(j)} \text{ s.t. } d \geq_P b - c\text{);} \\ &F' := F \cup \{h\} \text{ end;} \end{aligned}$$

for each  $(f, g) \in F_{\text{fall}} \times G^{(j)}$  s.t.  $d' := \max(d, b - c) \in D'$

$$\begin{aligned} &\text{begin } h := X^{\hat{d}-d-d'} f - d_f g; F' := F \cup \{h\} \text{ end;} \\ &\text{for each } \hat{d} \in \hat{D} \end{aligned}$$

if  $\exists d' \in D'$  s.t.  $\hat{d} \geq_P d'$  then for  $f \in F_{\text{fall}}$  s.t.  $d \leq_P \hat{d}$

$$\begin{aligned} &\text{begin } h := X^{\hat{d}-d} f; F' := F \cup \{h\} \text{ end;} \\ &(2) F := F' \setminus F_{\text{fail}}; G'' := \{g \in G^{(j)} \mid \exists f \in F_{\text{fall}} \text{ s.t. } c <_P b - d\}; \\ &G^{(j)} := (G^{(j)} \cup \{\frac{1}{d_f} f \mid f \in F_{\text{fall}}\}) \setminus G''; \end{aligned}$$

$$D := \{\text{le}(f) \mid f \in F' \setminus F_{\text{fall}}\}; \Sigma := \bigcup_{d \in D} \Sigma_d;$$

$$C^{(j)} := (C^{(j)} \cup \{b - d \mid \exists f \in F_{\text{fall}} \text{ s.t. } b - d >_P c\})$$

$$\setminus \{c \in C^{(j)} \mid \exists f \in F_{\text{fall}} \text{ s.t. } b - d >_P c\};$$

Step 4 (termination):  $j := j + 1$ : if  $j \leq \mu$  then go to Step 2

$$\text{else begin } j := 1; b := b \oplus 1;$$

$$\text{if } b < r \text{ then go to Step 2 else stop.}$$

## 4.2 Vectorial BMS Algorithm

In this subsection we present the vectorial BMS algorithm (Sakata 1991) for solving the vectorial BMS problem (6), for which the special 1D case of the vectorial BMS problem had not been treated and the vectorial BM algorithm had not given before.

As above-mentioned, we are given an array vector  $\mathbf{u} \in \mathcal{A}^m$ ,  $\mathbf{u} = (u^{(1)}, \dots, u^{(m)})$ , whose components are  $n$ -D arrays  $u^{(j)}$  over  $\mathbb{F}$ ,  $1 \leq j \leq m$ , defined over  $\Sigma^r \subset \mathbb{N}^n$  (w.r.t. a certain term ordering  $<$ ) for a fixed point  $r \in \mathbb{N}^n$ ;

The following is the vectorial BMS algorithm which finds  $m$  minimal polynomial vector sets  $\mathbf{F}^{(i)}(b)$ ,  $1 \leq i \leq m$  of the partial array vector  $\mathbf{u}^b$  as well as an auxiliary polynomial vector set  $\mathbf{G}(b)$  at each point  $b \in \Sigma^r$  iteratively w.r.t. the term ordering  $<$ , where all the elements  $\mathbf{f}$  of  $\mathbf{F}^{(i)}(b)$  have  $\text{lp}(\mathbf{f}) = i$ ,  $1 \leq i \leq m$ .  $\mathbf{F}^{(i)}(b)$ ,  $1 \leq i \leq m$  and  $\mathbf{G}(b)$  are denoted simply as  $\mathbf{F}^{(i)}$ ,  $1 \leq i \leq m$  and  $\mathbf{G}$ , respectively. During the execution of the algorithm, we have  $m$  delta-set  $\Delta^{(i)} = \Sigma \setminus \Sigma^{(i)}$ ,  $1 \leq i \leq m$  associated to  $\mathbf{F}^{(i)}$  with  $D^{(i)} = \{d = \text{le}(\mathbf{f}) \mid \mathbf{f} \in \mathbf{F}^{(i)}\}$  s.t.  $\Sigma^{(i)} = \bigcup_{d \in D^{(i)}} \Sigma_d$ ,  $1 \leq i \leq m$ , and an auxiliary polynomial vector set  $\mathbf{G} \in \mathcal{P}^m$ , to which a subset  $C = \{c = \text{span}(\mathbf{g}) \mid \mathbf{g} \in \mathbf{G}\} \subset \mathbb{N}^n$  is associated. Every initial  $\mathbf{F}^{(i)}$  is composed of a singleton  $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0) \in \mathcal{P}^m$ , i.e. the  $i$ -th unit vector,  $1 \leq i \leq m$ .

**Algorithm 4** (Vectorial BMS algorithm) Finding minimal polynomial vector sets of a finite  $n$ -D array vector  $\mathbf{u}^b \in \mathcal{A}^m$  over  $\mathbb{F}$  (Sakata 1991);

Step 1 (initialization):  $j := 1; b := 0; \mathbf{F}^{(i)} := \{\mathbf{e}_i\}, 1 \leq i \leq m; \mathbf{G} := \emptyset;$   
 $D^{(i)} := \{0\} (\subset \mathbb{N}^n), 1 \leq i \leq m; \Sigma^{(i)} := \mathbb{N}^n, 1 \leq i \leq m; C := \emptyset;$

Step 2 (discrepancy): for each  $\mathbf{f} \in \mathbf{F}^{(j)}$   $d_f := \mathbf{f}[\mathbf{u}]_b$ ;  
 $\mathbf{F}_{\text{fail}}^{(j)} := \{\mathbf{f} \in \mathbf{F}^{(j)} \mid d_f \neq 0\};$   
 $\mathbf{F}_{\text{fall}}^{(j)} := \{\mathbf{f} \in \mathbf{F}_{\text{fail}}^{(j)} \mid \exists c \in C \text{ s.t. } d \geq_P b - c\};$   
 $D_{\text{fall}}^{(j)} := \{d = \text{le}(\mathbf{f}) \in D^{(j)} \mid \mathbf{f} \in \mathbf{F}_{\text{fall}}^{(j)}\}; \hat{D} := \{\text{minimal } \hat{d} \in \Sigma \setminus \Gamma_b\};$   
 $D''^{(j)} := \{\max(d, b - c) \mid d \in D_{\text{fall}}^{(j)}, c \in C\};$   
 $D'^{(j)} := \{\text{minimal } d' \in D''^{(j)}\};$

Step 3 (updating): (1) for each  $\mathbf{f} \in \mathbf{F}_{\text{fail}}^{(j)} \setminus \mathbf{F}_{\text{fall}}^{(j)}$   
begin  $\mathbf{h} := \mathbf{f} - d_f X^{d-(b-c)} \mathbf{g}$  (for  $\mathbf{g} \in \mathbf{G}$  s.t.  $d \geq_P b - c$ );  
 $\mathbf{F}'^{(j)} := \mathbf{F}^{(j)} \cup \{\mathbf{h}\}$  end;  
for each  $(\mathbf{f}, \mathbf{g}) \in \mathbf{F}_{\text{fall}}^{(j)} \times \mathbf{G}$  s.t.  $d' := \max(d, b - c) \in D'^{(j)}$   
begin  $\mathbf{h} := X^{d'-d} \mathbf{f} - d_f \mathbf{g}$ ;  $\mathbf{F}'^{(j)} := \mathbf{F}^{(j)} \cup \{\mathbf{h}\}$  end;  
for each  $\hat{d} \in \hat{D}$  if  $\exists d' \in D'^{(j)}$  s.t.  $\hat{d} \geq_P d'$  then  
for  $\mathbf{f} \in \mathbf{F}_{\text{fall}}^{(j)}$  s.t.  $d \leq_P \hat{d}$   
begin  $\mathbf{h} := X^{\hat{d}-d} \mathbf{f}$ ;  $\mathbf{F}'^{(j)} := \mathbf{F}^{(j)} \cup \{\mathbf{h}\}$  end;  
(2)  $\mathbf{F}^{(j)} := \mathbf{F}'^{(j)} \setminus \mathbf{F}_{\text{fail}}^{(j)}; G'' := \{\mathbf{g} \in \mathbf{G} \mid \exists \mathbf{f} \in \mathbf{F}_{\text{fall}}^{(j)} \text{ s.t. } c <_P b - d\};$   
 $\mathbf{G} := (\mathbf{G} \cup \{\frac{1}{d_f} \mathbf{f} \mid \mathbf{f} \in \mathbf{F}_{\text{fall}}^{(j)}\}) \setminus \mathbf{G}'';$   
 $D^{(j)} := \{\text{le}(\mathbf{f}) \mid \mathbf{f} \in \mathbf{F}'^{(j)} \setminus \mathbf{F}_{\text{fail}}^{(j)}\}; \Sigma^{(j)} := \bigcup_{d \in D^{(j)}} \Sigma_d;$   
 $C := (C \cup \{b - d \mid \exists \mathbf{f} \in \mathbf{F}_{\text{fall}}^{(j)} \text{ s.t. } b - d >_P c\})$   
 $\setminus \{c \in C \mid \exists \mathbf{f} \in \mathbf{F}_{\text{fall}}^{(j)} \text{ s.t. } b - d >_P c\};$

Step 4 (termination):  $j := j + 1$ : if  $j \leq m$  then go to Step 2  
else begin  $j := 1; b := b \oplus 1$ ;  
if  $b < r$  then go to Step 2 else stop.

### 4.3 Non-Homogeneous BMS Algorithm

This is a two-path algorithm, the first path of which is just the (homogeneous) BMS algorithm for the given array  $u^{2r}$ . As a byproduct of the BMS algorithm, we get a series of minimal auxiliary polynomial sets  $\text{mAux}(u; c)$  for each  $c \in \Sigma^r$  based on the following two lemmas.

**Lemma 3** *Let  $h \in \text{Val}(u^b)$  and  $h[u]_b \neq 0$  (thus,  $\text{span}(h) = b - \text{le}(h)$ ). If  $\text{span}(h) = c \notin \Delta(u^b)$ , then  $h \in \text{mAux}(u; c)$ .*

**Lemma 4** *Let  $\Delta(u^b) = \Delta(u^{b \oplus 1}) = \dots = \Delta(u^{b \oplus l}) \subset \Delta(u^{b \oplus (l+1)})$  and  $h \in \text{mVal}\Delta(u^{b \oplus l})$ ,  $h[u]_{b \oplus l} \neq 0$ , where  $b \oplus (l+1) := (b \oplus l) \oplus 1$ ,  $1 \leq l (\in \mathbb{Z}_0)$ . Then,  $h \in \text{mAux}(u; c)$  for  $c := \text{span}(h)$  ( $= b \oplus l - \deg(h)$ ), and furthermore,  $X^a h \in \text{mAux}(u; c - a)$  for  $0 \leq a <_P c - c'$ , provided that there exists  $h' \in \text{mVal}(u^{b \oplus l})$  with  $\deg(h') = c' \leq_P c$ .*

First, by applying the BMS algorithm, we get a series of auxiliary polynomials  $g^c$  for several intermittent points  $c = c_1, c_2, \dots, c_\lambda$  s.t.  $0 \leq c_1 < c_2 < \dots < c_\lambda < r$  and  $c_i = \text{span}(g^{(c_i)})$ ,  $1 \leq i \leq \lambda$ , where  $\lambda$  is an integer determined by execution of BM algorithm. Furthermore, by Lemmas 3, 4, we can have a certain delta set  $\bar{\Delta}$  ( $= \bigcup_{1 \leq i \leq \lambda} \Gamma_{c_i}$ ) and an auxiliary polynomial  $h^{(c)}$  for each point  $c \in \bar{\Delta}$  s.t.  $\text{span}(h^{(c)}) = c$ ,  $c \in \bar{\Delta}$ . The following lemma leads us to have a fast algorithm of solving the nonhomogeneous BMS problem.

**Lemma 5** *Let  $h \in \text{mAux}(u; c)$ ,  $f \in \text{mVal}(u; v^c)$  and  $\text{le}(f) < \text{le}(h)$ . If  $f \notin \text{Val}(u; v^{c \oplus 1})$ , then there is no polynomial  $f' \in \text{Val}(u; v^{c \oplus 1})$  s.t.  $\text{le}(f') < \text{le}(h)$ .*

**Algorithm 5** (Nonhomogeneous BMS algorithm) Finding  $f \in \text{mVal}(u; v^r)$  for  $u = (u_a)$ ,  $a \in \Sigma^{2r}$  and  $v = (v_a)$ ,  $a \in \Sigma^r$  (Sakata 2003);

Step 1:  $b := 0$ ;  $f := 1$ ;  $d := 0$ ;

Step 2: If  $f \langle u \rangle_b \neq v_b$  then

begin  $h := h^{(b)}$ ;  $f := f + \frac{1}{d_b}(v_b - f \langle u \rangle_b)h$ ;  $d := \text{le}(h)$  end;

Step 3:  $b := b \oplus 1$ ; if  $b < r$  then go to Step 2 else stop.

If  $b \notin \bar{\Delta}$  in Step 2, then Algorithm 5 halts, which implies that (7) has no solution.

### 4.4 Submodule BMS Algorithm

To solve this problem, we can have a modification (Sakata 2007) of BMS algorithm which is obtained by modifying Step 1 (initialization) as follows:

Step 1 (initialization):  $b := \min_T \bar{\Delta}$ ;  $F := \{X^d \mid d \in \bar{\Delta}\}$ ;  $G := \emptyset$ ;  $C := \emptyset$ ;

The validity of the algorithm can be proven similarly to the original BMS algorithm. Particularly, it holds during the whole iterations of the algorithm that for  $\Delta(F) := \bar{\Sigma} \setminus \Sigma(F)$ , where  $\Sigma(F) = \Sigma(b)$ ,  $\Delta(F) = \Delta(b)$  ( $\subset \bar{\Sigma}$ ) for a minimal polynomial set  $F$  and an auxiliary polynomial set  $G$  of  $u^b$ , and  $\Delta(G) := \{a \in \mathbb{N}^n \mid a \leq_P \text{span}(g), g \in G\}$  ( $\subset \mathbb{N}^n$ ), it holds that  $\#\Delta(F) = \#\Delta(G)$ , although  $\Delta(F) \neq \Delta(G)$ .

## 4.5 Semigroup BMS Algorithm

In this case we also have a specific term ordering over  $\bar{\Sigma}$ , and we have a version (Sakata 1995) of the BMS algorithm, which is obtained by replacing every partial ordering  $\leq_P$  by  $\leq_P$  in the descriptions of the original BMS algorithm.

## 5 Conclusion

First we have discussed that the BMS algorithm (Sakata 1988, 1990) is related to Gröbner basis via multidimensional arrays and multidimensional linear recurrences satisfied by them, and that it can solve just the inverse problem of that of the Buchberger algorithm. Second, we have presented the essence of the BMS algorithm which outputs a minimal polynomial set of a given finite  $n$ -D array. Then, we have given theorems about the complete class of minimal polynomial sets, uniqueness condition and a Gröbner basis of the ideal  $\mathbf{I}(u)$  defined by an infinite array  $u$ , and discussed its computational complexity. Furthermore, we presented various extensions of the original BMS problem and their algorithms (Sakata 1989, 1991, 1995, 2003, 2007) which solve these problems. Those extended algorithms are useful for decoding several algebraic codes efficiently (Sakata 2009).

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

## Appendix A: Computation of BMS Algorithm

### Example of Computation

In Table 3 is shown a result of computations by BMS algorithm applied to the 2-D array shown in Table 2, where we take the graded reverse lexicographic ordering as the term ordering  $<$ . The symbol  $\star$  implies an updated polynomial. We can make sure by the Buchberger criterion that the minimal polynomial set obtained at the final iteration is a Gröbner basis. (Remark: A minimal polynomial set is not necessarily a Gröbner basis.)

**Table 2** 2-D array over  $\mathbf{F}_2$ :  
 $u = (u_{ij})$

$i \setminus j$	0	1	2	3	4
0	0	1	0	0	1
1	1	1	1	1	
2	1	0	0		
3	1	0			
4	0	*			
5	1				

**Table 3** Computations by  
BMS algorithm

$b$	$F$	$D$	$G$	$C$
(0, 0)	1	(0, 0)	–	–
(1, 0)	.	.	.	.
(0, 1)	$\star x^2$	(2, 0)	$\star 1$	(1, 0)
	$y$	(0, 1)		
(2, 0)	$x^2$	(2, 0)	1	(1, 0)
	$\star y + x$	(0, 1)		
(1, 1)	$\star x^2 + x$	(2, 0)	1	(1, 0)
	$y + x$	(0, 1)		
(0, 2)	.	.	.	.
(3, 0)	$x^2 + x$	(2, 0)	1	(1, 0)
	$\star xy + x^2$	(1, 1)	$\star y + x$	(0, 1)
	$\star y^2 + xy + x$	(0, 2)		
(2, 1)	.	.	.	.
(1, 2)	$x^2 + y$	(2, 0)	1	(1, 0)
	$\star xy + x^2 + 1$	(1, 1)	$y + x$	(0, 1)
	$y^2 + xy + x$	(0, 2)		
(0, 3)	.	.	.	.
(4, 0)	.	.	.	.
(3, 1)	.	.	.	.
(2, 2)	$\star x^3 + xy + y + x$	(3, 0)	$\star xy + x^2 + 1$	(2, 0)
	$\star x^2y + x^2 + x + 1$	(2, 1)	$\star x^2 + y$	(1, 1)
	$y^2 + xy + x$	(0, 2)	$y + x$	(0, 1)
(1, 3)	.	.	.	.
(0, 4)	$x^3 + xy + y + x$	(3, 0)	$xy + x^2 + 1$	(2, 0)
	$x^2y + x^2 + x + 1$	(2, 1)	$x^2 + y$	(1, 1)
	$\star y^2 + x^2 + x + 1$	(0, 2)	$y + x$	(0, 1)
(5, 0)	.	.	.	.
(4, 1)	*			

## References

- E. R. Berlekamp, *Algebraic coding theory*, McGraw–Hill, New York, 1968.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- G. L. Feng and K. K. Tzeng, *A generalized Euclidean algorithm for multisequence shift-register synthesis*, IEEE Trans. on Inf. Th. **35** (1989), no. 3, 584–594.
- G. L. Feng and K. K. Tzeng, *Decoding cyclic and BCH codes up to actual minimum distance using nonrecurrent syndrome dependence relations*, IEEE Trans. on Inf. Th. **37** (1991), no. 6, 1716–1723.
- T. Ikai, H. Kosako and Y. Kojima, *Basic theory of two-dimensional cyclic codes – generator polynomials and the position of check symbols*, IEICE Trans. Fundamentals **J59-A** (1976), 311–318.
- J. L. Massey, *Shift-register synthesis and BCH decoding*, IEEE Trans. on Inf. Th. **15** (1969), 122–127.
- T. Mora, *The FGLM problem and Möller's algorithm on zero-dimensional ideals*, this volume, 2009a, pp. 27–45.
- T. Mora, *Gröbner technology*, this volume, 2009b, pp. 11–25.
- S. Sakata, *General theory of doubly periodic arrays over an arbitrary finite field and its applications*, IEEE Trans. on Inf. Th. **24** (1978), no. 6, 719–730.
- S. Sakata, *On determining the independent point set for doubly periodic arrays and encoding two-dimensional cyclic codes and their duals*, IEEE Trans. on Inf. Th. **27** (1981), no. 5, 556–565.
- S. Sakata, *Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array*, J. Symbolic Comput. **5** (1988), no. 3, 321–337.
- S. Sakata,  *$n$ -dimensional Berlekamp–Massey algorithm for multiple arrays and construction of multivariate polynomials with preassigned zeros*, LNCS, vol. **357**, 1989, Springer, Berlin, pp. 356–376.
- S. Sakata, *Extension of the Berlekamp–Massey algorithm to  $N$  dimensions*, Inform. and Comput. **84** (1990), no. 2, 207–239.
- S. Sakata, *Finding a minimal polynomial vector set of a vector of  $nD$  arrays*, LNCS, vol. **539**, Springer, Berlin, 1991, pp. 414–425.
- S. Sakata, *Shift register synthesis on convex cones and cylinders and fast decoding of general one-point AG codes*, Bull. Univ. Electro-Comm. **8** (1995), no. 2, 187–203.
- S. Sakata, *Efficient factorization methods for list decoding of code from curves*, Proc. of ISIT 2003, 2003, pp. 363–363.
- S. Sakata, *Fast decoding of two-point Hermitian codes*, preprint, 2007.
- S. Sakata, *The BMS algorithm and decoding of AG codes*, this volume, 2009, pp. 165–185.
- S. Sakata, H. E. Jensen and T. Höholdt, *Generalized Berlekamp–Massey decoding of algebraic-geometric codes up to half the Feng–Rao bound*, IEEE Trans. on Inf. Th. **41** (1995), no. 6, 1762–1768, part 1.
- Y. Sugiyama, *An algorithm for solving discrete-time Wiener–Hopf equations based upon Euclid's algorithm*, IEEE Trans. on Inf. Th. **32** (1986), no. 3, 394–409.

# The BMS Algorithm and Decoding of AG Codes

Shojiro Sakata

**Abstract** In this paper, we review various decoding methods of algebraic geometry (or algebraic-geometric) codes (Goppa in Soviet Math. Dokl. 24(1):170–172, 1981; Høholdt et al. in Handbook of coding theory, vols. I, II, North-Holland, Amsterdam, pp. 871–961, 1998; Geil in Algebraic geometry codes from order domains, this volume, pp. 121–141, 2009) mainly based on the Gröbner basis theory (Buchberger in Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal, Ph.D. thesis, Innsbruck, 1965; Aequationes Math. 4:374–383, 1970; Multidimensional systems theory, Reidel, Dordrecht, pp. 184–232, 1985; London Math. Soc. LNS 251:535–545, 1998; J. Symb. Comput. 41(3–4):475–511, 2006; Mora in Gröbner technology, this volume, pp. 11–25, 2009b) as well as the BMS algorithm (Sakata in J. Symbolic Comput. 5(3):321–337, 1988; Inform. and Comput. 84(2):207–239, 1990) and its variations (Sakata in  $n$ -dimensional Berlekamp–Massey algorithm for multiple arrays and construction of multivariate polynomials with preassigned zeros, LNCS, vol. 357, pp. 356–376, 1989; Finding a minimal polynomial vector set of a vector of  $n$ D arrays, LNCS, vol. 539, pp. 414–425, 1991), where the BMS algorithm itself is reviewed in another paper (Sakata in The BMS algorithm, this volume, pp. 143–163, 2009) in this issue. The main subjects are:

(1) Syndrome decoding of dual codes up to the designed distance (Saints and Heegard in IEEE Trans. Inform. Theory 41(6):1733–1751, 1995; Sakata et al. in Finite Fields Appl. 1(1):83–101, 1995b; IEEE Trans. on Inf. Th. 41(6):1672–1677, 1995c; IEEE Trans. on Inf. Th. 41(6):1762–1768, 1995a) by using the BMS algorithm. (There have been published several methods of decoding algebraic geometry codes, e.g. Kötter in On decoding of algebraic-geometric and cyclic codes, Ph.D. thesis, Linköping University, 1996; O’Sullivan in IEEE Trans. on Inf. Th. 41(6):1709–1719, 1995; Guerrini and Rimoldi in FGLM-like decoding: from Fitzpatrick’s approach to recent developments, this volume, pp. 197–218, 2009, which are described in some terminology rather from the perspective of algebraic geometry, but are in principle equivalent to the BMS decoding method. We omit their descriptions here.)

(2) List decoding of primal codes (Numakami et al. in IEICE Trans. Fundamentals J83:1309–1317, 2000; Sakata in LNCS, vol. 2227, pp. 172–181, 2001; Proc. of ISIT2003, pp. 363–363, 2003). (The original list decoding algorithms are given for RS codes by Sudan in J. of Complexity 13:180–193, 1997, and for algebraic geometry codes by Shokrollahi and Wassermann in IEEE Trans. on Inf. Th. 45(2):432–437,

---

S. Sakata

The University of Electro-Communications, Chofu-shi, Tokyo 182-8585, Japan  
e-mail: [sakata@ice.uec.ac.jp](mailto:sakata@ice.uec.ac.jp)

1999, and their improved versions by Guruswami and Sudan in IEEE Trans. on Inf. Th. 45(6):1757–1767, 1999.)

(3) Other relevant decoding algorithms of primal and dual codes (Augot in Proc. of ISIT2002, pp. 86–86, 2002; Justesen and Høholdt in A course in error-correcting codes, EMS Textbooks in Mathematics, EMS, 2004; Fujisawa and Sakata in Proc. of SITA2005, pp. 543–546, 2005; Sakata and Fujisawa in Proc. of SITA2006, pp. 93–96, 2006; Fujisawa et al. in Proc. of SITA2006, pp. 101–104, 2006).

In discussing list decoding and usual bounded-distance decoding of primal/dual codes we show that multi-variate interpolation problem is a key and that it can be solved by using the BMS algorithm efficiently. The computational complexities of our methods are less than the other decoding methods including the Feng–Rao (IEEE Trans. on Inf. Th. 39(1):37–45, 1993) algorithm simply based on Gaussian elimination. These reductions in computational complexity are based on the special structures or properties of the given input data (syndrome arrays, etc.) which originate in the definition of codes themselves and are used cleverly by the BMS algorithm. In Leonard (A tutorial on AG code decoding from a Gröbner basis perspective, this volume, pp. 187–196, 2009b), Guerrini and Rimoldi (FGLM-like decoding: from Fitzpatrick’s approach to recent developments, this volume, pp. 197–218, 2009) in this issue, several other efficient decoding methods of algebraic geometry codes from Gröbner basis perspectives are reviewed. Additionally, we mention a recent development of decoding algorithm based on higher-dimensional interpolation (Parvaresh and Vardy in Proc. of IEEE FOCS2005, IEEE Computer Society, pp. 285–294, 2005), which has error correction performance superior to the improved list decoding by Guruswami and Sudan. As a general method of multivariate interpolation the BMS algorithm is an alternative of the Buchberger–Möller (The construction of multivariate polynomials with preassigned zeros, LNCS, vol. 144, pp. 24–31, 1982), Mora (The FGLM problem and Möller’s algorithm on zero-dimensional ideals, this volume, pp. 27–45, 2009a) algorithm and the Marinani–Möller–Mora (AAECC 4:(2):103–145, 1993) algorithm, but any exact comparisons of computational complexities of these methods remain to be investigated.

## 1 Introduction

In this paper, we review various decoding methods of algebraic geometry (or algebraic-geometric) codes over finite fields, particularly one-point codes from algebraic curves mainly based on the BMS algorithm (Sakata 1988, 1990), which we review in another paper (Sakata 2009) in this issue, and we use almost the same terminology as *ibid*. These *algebraic geometry codes* are the most important class of error-correcting codes from both practical and theoretical viewpoints. They are a subclass of so-called *linear codes* which are defined as linear subspaces of the vector space  $\mathbb{F}_q^n = (\mathbb{F}_q)^n$  over a finite field  $\mathbb{F}_q$ . Since most of the basic concepts in Coding Theory are introduced in another paper (Augot et al. 2009) in this issue, we omit many of their detailed descriptions here and assume that the readers know terminologies such as  $(n, k, d)$ -code  $C$  ( $\subset \mathbb{F}_q^n$ ) over  $\mathbb{F}_q$ , codelength  $n$ , dimension  $k$ ,

minimum distance  $d$ , the number  $t = \lfloor \frac{d-1}{2} \rfloor$  of correctable errors, etc. Decoding, which is to recover or estimate the sent codeword  $\mathbf{c} \in C$  from the given received word  $\mathbf{r} \in \mathbb{F}_q^n$ , is a kind of algebraic computation procedure over the finite field  $\mathbb{F}_q$ , and it is given basically in the form of an algorithm. If the received word  $\mathbf{r}$  contains more errors than  $t$ , the decoding algorithm might output a wrong codeword which is different from the sent codeword. But, error events are probabilistic phenomena in practical applications, and more errors can occur with less probability, which usually is negligibly smaller. Therefore, in decoding, we have only to find candidate codewords which are as close to the received word  $\mathbf{r}$  as possible.

The algebraic geometry codes which we are going to discuss in this paper are defined based on a triplet  $(\mathcal{K}, \mathcal{L}, \mathcal{C})$ , where  $\mathcal{K}$  is the set of symbols carrying information with them and  $\mathcal{L}$  is the set of *locators* (or labels)  $P_j$  denoting the position or index  $j$  of each component symbol  $c_j (\in \mathbb{F}_q)$  of a codeword  $\mathbf{c} = (c_j)_{0 \leq j \leq n-1}$ . We call  $\mathcal{K}$  and  $\mathcal{L}$  the *information symbol set* and the *symbol locator set*, respectively. The set  $\mathcal{C}$  is a linear space of functions defined on a domain including  $\mathcal{L}$ , from which we have two kinds of codes as follows. First, we have a code  $C$  which is the subspace of  $(\mathbb{F}_q)^n$  composed of the vectors  $\text{ev}(f) := (f(P_0), \dots, f(P_{n-1})) \in \mathbb{F}_q^n$  corresponding to a function  $f \in \mathcal{C}$ . Second, we have another code which is the orthogonal complement (*null space*) of the subspace  $C$  in  $\mathbb{F}_q^n$

$$C^\perp := \{\mathbf{c} = (c_j) \in \mathbb{F}_q^n \mid \mathbf{c} \cdot \text{ev}(f) = 0\},$$

where  $\mathbf{c} \cdot \text{ev}(f) := \sum_{0 \leq j \leq n-1} c_j f(P_j) (\in \mathbb{F}_q)$  is the inner product of two vectors  $\mathbf{c}$  and  $\text{ev}(f) (\in \mathbb{F}_q^n)$ . Sometimes we call  $C$  and  $C^\perp$  *primal* and *dual* codes, respectively.<sup>1</sup>

For example, primal and dual Reed–Solomon codes  $C$  and  $C^\perp$ , which are nowadays one of the most practically used algebraic error-correcting codes, are defined<sup>2</sup> by taking  $\mathcal{K} := \mathbb{F}_q$ ,  $n := q - 1$ ,  $\mathcal{L} := \{P_j(:= \alpha^j) \mid 0 \leq j \leq n - 1 (= q - 2)\} (= \mathbb{F}_q \setminus \{0\})$ , and  $\mathcal{C} := \{f \in \mathbb{F}_q[x] \mid \deg(f) \leq h - 1\}$  for a certain integer  $h$  s.t.  $0 < h < n$ . Their dimensions and minimum distances are

$$k(C) = h, \quad k(C^\perp) = n - h; \quad d(C) = n - h + 1, \quad d(C^\perp) = h + 1.$$

RS codes are among the broader class of *one-point* codes from algebraic curves which contains codes having better performance and greater potentialities in the near future. One-point codes from an algebraic curve  $\mathcal{X}$  over a finite field  $\mathbb{F}_q$  are

<sup>1</sup>About the definition of these codes, see also another paper (Leonard 2009a) in this issue, where  $C$  and  $C^\perp$  are called *functionally encoded* and *functionally decoded codes*, respectively. Furthermore, about codes from order domains, which are a generalization of these codes and can be decoded by our methods, see Geil (2009).

<sup>2</sup>This definition of the dual RS code  $C^\perp$  is equivalent to the conventional definition  $C^\perp := \{c(x) = a(x)g(x) \mid a(x) \in \mathbb{F}_q[x], \deg(a) \leq n - h - 1\}$  s.t. each codeword  $\mathbf{c} = (c_j) \in C^\perp$  is represented as a polynomial  $c(x) = \sum_{0 \leq j \leq n-1} c_j x^j$ , where  $g(x) := \prod_{0 \leq i \leq h-1} (x - \alpha^i)$  is the generator polynomial of the code.

defined by taking  $\mathcal{K} := \mathbb{F}_q$ ,  $\mathcal{L} := \{P_j \mid 0 \leq j \leq n-1\}$ , which is a set of  $\mathbb{F}_q$ -rational points on the curve  $\mathcal{X}$ , and  $\mathcal{C} := L(mP_\infty)$ , which is the set of algebraic functions on the curve  $\mathcal{X}$  having a single pole at the infinity point  $P_\infty$  with *pole order* less than or equal to  $m$ , where  $m$  is a given integer. Similarly, we have primal and dual codes  $C$  and  $C^\perp$ . As a special case, if we take as  $\mathcal{X}$  the projective line over  $\mathbb{F}_q$  containing the infinity point  $P_\infty$  as well, and let  $\mathcal{L}$  be the set of all affine points on  $\mathcal{X}$  or equivalently the finite field  $\mathbb{F}_q$ , then we have the *extended RS code* with length  $n = q$ . By deleting 0 from  $\mathcal{L}$ , we have the ordinary RS code of length  $n = q - 1$ .

Although we can take the defining curve  $\mathcal{X}$  in the projective space of any dimension  $N$ , we restrict to a plane curve  $\mathcal{X}$  (i.e.  $N = 2$ ) or particularly the Hermitian curve over  $\mathbb{F}_q$  as follows, where  $q = q_1^2$ .

$$\mathcal{X} : y^{q_1} - x^{q_1+1} + y = 0.$$

We take as  $\mathcal{L}$  all the  $\mathbb{F}_q$ -rational points on  $\mathcal{X}$  excluding the infinity point  $P_\infty$ , where we remember that the coordinate functions  $x$  and  $y$  have pole orders  $o(x) = q_1$  and  $o(y) = q_1 + 1$ , respectively at the single pole  $P_\infty$ . For  $a = (a_1, a_2) \in \mathbf{N}^2$ , we denote  $X^a := x^{a_1}y^{a_2}$ , which has pole order  $o(X^a) = q_1a_1 + (q_1 + 1)a_2$ . Letting  $\Pi := \{a = (a_1, a_2) \in \mathbf{N}^2 \mid 0 \leq a_2 \leq q_1 - 1\}$ ,  $\Pi(m) := \{a = (a_1, a_2) \in \Pi \mid o(X^a) = q_1a_1 + (q_1 + 1)a_2 \leq m\}$ , and  $\mathcal{C} = \langle X^a = x^{a_1}y^{a_2} \mid a = (a_1, a_2) \in \Pi(m) \rangle_{\mathbb{F}_q} \subset \mathbb{F}_q[\Pi]$  ( $:= \langle X^a \mid a \in \Pi \rangle_{\mathbb{F}_q}$ ), we can have the primal code  $C = C(m)$  and the dual code  $C^\perp = C^\perp(m)$  with length  $n := q_1^3$ , whose dimensions and minimum distances are as follows in case of  $2g - 1 \leq m < n$ , where  $g = \frac{q_1(q_1-1)}{2}$  is the genus of the curve  $\mathcal{X}$ :

$$\begin{aligned} k(C) &= m - g + 1, & d(C) &\geq n - m; \\ k(C^\perp) &= n - m + g - 1, & d(C^\perp) &\geq m - 2g + 2, \end{aligned}$$

where  $d_G := n - m$  and  $d_G^\perp := m - 2g + 2$  are called *Goppa bounds* of the primal code  $C$  and the dual code  $C^\perp$ , respectively. Actually, if  $m + m' = q_1^3 + q_1^2 - q_1 - 2$ , the primal Hermitian code  $C(m)$  and the dual Hermitian code  $C^\perp(m')$  are equivalent (Stichtenoth 1988).

## 2 Syndrome Decoding of Dual Codes

First we show that decoding of a dual RS code  $C^\perp$  with minimum distance  $d = h + 1$  is reduced to the problem of finding a polynomial in  $\mathbb{F}_q[x]$  which is *valid* for a certain one-dimensional (1-D) array derived from the received word. Let  $\mathbf{c} = (c_j)_{0 \leq j \leq n-1} \in C^\perp$  and  $\mathbf{e} = (e_j)_{0 \leq j \leq n-1} \in \mathbb{F}_q^n$  be a *sent codeword* and an *error vector*, respectively. Then, the received word is  $\mathbf{r} = \mathbf{c} + \mathbf{e} = (r_j)_{0 \leq j \leq n-1} \in \mathbb{F}_q^n$ , where  $r_j = c_j + e_j$ ,  $0 \leq j \leq n - 1$ . We assume that the number of errors, or in other words the size of the set  $\mathcal{E} := \{P_j \mid e_j \neq 0\} (\subset \mathcal{L})$  of *error locators*, is  $t' := \#\mathcal{E} \leq t$ , where  $t (= \lfloor \frac{h}{2} \rfloor)$  is the number of correctable errors. The receiver gets the received word  $\mathbf{r} = (r_j)$ , but he has no knowledge of both  $\mathbf{c}$  and  $\mathbf{e}$ . How can he find either  $\mathbf{c}$  or

**e** from **r**? Since no error, i.e. the case of  $\mathbf{e} = 0$  is the most likely in actual channels, he begins with checking whether the received word **r** contains any error or not. For a dual RS code, it is very easy and he has only to check for some  $f \in \mathcal{C}$  whether the inner product  $\mathbf{r} \cdot \text{ev}(f) = 0$  or not. More precisely, he calculates the syndromes  $s_i := \mathbf{r} \cdot \text{ev}(x^i)$  corresponding to the basis functions  $x^i$ ,  $0 \leq i \leq h - 1$  of the function space  $\mathcal{C}$ , and obtains the array  $s = (s_i)_{0 \leq i \leq h-1}$ . If  $s = 0$ , then he most probably can suppose no error so that he does not need to go further. But, if  $s \neq 0$ , then he enters the procedure of decoding. A basic decoding method consists of two stages, finding the error locators, i.e. the unknown  $j_i$  or  $\alpha^{j_i}$ ,  $1 \leq i \leq t'$  for  $\mathcal{E} = \{\alpha^{j_i} \mid 1 \leq i \leq t'\}$ , and calculating the error values  $e_{j_i}$ ,  $1 \leq i \leq t'$ . Provided the error locators  $\mathcal{E}$  are found in the first stage, the second stage is easier and reduced to finding the unique solution  $e_{j_i}$ ,  $1 \leq i \leq t'$  of the linear system of equations:  $\sum_{1 \leq i \leq t'} e_{j_i} \alpha^{j_i j} = s_j$ ,  $0 \leq j \leq h - 1$ .

Now, our main concern is in the first stage. Assuming  $t' \leq t$  for  $\mathcal{E} = \{\alpha^{j_i} \mid 1 \leq i \leq t'\}$ , where  $t'$  and  $j_i$ ,  $1 \leq i \leq t'$  are unknown, we consider an infinite array  $u = (u_j)$  defined by  $u_j := \mathbf{e} \cdot \text{ev}(x^j) = \sum_{1 \leq i \leq t'} e_{j_i} \alpha^{j_i j}$ ,  $j \in \mathbb{N}$  instead of  $s$ , and further the ideal  $\mathbf{I} = \mathbf{I}(u) := \{f \in \mathbb{F}_q[x] \mid f \circ u = 0\}$ , which is called the *characteristic ideal* of  $u$ , as well as the zero variety  $V(\mathbf{I}) := \{\gamma \in \mathbb{F}_q \mid f(\gamma) = 0, \forall f \in \mathbf{I}\}$  defined by it, where for  $f = f(x) = \sum_{0 \leq l \leq d} f_l x^l$ ,  $v = f \circ u := (v_j)_{j \in \mathbb{N}}$  is the array defined by  $v_j := \sum_{0 \leq l \leq d} f_l u_{l+j}$ ,  $j \in \mathbb{N}$  (see Sakata 2009). Actually, we have

**Lemma 1**  $\mathcal{E} = V(\mathbf{I})$ .

*Proof* For  $f = f(x) = \sum_{0 \leq l \leq d} f_l x^l$ , we have

$$\begin{aligned} f(\alpha^{j_i}) = 0, \quad 1 \leq i \leq t' &\Leftrightarrow \sum_{0 \leq l \leq d} f_l \alpha^{j_i l} = 0, \quad 1 \leq i \leq t' \\ &\Leftrightarrow \sum_{1 \leq i \leq t'} \left( \sum_{0 \leq l \leq d} f_l \alpha^{j_i l} \right) e_{j_i} \alpha^{j_i j} = 0, \quad \forall j \in \mathbb{N} \\ &\Leftrightarrow \sum_{0 \leq l \leq d} f_l \sum_{1 \leq i \leq t'} e_{j_i} \alpha^{j_i(l+j)} = 0, \quad \forall j \in \mathbb{N}, \end{aligned}$$

where the last identity is equivalent to  $\sum_{0 \leq l \leq d} f_l u_{l+j} = 0$ ,  $\forall j \in \mathbb{N}$ , i.e.  $f \circ u = 0$ . By the way, the equivalence between the second and third identities comes from the fact that  $t'$  arrays  $u^{(i)} := (u_j^{(i)})$ ,  $1 \leq i \leq t'$  which are defined by  $u_j^{(i)} := \alpha^{j_i j}$  are linearly independent of each other.  $\square$

Since we have that  $s_i = \mathbf{r} \cdot \text{ev}(x^i) = (\mathbf{c} + \mathbf{e}) \cdot \text{ev}(x^i) = \mathbf{e} \cdot \text{ev}(x^i)$ ,  $0 \leq i \leq h - 1$ , the subarray  $u^h := (u_j)_{0 \leq j \leq h-1}$  of the above infinite array  $u$  coincides with the syndrome array  $s = (s_j)_{0 \leq j \leq h-1}$ , although we cannot obtain the whole infinite array  $u$ . Particularly, the values  $u_j$ ,  $j \geq h$  sometimes are called *unknown syndromes*. However, if  $\deg(f) = t' \leq t$ , in view of  $h - 1 - t' \geq t' - 1$ , for  $1 \leq i \leq t'$ , we have  $t'$  finite arrays  $u_j^{(i)} := \alpha^{j_i j}$ ,  $0 \leq j \leq h - 1 - t'$ , which also are linearly independent of

each other. Consequently, we have for  $V(f) := \{\gamma \in \mathbb{F}_q \mid f(\gamma) = 0\}$ ,

$$\mathcal{E} = V(f) \Leftrightarrow \sum_{0 \leq l \leq t'} f_l u_{l+j} = 0, \quad 0 \leq j \leq h-1-t', \quad (1)$$

which implies that we can find the error locators  $\mathcal{E}$  as the roots of a polynomial  $f$  which is valid for the *known syndromes*  $u_i (= s_i)$ ,  $0 \leq i \leq h-1$  obtained from the received word  $\mathbf{r}$  and has the minimum degree, provided the actual number  $t'$  of errors contained in  $\mathbf{r}$  does not exceed the number  $t$  of correctable errors.

As we have seen, the problem of decoding dual RS codes is reduced to finding a valid polynomial for a certain finite (1-D) array. Naturally this fact can be extended to the problem of decoding more general codes including *codes from algebraic curves*. Particularly, in the multidimensional case, it also implies that we must find a Gröbner basis of the characteristic ideal of the array. Below we will show that the decoding of a dual Hermitian code  $C^\perp$  is reduced to the problem of finding a minimal polynomial set (in  $\mathbb{F}_q[x, y]$ ) of a certain 2-D array derived from a received word.

Let  $\mathbf{c} = (c_j) \in C^\perp$ ,  $\mathbf{e} = (e_j) \in \mathbb{F}_q^n$ ,  $\mathbf{r} = \mathbf{c} + \mathbf{e} = (v_j) \in \mathbb{F}_q^n$  be the sent codeword, the error vector, and the received word, respectively. We assume that the size of the error locators  $\mathcal{E} := \{P_j \mid e_j \neq 0\} = \{P_{l_i} \mid 1 \leq i \leq t'\} (\subset \mathcal{L})$  is  $t' := \#\mathcal{E} \leq t_G^\perp := \lfloor \frac{d_G^\perp - 1}{2} \rfloor$ . As each point of the curve can be represented as  $P_l = (\alpha_l, \beta_l) \in (\mathbb{F}_q)^2$ , the syndrome  $s = (s_a)$ , with  $a \in \Pi(m)$ , obtained by  $s_a := \mathbf{r} \cdot \text{ev}(X^a)$  from the received word  $\mathbf{r}$  is a finite subarray of the infinite 2-D array  $u = (u_a)$ ,  $a \in \mathbb{N}^2$ , defined by

$$u_a := \mathbf{e} \cdot \text{ev}(X^a) = \sum_{1 \leq i \leq t'} e_{l_i} \alpha_{l_i}^{a_1} \beta_{l_i}^{a_2}, \quad a = (a_1, a_2) \in \mathbb{N}^2,$$

which we call *error locator array*. About the *characteristic ideal (submodule)*  $\mathbf{I} = \mathbf{I}(u) := \{f \in \mathbb{F}_q[\Pi] \mid f \circ u = 0\}$  of a 2-D array  $u = (u_a)$ ,  $a \in \mathbb{N}^2$  and its zero variety  $V(\mathbf{I}) := \{P \in \mathcal{L} \mid f(P) = 0, \forall f \in \mathbf{I}\}$ , we have the following lemma similar to Lemma 1. Thus, we call  $\mathbf{I}$  also the *error locator ideal* (or *submodule*), and sometimes denote it as  $\mathbf{I}(\mathbf{e})$  (or  $\mathbf{M}(\mathbf{e})$ ).

**Lemma 2**  $\mathcal{E} = V(\mathbf{I})$ .

*Proof* For  $f = f(x, y) = f(X) = \sum_{a \in \text{supp}(f)} c(f, a) X^a \in \mathbb{F}_q[\Pi]$ , we have

$$\begin{aligned} f(\alpha_{l_i}, \beta_{l_i}) = 0, \quad 1 \leq i \leq t' &\Leftrightarrow \\ \sum_{a=(a_1,a_2) \in \text{supp}(f)} c(f, a) \alpha_{l_i}^{a_1} \beta_{l_i}^{a_2} = 0, \quad 1 \leq i \leq t' &\Leftrightarrow \\ \sum_{1 \leq i \leq t'} \left( \sum_{a \in \text{supp}(f)} c(f, a) \alpha_{l_i}^{a_1} \beta_{l_i}^{a_2} \right) e_{l_i} \alpha_{l_i}^{b_1} \beta_{l_i}^{b_2} &= 0, \quad (*) \Leftrightarrow \\ \sum_{a \in \text{supp}(f)} c(f, a) \sum_{1 \leq i \leq t'} e_{l_i} \alpha_{l_i}^{a_1+b_1} \beta_{l_i}^{a_2+b_2} &= 0, \quad (*) \end{aligned}$$

where  $(*)$  implies “ $\forall b = (b_1, b_2) \in \mathbf{N}^2$ ”. The last identity is equivalent to  $\sum_{a \in \text{supp}(f)} c(f, a)u_{a+b} = 0, \forall b \in \mathbf{N}^2$ , i.e.  $f \circ u = 0$ . The equivalence between the second and third identities comes from the fact that  $t'$  arrays  $u^{(l)} := (u_a^{(l)}), 1 \leq l \leq t'$  defined by  $u_a^{(l)} := \alpha_l^{a_1} \beta_l^{a_2}, a \in \mathbf{N}^2$ , are linearly independent from each other.  $\square$

In the above, for the ring  $\mathcal{P} := \mathbb{F}_q[x, y]$ , the function space  $\mathbb{F}_q[\Pi] := \langle X^a = x^{a_1}y^{a_2} \mid a = (a_1, a_2) \in \Pi \rangle_{\mathbb{F}_q}$  is viewed as a  $\mathcal{P}$ -submodule which coincides with the whole set  $\mathcal{P}$  (as a module) modulo the  $\mathcal{P}$ -submodule  $\mathbf{M}_{\mathcal{X}} := \langle y^{q_1} - x^{q_1+1} + y \rangle_{\mathcal{P}}$ . The known syndromes  $s_a = \mathbf{r} \cdot \text{ev}(X^a)$ ,  $a \in \Pi(m)$ , which are obtained from the received word, are identical with the subarray  $u_a, a \in \Pi(m)$ , but the part  $u_a, a \in \Pi \setminus \Pi(m)$  are unknown syndromes. On the other hand, among the functions defined on the curve, since  $X^a, a \in \mathbf{N}^2 \setminus \Pi$  are linearly dependent on  $\{X^b \mid b \in \Pi, o(X^b) \leq o(X^a)\}$ , the subarray  $u_a, a \in 2\Pi(m)$  also is known, where  $2\Pi(m) := \{a+b \mid a, b \in \Pi(m), o(X^{a+b}) \leq m\}$ . In the linear recurrence  $f \circ u = 0$ , i.e.

$$\sum_{a \in \text{supp}(f)} c(f, a)u_{a+b} = 0, b \in \Pi,$$

not only the components  $u_a, a \in \Pi(m)$  but also the components  $u_a, a \in 2\Pi(m) \setminus \Pi(m)$  are concerned. Therefore, all the components  $u_a, a \in 2\Pi(m)$  are necessary for decoding by using the BMS algorithm. Furthermore, treating only the known syndrome is not enough for decoding of this kind of codes up to half of the designed distance, which we will discuss below.

There have been several investigations on *designed distances* or *lower bounds for minimum distances* of codes from curves. We consider the Feng–Rao (1993) bound of dual Hermitian codes, which is equal to the so-called order bound (Høholdt et al. 1998; Geil 2009) as well as to the Goppa bound  $d_G^\perp$  in case of  $2g - 1 \leq m < n$  for these codes. Although the Feng–Rao decoding algorithm based on Gaussian elimination and majority logic can decode up to  $t_G^\perp = \lfloor \frac{d_G^\perp - 1}{2} \rfloor$  errors, it will turn out that the BMS algorithm with majority logic can do the same more efficiently (Sakata et al. 1995a). By using the BMS algorithm w.r.t. the term ordering corresponding to the pole order  $o(X^a)$  as mentioned in the next paragraph, we can determine the unknown syndromes based on majority logic in its unique (basically, similar to the Feng–Rao algorithm) fashion so that we can find a minimal polynomial set of the array  $u$  which is a Gröbner basis of the error locator ideal  $\mathbf{I}(\mathbf{e})$ .

Let  $\mathcal{O}$  be the set of pole orders  $o(f)$  of functions  $f$  on the algebraic curve  $\mathcal{X}$  over the closed extension (closure)  $\tilde{\mathbb{F}}_{q_1} := \bigcup_{i \geq 1} \mathbb{F}_{q_1^i}$  of  $\mathbb{F}_{q_1}$ , and  $\mathcal{O}(m) := \{l \in \mathcal{O} \mid l \leq m\}$ . Particularly, we denote the pole order  $o(X^a)$  of the coordinate function  $X^a$  simply as  $o(a)$ ,  $a \in \mathbf{N}^2$ , which determines the term ordering  $<$  together with a certain lexicographic ordering  $<_L$ . Then, via  $o(a), a \in \mathbf{N}^2$ ,  $\mathcal{O}$  and  $\mathcal{O}(m)$  one-to-one correspond to  $\Pi$  and  $\Pi(m)$ , respectively. For  $l \in \mathcal{O}$ ,

$$v(l) := \#\{(i, j) \in \mathcal{O}^2 \mid i + j = l\}$$

is introduced and the order bound of the code  $C^\perp(m)$  is defined as

$$d(m) := \min\{v(l) \mid l \geq m + 1\}.$$

On the other hand we sometimes have a couple of points  $r \in \Pi$  and  $r' = r \oplus 1$  (i.e. the next point after  $r$  w.r.t. the term ordering  $<$ )  $\in 2\Pi \setminus \Pi$  s.t.  $o(r) = o(r')$ , and thus,  $X^{r'} - X^r = \sum_{a: o(a) < o(r)} c_a X^a \pmod{\mathbf{M}_X}$  and so it holds that the value  $u_{r'}$  is determined from  $u_r$  via the values  $u_a$ ,  $a \in \Pi$  s.t.  $o(a) < o(r)$ , and vice versa, where  $r$  and  $r'$  are called *conjugate* to each other. We consider subsets  $\Gamma_r := \{a \leq_P r \mid a \in \mathbf{N}^2\}$  and  $\Gamma_{r'} := \{a \leq_P r' \mid a \in \mathbf{N}^2\}$ . In our terminology, we have that if  $o(r) = o(r') = l \in \mathcal{O}$ ,

$$v(l) = \#(\Gamma_r \cup \Gamma_{r'}) \cap \Pi,$$

where if such a couple does not exist,  $\Gamma_r \cup \Gamma_{r'}$  should be regarded simply as  $\Gamma_r$  for  $r$  s.t.  $o(r) = l$ .

As we show below, in case of  $t_G^\perp$  or less errors, we can find iteratively at each  $a \in 2\Pi \setminus 2\Pi(m)$  the value of the unknown syndrome  $u_a$  and update a pair of minimal polynomial set  $F$  and auxiliary polynomial set  $G$  by using the modified BMS algorithm with majority voting among the candidate syndrome values, where a pair of conjugate points are treated simultaneously at each BMS iteration, i.e.  $F$  and  $G$  are updated at each pole order  $l$  s.t.  $o(r) = o(r') = l$ . Thus, we consider the syndrome subarray  $u(l) := u^{r'}$  s.t.  $o(r') = o(r) = l$ , where  $r' = r \oplus 1 \in 2\Pi \setminus \Pi$  (if it exists), for each  $l > m$ . First we remark that  $v(l) > 2t_G$ ,  $l \geq m + 1$ . From the known syndromes  $u_a$ ,  $a \in \Pi(m)$ , we can get a minimal polynomial set  $F$  of the subarray  $u(m) = (u_a, a \in 2\Pi(m))$ . Now, assume that we have got already the syndrome subarray  $u(l)$  for some  $l \geq m$  together with  $F$  and  $G$  of  $u(l)$ , which is accompanied with the stable subsets  $\Sigma(F)$ ,  $\Delta(F)$ , and  $\Delta(G)$  (see Sakata 2009). We stipulate the following as the *total number of votes* at  $l$

$$v(l) := \#((\Gamma_r \cup \Gamma_{r'}) \cap \Pi \cap \Sigma(F)) \setminus ((r - \Delta(G)) \cup (r' - \Delta(G))),$$

where  $r - \Delta(G) := \{r - a \in \Pi \mid a \in \Delta(G)\}$ . Furthermore, for a subset  $\bar{F} \subset F$  at  $l$ , we stipulate the following as the *number of votes for  $\bar{F}$*  or *for the candidate values of the unknown syndromes determined by using  $f \in \bar{F}$  at  $l$*

$$v(\bar{F}) := \#((\Gamma_r \cup \Gamma_{r'}) \cap \Pi \cap \Sigma(\bar{F})) \setminus ((r - \Delta(G)) \cup (r' - \Delta(G))).$$

From the nature of iteration of BMS algorithm, we have the following:

**Lemma 3** *If we have a minimal polynomial set  $F^\oplus$  of  $u(l+1)$  by updating  $F$  at the iteration at  $l$ , the difference  $\#\Delta(F^\oplus) - \#\Delta(F)$  is identical with the number of votes for  $F_{\text{fail}} := \{f \in F \mid f[u]_r \neq 0 \vee f[u]_{r'} \neq 0\}$  for the pair of conjugate points  $r$  and  $r'$  at  $l$ .*

Then, we have the following conclusion, which assures the validity of the BMS algorithm with majority voting for finding the correct values of the unknown syndrome in case of correctable number of errors.

**Lemma 4** Provided the number of errors is  $t' \leq t_G^\perp$ , the polynomials  $f$  in  $F$  which give the correct syndrome values  $u_r$  or  $u_{r'}$  have the majority of votes among  $F$ .

*Proof* It is shown that  $\#((r - \Delta(G)) \cup (r' - \Delta(G))) \cap \Pi = \#\Delta(G)$ , and thus if the subset  $F_{\text{fail}}$  of  $F$  which does not give the correct syndrome values  $u_r$  or  $u_{r'}$  at  $l$  has the majority of votes, in view of Lemma 3 and  $\#\Delta(F) \setminus \Delta = \#\Delta(G)$ , we should have  $\#\Delta(F^\perp) \setminus \Delta > \#\Delta(F) \setminus \Delta + \frac{1}{2}v(l) = \#\Delta(F) \setminus \Delta + \frac{1}{2}(2t_G^\perp - \#\Delta(F) \setminus \Delta - \#\Delta(G)) = t_G^\perp$ , which contradicts the fact that for the eventual minimal polynomial set  $F$  and auxiliary polynomial set  $G$ , we have  $\#\Delta(F) \setminus \Delta (= \#\Delta(G)) = t'$ , where  $t' = \#\mathcal{E}$  for the zero variety  $V(\mathbf{M}(\mathbf{e})) = \mathcal{E}$  of the error locator submodule  $\mathbf{M}(\mathbf{e})$ .  $\square$

Our syndrome decoding method for Hermitian codes of codelength  $n$  has computational complexity  $\mathcal{O}(n^{\frac{7}{3}})$  compared with  $\mathcal{O}(n^3)$  of the method based on Gaussian elimination. This method can be applied to not only any one-point codes from algebraic curves but also codes from order domains (Høholdt et al. 1998; Geil 2009) at least when the transcendence degree is one.

### 3 Multivariate Polynomial Interpolation and List Decoding of Primal Codes

A univariate polynomial interpolation is given by the well-known *Lagrange interpolating polynomial*, i.e. given a set of  $M$  points  $\{(x^{(l)}, y^{(l)}) \in \mathbb{F}_q^2 \mid 1 \leq l \leq M\}$  in the 2-D space  $\mathbb{F}_q^2$ , where  $x^{(j)} \neq x^{(l)}$ ,  $j \neq l$ ,  $1 \leq j, l \leq M$ , a polynomial with minimum degree satisfying the interpolation condition  $f(x^{(l)}) = y^{(l)}$ ,  $1 \leq l \leq M$  is

$$f(x) = \sum_{l=1}^M y_l \frac{\prod_{j \neq l} (x - x^{(j)})}{\prod_{j \neq l} (x^{(l)} - x^{(j)})}.$$

We can consider any field, provided exact computation without numerical errors is done. However, we restrict to finite fields  $\mathbb{F}_q$  with sufficiently large  $q$  to concern ourselves with decoding of algebraic geometry codes and to make our discussions simpler.

In the general case of multivariate interpolation, we cannot always have such an explicit interpolating polynomial as above. This is the following problem. Given a set of  $M$  points  $\{(X^{(l)}, y^{(l)}) \in (\mathbb{F}_q)^{N+1} \mid 1 \leq l \leq M\}$  in the  $(N+1)$ -dimensional space  $\mathbb{F}_q^{N+1}$  over  $\mathbb{F}_q$ , where  $X^{(l)} = (x_1^{(l)}, \dots, x_N^{(l)}) \in \mathbb{F}_q^N$ ,  $y^{(l)} \in \mathbb{F}_q$ ,  $1 \leq l \leq M$  and we assume  $X^{(j)} \neq X^{(l)}$ ,  $j \neq l$ ,  $1 \leq j, l \leq M$ , we want to find a  $N$ -variate polynomial  $f$ , which is *simplest* in some sense, satisfying the following condition:

$$f(X^{(l)}) = y^{(l)}, \quad 1 \leq l \leq M. \tag{2}$$

Since this is a system of linear equations for the unknown coefficients of  $f$ , its solution is not always unique (if it exists), which is given as a sum of a (special)

solution of (2) and a general solution  $f$  of the following homogeneous system which is derived from (2) by putting  $y^{(l)} = 0$ ,  $1 \leq l \leq M$ :

$$f(X^{(l)}) = 0, \quad X^{(l)} \in V, \quad (3)$$

where  $V := \{X^{(l)} \mid 1 \leq l \leq M\} \subset \mathbb{F}_q^N$ . The set of solutions  $f$  of (3)

$$\mathbf{I}(V) := \{f \in \mathcal{P} \mid f(X^{(l)}) = 0, \quad X^{(l)} \in V\}$$

is an ideal of the ring  $\mathcal{P} = \mathbb{F}_q[x_1, \dots, x_N]$ . Thus, provided ‘simplicity’ is interpreted as ‘minimality’ as in Gröbner basis theory, the interpolation problem (2) can be divided into two subproblems, i.e. finding a Gröbner basis of the ideal corresponding to the homogeneous system (3) and obtaining a special (*minimal*) solution of the non-homogeneous system (2).

Now, for the arrays  $u^{(l)} = (u_a^{(l)})$ ,  $v^{(l)} = (v_a^{(l)})$ ,  $a \in \mathbf{N}^N$ ,  $1 \leq l \leq M$  and  $u = (u_a)$ ,  $v = (v_a)$ ,  $a \in \mathbf{N}^N$  defined by

$$\begin{aligned} u_a^{(l)} &:= (X^{(l)})^a, & v_a^{(l)} &:= y^{(l)}(X^{(l)})^a, & a \in \mathbf{N}^N, & 1 \leq l \leq M; \\ u_a &:= \sum_{1 \leq l \leq M} u_a^{(l)}, & v_a &:= \sum_{1 \leq l \leq M} v_a^{(l)}, & a \in \mathbf{N}^N, \end{aligned}$$

it holds that

**Lemma 5** A polynomial  $f = \sum_{a \in \text{supp}(f)} c(f, a)X^a$  satisfies the interpolation condition (2) iff  $f \circ u = v$ , i.e.

$$f(u)_b =: \sum_{a \in \text{supp}(f)} c(f, a)u_{a+b} = v_b, \quad b \in \mathbf{N}^N. \quad (4)$$

*Proof*

$$\begin{aligned} \sum_{a \in \text{supp}(f)} c(f, a)(X^{(l)})^a &= y^{(l)}, \quad 1 \leq l \leq M \quad \Leftrightarrow \\ \sum_{a \in \text{supp}(f)} c(f, a)(X^{(l)})^{a+b} &= y^{(l)}(X^{(l)})^b, \quad b \in \mathbf{N}^N, \quad 1 \leq l \leq M \quad \Leftrightarrow \\ \sum_{a \in \text{supp}(f)} c(f, a)u_{a+b}^{(l)} &= v_b^{(l)}, \quad b \in \mathbf{N}^N, \quad 1 \leq l \leq M \quad \Leftrightarrow \\ \sum_{a \in \text{supp}(f)} c(f, a)u_{a+b} &= v_b, \quad b \in \mathbf{N}^N, \end{aligned}$$

where the equivalence between the third and fourth conditions comes from the linear independence of the arrays  $u^{(l)}$ ,  $1 \leq l \leq M$  (Remark: we assume that  $q$  is sufficiently large).  $\square$

The linear recurrence corresponding to the homogeneous system (3) is just the homogeneous linear recurrence which is derived from (4) by letting the right-hand array  $v := 0$ , and it is easy to see that the characteristic ideal  $\mathbf{I}(u)$  of the left-hand array  $u$  is identical with  $\mathbf{I}(V)$ .

Such a multivariate interpolation problem as above appears in the context of *list decoding* (Sudan 1997; Shokrollahi and Wasserman 1999; Guruswami and Sudan 1999), which is a generalization of conventional *bounded-distance decoding* (including syndrome decoding) of algebraic geometry codes. First, we give a simple sketch of list decoding of (primal) RS codes. We take a primal ( $n = q - 1, k, d = q - k$ ) RS code  $C = \{\mathbf{c} = (f(\alpha^i))_{0 \leq i \leq n-1} \mid f \in \mathbb{F}_q[x], \deg(f) \leq k - 1\}$  and an integer  $\tau (< n)$  which is more than the number of correctable errors  $t = \lfloor \frac{n-k}{2} \rfloor$ . Given a received word  $\mathbf{r} = (r_j)_{0 \leq j \leq n-1} \in \mathbb{F}_q^n$ , we want to find all the codewords  $\mathbf{c} = (c_j)_{0 \leq j \leq n-1} \in C$  whose components differ from  $\mathbf{r}$  by at most  $\tau$  components, i.e. for  $\mathbf{r} = \mathbf{c} + \mathbf{e}$  with  $\mathbf{e} = (e_j)_{0 \leq j \leq n-1} \in \mathbb{F}_q^n$ , we assume that the size  $t' := \#\mathcal{E}$  of the error locators  $\mathcal{E} = \{\alpha^j \mid e_j \neq 0, 0 \leq j \leq n - 1\}$  is less than or equal to  $\tau$ . Then, it is shown below that list decoding is reduced to an interpolation problem, where the *leading exponent*  $\text{le}(Q) (\in \mathbb{N}^2)$  of a bivariate polynomial  $Q = Q(x, y)$  is introduced according to the term ordering  $<$  defined by the weight  $w = (1, k - 1)$  (and the lexicographic ordering  $<_L$  s.t.  $x <_L y$ ).

**Lemma 6** Assume that a nonzero bivariate polynomial  $Q(x, y)$  in  $\mathbb{F}_q[x, y]$ ,  $Q(x, y) = \sum_{(i, j) \in \text{supp}(Q)} Q_{ij}x^i y^j$ , satisfies the condition

$$Q(\alpha^j, r_j) = 0, \quad 0 \leq j \leq n - 1 \tag{5}$$

and that its leading exponent  $\text{le}(Q) < (n - \tau, 0)$ . Then, the polynomial  $f$  corresponding to a codeword  $\mathbf{c}$  within the radius  $\tau$  from the received word  $\mathbf{r}$  satisfies  $y - f(x) \mid Q(x, y)$ .

*Proof* By the condition  $\text{le}(Q) < (n - \tau, 0)$ , the univariate polynomial  $Q(x, f(x))$  has degree at most  $n - \tau - 1$ . On the other hand, since the identities  $r_l = f(\alpha^l)$  hold except for at most  $\tau$  integers  $l$ ,  $1 \leq l \leq n$ , we have that  $Q(\alpha^l, f(\alpha^l)) = 0$  for at least  $n - \tau$  integers  $l$ , from which it follows that  $Q(x, f(x)) = 0$  identically. Thus,  $y - f(x) \mid Q(x, y)$  as univariate polynomials over the polynomial ring  $\mathbb{F}_q[x]$ .  $\square$

Therefore, by finding  $Q(x, y)$  satisfying the interpolation condition (5) and furthermore finding its factors in the form of  $y - f(x)$ , we can obtain  $f$  which gives a candidate codeword. The 2-D linear recurrence derived from (5) is a special case of the homogeneous linear recurrence (4), where the right-hand side is 0. As a conclusion, we can obtain  $Q$  among a Gröbner basis of the characteristic ideal of the 2-D array  $u = (u_a)$  defined by  $u_a := \sum_{0 \leq j \leq n-1} (X^{(j)})^a$ ,  $a \in \mathbb{N}^2$  for  $X^{(j)} = (\alpha^j, r_j)$ ,  $0 \leq j \leq n - 1$ . Our method of finding the interpolation polynomial for list decoding of RS codes of codelength  $n$  and coding rate  $\frac{k}{n} = R$  has computational complexity  $\mathcal{O}(R^{-\frac{1}{2}}n^2)$ , which is  $\mathcal{O}(n^2)$  if the coding rate  $R$  is fixed as a constant when both

values  $n$  and  $k$  become asymptotically larger, compared with  $\mathcal{O}(n^3)$  of the method based simply on Gaussian elimination.

We do not discuss the existence condition of such an interpolation polynomial as above, although it is related with a practically important problem of how much list decoding can contribute to improvement of reliability in transmission. If it exists, it is the most convenient to have an interpolation polynomial  $Q$  with minimal leading exponent  $\text{le}(Q)$ .

List decoding of codes from curves also is reduced to an interpolation problem. For simplicity, we consider only primal Hermitian codes  $C := \{\mathbf{c} = (f(P_j))_{0 \leq j \leq n-1} \mid f \in L(mP_\infty) (= \mathbb{F}_q[\Pi(m)])\}$ . In this case, the *leading exponent* of a tri-variate polynomial  $Q(x, y, z)$  with support  $\text{supp}(Q)$  ( $\subset \Pi(m) \times \mathbf{N}$ ) is introduced over  $\Pi(m) \times \mathbf{N}$  according to the term ordering  $<$  defined by the weight  $w = (q_1, q_1 + 1, m)$  (and the lexicographic ordering  $<_L$  s.t.  $x <_L y <_L z$ ). Then, we have:

**Lemma 7** *We assume that a nonzero polynomial (or rather function)  $Q(P, z) = Q(x, y, z) = \sum_{(a,l) \in \text{supp}(Q)} q_{a,l} P^a z^l$  ( $\in \mathbb{F}_q[\Pi(m)][z]$ ) satisfies the condition*

$$Q(P_j, r_j) = 0, \quad 0 \leq j \leq n - 1 \quad (6)$$

*and has leading exponent  $\text{le}(Q) < (\lfloor \frac{n-\tau}{q_1} \rfloor, 0, 0)$ , where the components of  $P = (x, y)$  are viewed not only as the coordinates of  $P$  but also as functions on the curve  $\mathcal{X}$ . Then, the function  $f(x, y) \in \Pi(m)$  corresponding to a codeword  $\mathbf{c}$  within the radius  $\tau$  from the received word  $\mathbf{r}$  satisfies  $z - f(x, y) \mid Q(x, y, z)$ .*

*Proof* Since  $\text{le}(Q) < (\lfloor \frac{n-\tau}{q_1} \rfloor, 0, 0)$ , the algebraic function  $Q(x, y, f(x, y))$  has pole order less than  $n - \tau$  (at the pole  $P_\infty$ ). On the other hand, since  $r_j = f(P_j)$  except for at most  $\tau$  integers  $j$ , we have that  $Q(P_j, f(P_j)) = 0$  for at least  $n - \tau$  integers  $j$ , from which it follows that  $Q(P, f(P))$  has the total zero order of  $n - \tau$  or more. Since it does not have any other pole except for  $P_\infty$ , we have that  $Q(P, f(P)) = 0$  identically, which implies that  $z - f(x, y) \mid Q(x, y, z)$  when  $Q(x, y, z) = Q(P, z)$  is viewed as a univariate polynomial w.r.t. the main variable  $z$  over the ring  $\mathbb{F}_q[\Pi]$ .  $\square$

Also in this situation, the interpolation condition (6) is reduced to a homogeneous linear recurrence. Consequently, we can obtain  $Q$  among a Gröbner basis of the characteristic ideal of the 3-D array  $u = (u_a)$  defined by  $u_a := \sum_{0 \leq j \leq n-1} (X^{(j)})^a$ ,  $a \in \mathbf{N}^3$  for  $X^{(j)} = (P_j, r_j)$ ,  $0 \leq j \leq n - 1$ .

From the viewpoint of linear algebra, the linear recurrence (4) is nothing but a system of linear equations for unknowns  $c(f, a)$ ,  $a \in \text{supp}(f)$ . Particularly, in the 2-D case, it is just a 2-D block-Hankel or 2-D block-Toeplitz system of linear equations, where the extent  $\text{supp}(f)$  of a solution  $f$  is also unknown in our situation, distinctly from solving the ordinary system of linear equations. For the purpose of multivariate interpolation or decoding of codes, our method is unique and distinct from the known fast methods of solving block-Hankel systems or other interpolation methods.

Soon after Sudan (1997) proposed his list decoding method, Guruswami and Sudan (1999) gave an improvement called the *GS list decoding* method, which can be effective even for higher coding rate, while the original Sudan list decoding works only for coding rate  $\leq \frac{1}{3}$ . It is based on the notion of *zeros with multiplicity* defined as follows. Here we consider RS codes as in Lemma 6 for simplicity. A point  $X^{(l)} = (x^{(l)}, y^{(l)}) \in (\mathbb{F}_q)^2$  is called a *zero with multiplicity s or more* of a polynomial  $Q(x, y) = \sum_{(i,j) \in \text{supp}(Q)} Q_{ij} x^i y^j = \sum_{a \in \text{supp}(Q)} c(Q, a) X^a \in \mathbb{F}_q[x, y]$  iff in the expansion

$$Q^{(l)}(x, y) = \sum_{a \in \mathbb{N}^2} c(Q^{(l)}, a) X^a \quad (7)$$

of the polynomial  $Q^{(l)}(x, y) := Q(x + x^{(l)}, y + y^{(l)})$ , all the terms  $c(Q^{(l)}, a) X^a$  vanish, i.e.  $c(Q^{(l)}, a) = 0$ , for  $\forall a = (a_1, a_2) \in \mathbb{N}^2$  s.t.  $a_1 + a_2 < s$ . Then, we have a modification of Lemma 6:

**Lemma 8** *Assume that a nonzero bivariate polynomial  $Q(x, y) = \sum_{(i,j) \in \text{supp}(Q)} Q_{ij} x^i y^j$  ( $\in \mathbb{F}_q[x, y]$ ) has zeros  $(\alpha^j, r_j)$ ,  $0 \leq j \leq n - 1$ , each with multiplicity s or more and that it has  $\deg(Q) <_T (s(n - \tau), 0)$ . Then, the polynomial f corresponding to a codeword within the radius  $\tau$  from  $\mathbf{r}$  satisfies  $y - f(x) \mid Q(x, y)$ .*

Neglecting discussions on the error correction performance of GS list decoding, we will show that one can apply the BMS algorithm to find such an interpolation polynomial with minimal degree. First we remark the following facts.

**Lemma 9** *For a finite subset  $V = \{X^{(l)} = (x^{(l)}, y^{(l)}) \mid 0 \leq l \leq n - 1\} \subset \mathbb{F}_q^2$ , any integer s, and any point  $c \in \mathbb{N}^2$ , each of the following sets is an ideal of  $\mathbb{F}_q[x, y]$ , the former of which we call the ideal of the zero variety V with multiplicity s.*

$$\begin{aligned} \mathbf{I}(V; s) &:= \{Q(x, y) \in \mathbb{F}_q[x, y] \mid c(Q^{(l)}, a) = 0, a = (a_1, a_2) \in \mathbb{N}^2, \\ &\quad a_1 + a_2 < s, 0 \leq l \leq n - 1\}, \\ \mathbf{I}(V; c) &:= \{Q(x, y) \in \mathbb{F}_q[x, y] \mid c(Q^{(l)}, a) = 0, a = (a_1, a_2) \in \mathbb{N}^2, \\ &\quad a \leq_P c, 0 \leq l \leq n - 1\}. \end{aligned}$$

Next, for two points  $a = (a_1, a_2)$ ,  $b = (b_1, b_2) \in \mathbb{N}^2$ , we introduce the 2-D binomial coefficients

$$\binom{b}{a} := \binom{b_1}{a_1} \binom{b_2}{a_2},$$

where if it does not hold that  $a \leq_P b$ ,  $\binom{b}{a} = 0$ . Then, the coefficients  $c(Q^{(l)}, a)$  of the expansion of (7) are written as

$$c(Q^{(l)}, a) = \sum_{b \in \text{supp}(Q): b \geq_P a} \binom{b}{a} c(Q, b) (X^{(l)})^{b-a}.$$

Therefore,

**Lemma 10**  $Q = \sum_{a \in \text{supp}(Q)} c(Q, a) X^a \in \mathbf{I}(V, c) \Leftrightarrow$

$$\sum_{b \in \text{supp}(Q): b \geq pa} \binom{b}{a} c(Q, b) (X^{(l)})^{b-a} = 0, \quad a \in \Gamma_c, \quad 0 \leq l \leq n-1.$$

For a point  $c \in \mathbf{N}^2$ , we introduce a 2-D array  $u = (u_b)$  as follows:

$$u_b := \sum_{0 \leq l \leq n-1} \binom{b}{c} (X^{(l)})^{b-c}, \quad b \in \mathbf{N}^2.$$

Then,

**Lemma 11**  $Q = \sum_{a \in \text{supp}(Q)} c(Q, a) X^a \in \mathbf{I}(V, c) \Leftrightarrow Q \circ u = 0$ , i.e.

$$\sum_{a \in \text{supp}(Q)} c(Q, a) u_{a+b} = 0, \quad b \in \mathbf{N}^2.$$

For the ideal  $\mathbf{I}(V, s)$ , we introduce  $s$  2-D arrays  $u^{(i)} = (u_b^{(i)})$ ,  $1 \leq i \leq s$  as follows:

$$u_b^{(i)} := \sum_{0 \leq l \leq n-1} \binom{b}{c^{(i)}} (X^{(l)})^{b-c^{(i)}}, \quad b \in \mathbf{N}^2, \quad (8)$$

where  $c^{(i)} := (i-1, s-i) \in \mathbf{N}^2$ ,  $1 \leq i \leq s$ . Then, in view of  $\{a = (a_1, a_2) \in \mathbf{N}^2 \mid a_1 + a_2 < s\} = \cup_{1 \leq i \leq s} \Gamma_{c^{(i)}}$ , we have

**Corollary 1**  $Q \in \mathbf{I}(V, s) \Leftrightarrow Q \circ u^{(i)} = 0$ ,  $1 \leq i \leq s$ , i.e.

$$\sum_{a \in \text{supp}(Q)} c(Q, a) u_{a+b}^{(i)} = 0, \quad b \in \mathbf{N}^2, \quad 1 \leq i \leq s.$$

Consequently, it turns out that GS list decoding of primal RS codes can be solved by the multiple-array BMS algorithm (Sakata 1989), which is a modification of the BMS algorithm for finding a minimal polynomial set of a finite set of 2-D arrays  $u^{(i)}$ ,  $1 \leq i \leq s$  as in (8) with  $X^{(l)} = (\alpha^l, r_l) \in \mathbb{F}_q^2$ ,  $0 \leq l \leq n-1$ .

Compared with  $\mathcal{O}(n^3 s^6)$  of the method based simply on Gaussian elimination, our method (Numakami et al. 2000) of finding the interpolation function for GS list decoding with multiplicity  $s$  of RS codes of codelength  $n$  and coding rate  $R$  has the same computational complexity  $\mathcal{O}(R^{-\frac{1}{2}} n^2 s^4)$  as other efficient algorithms, e.g. Koetter–Vardy (2003), O’Kieffe–Fitzpatrick (2002), Lee–O’Sullivan (2006), but our method is unique in the sense that it uses (syndrome-like) arrays which contain in

the condensed form all the information necessary for decoding. For GS list decoding of algebraic geometry codes, there have been several approaches (Sakata 2001; O’Keeffe and Fitzpatrick 2007; Lee and O’Sullivan 2008), etc., which we do not treat here because we need more involved discussions for that purpose. For general multivariate interpolation the Buchberger–Möller (1982, 2009a) and the Marinami–Möller–Mora (1993) algorithm are alternatives, in comparison with which the BMS algorithm is conjectured to have less computational complexity, depending on the situations, although the exact estimations remain to be investigated.

## 4 Other Relevant Decoding Methods of Primal/Dual Codes

In this section, we consider a special case of Sudan list decoding, i.e. the case of list size 1. In this case, we treat nothing but polynomials of degree 1 w.r.t. the main variable and bounded-distance decoding of primal codes up to half the correction bound.<sup>3</sup>

Again we take a primal ( $n = q - 1, k, d = q - k$ ) RS code, and we assume that the number  $\tau$  of errors is less than  $\frac{d}{2}$  as in Sect. 2. As a corollary of Lemma 6, we have

**Lemma 12<sup>4</sup>** *If a bivariate polynomial of the form*

$$Q(x, y) = Q_0(x) - yQ_1(x) \quad (\neq 0) \quad (\in \mathbb{F}_q[x, y])$$

*satisfies the conditions*

- (1)  $\deg(Q_0(x)) < n - \tau, \deg(Q_1(x)) < n - \tau - (k - 1);$
  - (2)  $Q(\alpha^j, r_j) = 0, \quad 0 \leq j \leq n - 1,$
- (9)

*then  $Q_1(x)$  is an error locator polynomial which has  $\mathcal{E}$  as its zeros, i.e.  $Q_1(\alpha^j) = 0$  for  $\alpha^j \in \mathcal{E}$ , and  $Q_1(x) \mid Q_0(x)$  so that the quotient  $f(x) = \frac{Q_0(x)}{Q_1(x)}$  is the message polynomial corresponding to the sent codeword  $\mathbf{c} = (c_j)$ , i.e.  $c_j = f(\alpha^j)$ .*

In fact, such a polynomial  $Q(x, y)$  exists as shown in the following lemma so that we can obtain it by applying the BMS algorithm to the 2-D array  $u = (u_a)$  defined by  $u_a := \sum_{0 \leq j \leq n-1} (X^{(j)})^a, a \in \mathbb{N}^2$  for  $X^{(j)} = (\alpha^j, r_j), 0 \leq j \leq n - 1$  similarly to list decoding, where in this case we do not need to be worried about factorization of  $Q(x, y)$ .

<sup>3</sup>Of course, a primal code can be decoded as a dual code of its dual by using syndrome decoding. But, sometimes from both the practical and theoretical points of view it is required to have some direct decoding method as a primal code itself.

<sup>4</sup>This lemma is given in Justesen and Høholdt (2004).

**Lemma 13** *There exists at least one nonzero polynomial  $Q(x, y)$  as in Lemma 12.*

Sine we assume that  $\tau$  is less than or equal to the number of correctable errors  $t = \lfloor \frac{d-1}{2} \rfloor$ , there exists only a single codeword  $\mathbf{c}$  s.t.  $\text{dis}(\mathbf{c}, \mathbf{r}) \leq \tau$  and thus the above method gives us the ordinary bounded-distance decoding of primal RS codes. By the way, the above method based on the 2-D BMS algorithm can be replaced by the vectorial BM algorithm, which is the 1-D vectorial BMS algorithm. First, we take  $n$  pairs of 1-D arrays  $v^{(j)} = (v_i^{(j)})$ ,  $w^{(j)} = (w_i^{(j)})$ ,  $i \in \mathbb{N}$ ,  $0 \leq j \leq n - 1$  defined by

$$v_i^{(j)} := (\alpha^j)^i, \quad w_i^{(j)} := -r_j(\alpha^j)^i, \quad i \in \mathbb{N},$$

from which we have a pair of 1-D arrays  $v = (v_i)$ ,  $w = (w_i)$  defined by

$$v_i := \sum_{j=1}^n v_i^{(j)}, \quad w_i := \sum_{j=1}^n w_i^{(j)}, \quad i \in \mathbb{N}.$$

Then, we have

**Lemma 14** *The condition (9) is equivalent to the compound linear recurrence*

$$\sum_{i=0}^{d_0} c(Q_0, i)v_{i+j} + \sum_{i=0}^{d_1} c(Q_1, i)w_{i+j} = 0, \quad j \in \mathbb{N}. \quad (10)$$

Thus, we can apply the vectorial BM algorithm (Sakata 1991, 2009) to the pair  $(v, w)$  of 1-D arrays so that we can have a Gröbner basis of the module defined by the pair of arrays as a minimal polynomial vector set, in which the desired solution  $(Q_0, Q_1)$  is contained. Thus, we have another method of the ordinary bounded-distance decoding of primal RS codes.<sup>5</sup> In form, this method is similar to the decoding method (Sakata 2006) based on the vectorial BM algorithm which we gave as an alternative to the Welch–Berlekamp (1986) decoding algorithm of the dual RS code, where we have instead of the condition (9)

$$Q\left(\alpha^j, \frac{r_j}{p_j \alpha^j}\right) = 0, \quad 0 \leq j \leq d - 2, \quad (11)$$

where  $p_j$ ,  $0 \leq j \leq d - 2$  are defined by

$$p(x) = \prod_{i=1}^{d-2} (x - \alpha^i) = \sum_{j=0}^{d-2} p_j x^j. \quad (12)$$

---

<sup>5</sup>The vectorial BMS algorithm (Sakata 1991, 2009) for any dimension  $N$  is given in 1991. Fitzpatrick (1995) gave a similar method, which may be considered to be equivalent to a version of the vectorial BM algorithm according to Blackburn–Chambers' (1996) explanation, where the swapping based on the special term ordering  $<_r$  used in the Fitzpatrick algorithm corresponds to the degree change in the (vectorial) BM algorithm.

For the primal Hermitian code  $C(m)$  we have a corollary of Lemma 7.

**Lemma 15** *If a trivariate polynomial*

$$Q(x, y, z) = Q_0(x, y) - zQ_1(x, y) \in \mathbb{F}_q[\Pi(m)][z]$$

satisfies the conditions

- $$\begin{aligned} (1) \quad & o(Q_0) \leq m + \tau + g, \quad o(Q_1) \leq \tau + g; \\ (2) \quad & Q(x_l, y_l, r_l) = 0, \quad 0 \leq l \leq n-1, \end{aligned} \tag{13}$$

then  $Q_1(x, y)$  is an error locator function which has  $\mathcal{E}$  as its zeros, i.e.  $Q_1(P_j) = 0$  for  $P_j \in \mathcal{E}$ , and  $Q_1(x, y) \mid Q_0(x, y)$  so that the quotient  $f(x, y) := \frac{Q_0(x, y)}{Q_1(x, y)}$  is the message function corresponding to the sent codeword  $\mathbf{c}$ , i.e.  $c_j = f(P_j)$ ,  $0 \leq j \leq n-1$ .

In fact, such a function  $Q(x, y, z)$  exists as shown in the following lemma so that we can obtain it by applying the 3-D BMS algorithm to the 3-D array  $u = (u_a)$  defined by  $u_a := \sum_{0 \leq j \leq n-1} (X^{(j)})^a$ ,  $a \in \mathbb{N}^3$  for  $X^{(j)} = (P_j, r_j)$ ,  $0 \leq j \leq n-1$  similarly to the list decoding, where in this case we do not need to be worried about factorization of  $Q(x, y, z)$ .

**Lemma 16** *There exists at least one nonzero function  $Q(x, y, z)$  as in Lemma 15.*

If  $\tau$  is less than or equal to  $\hat{\tau} = \lfloor \frac{d_G - g - 1}{2} \rfloor$  ( $< t_G$ ), then there exists only a single codeword  $\mathbf{c}$  s.t.  $\text{dis}(\mathbf{c}, \mathbf{r}) \leq \tau$  and thus this method (Fujisawa and Sakata 2005) gives us the ordinary bounded-distance decoding of primal Hermitian codes up to  $\hat{\tau}$ . By the way, the method based on the 3-D BMS algorithm can be replaced by the vectorial 2-D BMS algorithm. Instead of the 3D array  $u$  as above, we take a pair of 2D arrays  $v = (v_a)$ ,  $w = (w_a)$ ,  $a = (a_1, a_2) \in \Pi$  defined by

$$v_a := \sum_{0 \leq l \leq n-1} P_l^a = \sum_{0 \leq l \leq n-1} (\alpha_l)^{a_1} (\beta_l)^{a_2}, \tag{14}$$

$$w_a := - \sum_{0 \leq l \leq n-1} r_l P_l^a = - \sum_{0 \leq l \leq n-1} r_l (\alpha_l)^{a_1} (\beta_l)^{a_2}, \tag{15}$$

for which the following *compound* linear recurrence must hold:

$$\sum_{a \in \text{supp}(g)} c(g, a) v_{a+b} + \sum_{a \in \text{supp}(h)} c(h, a) w_{a+b} = 0, \quad b \in \Pi, \tag{16}$$

where  $g(:= Q_0) = \sum_{a \in \text{supp}(g)} c(g, a) X^a$  and  $h(:= Q_1) = \sum_{a \in \text{supp}(h)} c(h, a) X^a$ . Thus, we can apply the vectorial BMS algorithm to the pair  $(v, w)$  of 2-D arrays

so that we can have a Gröbner basis of the module defined by the pair of arrays as a minimal polynomial vector set, in which the desired solution  $(g, h) = (Q_0, Q_1)$  is contained. Thus, we have another method of the ordinary bounded-distance decoding of primal Hermitian codes up to  $\hat{t}$ . Furthermore, it is shown in Fujisawa et al. (2006) that most of errors up to half the Goppa bound  $d_G$  of the code  $C(m)$  over a large finite field  $\mathbb{F}_q$  can be corrected by the decoding method, i.e. for  $t := \lfloor \frac{d_G - 1}{2} \rfloor$ ,  $1 - \frac{1}{q}$  of  $t$  or less errors can be corrected.

We should not ignore the fact that the interpolation problems (9), (13) can be solved either by Buchberger–Möller (1982) algorithm or Mariani–Möller–Mora (1993) algorithm, both of which are a general method of multi-variate interpolation problem although our method based on the BMS algorithm discussed above also is a general method of multi-variate interpolation problem, or by the Farr–Gao (2005) algorithm which is explained as a generalization of Newton’s interpolation for univariate polynomial. Our method seems to have less computational complexity than them, but the exact comparison remains to be investigated.

Recently a novel decoding algorithm of primal RS codes which is based on higher-dimensional interpolation has been published by Parvaresh and Vardy (2005). Its error correction performance is superior to GS list decoding, where the ratios of the number of correctable errors per the codelength are  $\frac{\tau_{PV}}{n} = 1 - R^{\frac{N}{N+1}}$ , if  $(N + 1)$ -variate polynomial interpolation is used, for the Parvaresh–Vardy (PV) method and  $\frac{\tau_{GS}}{n} = 1 - R^{\frac{1}{2}}$  for GS method, respectively. In fact, GS list decoding is a special case of  $N = 1$  of the PV method. In case of  $N = 2$ , in encoding, the PV method gives not only the codeword of  $\mathbf{c} = (c_j) = \text{ev}(f) \in C$  for a message polynomial  $f(x) = \sum_{i=0}^{k-1} f_i x^i \in \mathcal{K}[x]$  of the actual RS code  $C (\subset \mathcal{K}^n)$  but also another codeword  $\mathbf{c}' := \text{ev}(g) \in C$  for  $g(x) = (f(x))^a \bmod h(x)$ , and then sends the pair of codewords  $\mathbf{c}, \mathbf{c}' \in C$ , where  $h(x) \in \mathcal{K}[x]$  is an irreducible polynomial over  $\mathcal{K}$  of degree  $k$ , and  $a$  is any integer satisfying a special condition. In decoding, given a pair of received words  $\mathbf{y} = (y_j), \mathbf{z} = (z_j) \in \mathcal{K}^n$ , one tries to find a Gröbner basis of the ideal

$$\mathbf{I}(\mathbf{y}, \mathbf{z}) := \{Q(x, y, z) \in \mathcal{K}[x, y, z] \mid Q(\alpha^j, y_j, z_j) = 0, 0 \leq j \leq n - 1\}$$

w.r.t. the term order defined by the weight  $(1, k - 1, k - 1)$ . Then, from the minimum element  $Q_m(x, y, z)$  of  $\mathbf{I}(\mathbf{y}, \mathbf{z})$  one computes  $P(y, z) = Q_m(x, y, z) \bmod h(x)$ , interpreted as an element of  $\tilde{\mathcal{K}}[y, z]$ , where  $\tilde{\mathcal{K}} \simeq \mathcal{K}[x]/\langle h(x) \rangle$  is the extension field of  $\mathcal{K}$ , and obtains the univariate polynomial  $\tilde{P}(y) := P(y, y^a) \in \tilde{\mathcal{K}}[y]$ , whose roots  $\in \tilde{\mathcal{K}}$  can be candidates of the message polynomial  $f(x) \in \mathcal{K}[x]$ . Thus, the multivariate interpolation, which is a key step of the PV decoding method, can be solved by the BMS algorithm efficiently.

## 5 Conclusion

We have discussed how the BMS algorithm and its variations (Sakata 1988, 1989, 1990, 1991, 2009) are applied to various decoding methods of algebraic geometry

codes and multivariate interpolation related to list decoding, and how these decoding methods are connected with Gröbner bases via multidimensional arrays and linear recurrences. Although we have explained our decoding methods mainly as regards Reed–Solomon codes and Hermitian codes, our methods work for one-point codes from any algebraic curves and codes from order domains. For example, primal and dual one-point codes which have an  $\overline{\mathbb{F}}_q(f_\rho)$ -module basis (see Sect. 7 of Leonard 2009a) can be decoded by the vectorial BMS algorithm. In the sequel, we have clarified that these problems are reduced to finding a set of minimal polynomials, which corresponds to a Gröbner basis, of a given (set of) multidimensional array(s).<sup>6</sup> We have given a basic set of algorithms for solving these problems, which constitute a unified system of unique methods in comparison with other various relevant methods related to Gröbner bases. In fact, there have been many other pioneering investigations (Justesen et al. 1989, 1992; Pellikaan 1989, 1993; Skorobogatov and Vlăduț 1990; Porter et al. 1992; Shen 1992; Duursma 1993; Ehrhard 1993; Feng et al. 1994), etc.<sup>7</sup> on decoding algebraic geometry codes, but those are less efficient than our methods based on the Gröbner basis theory (Buchberger 1965, 1970, 1985, 1998, 2006) and the BMS algorithm (Sakata 1988, 1990). In Leonard (2009b), Guerrini and Rimoldi (2009) in this issue, other decoding methods from Gröbner basis perspectives are discussed. For encoding of AG codes, see Little (2009).

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

## References

- D. Augot, *A parallel version of a special case of the Sudan list decoding algorithm*, Proc. of ISIT2002, 2002, pp. 86–86.
- D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- S. R. Blackburn and W. G. Chambers, *Some remarks on an algorithm of Fitzpatrick*, IEEE Trans. on Inf. Th. **42** (1996), no. 4, 1269–1271.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.

<sup>6</sup>These facts are based on the well-known correspondence between ideals and varieties (zeros) (Cox et al. 1992), but a little bit distinct from the duality discussed by Mora (2009a) in this issue.

<sup>7</sup>See Høholdt et al. (1998).

- D. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms*, Springer, Berlin, 1992, An introduction to computational algebraic geometry and commutative algebra.
- I. M. Duursma, *Majority coset decoding*, IEEE Trans. on Inf. Th. **39** (1993), no. 3, 1067–1070.
- D. Ehrhard, *Achieving the designed error capacity in decoding algebraic-geometric codes*, IEEE Trans. on Inf. Th. **39** (1993), no. 3, 743–751.
- J. B. Farr and S. Gao, *Gröbner bases, Padé approximation, and decoding of linear codes*, Contemp. Math. **381** (2005), 3–18.
- G. L. Feng and T. R. N. Rao, *Decoding algebraic-geometric codes up to the designed minimum distance*, IEEE Trans. on Inf. Th. **39** (1993), no. 1, 37–45.
- G. L. Feng, V. K. Wei, T. R. N. Rao, and K. Tzeng, *Simplified understanding and efficient decoding of algebraic geometric codes*, IEEE Trans. on Inf. Th. **40** (1994), no. 4, 981–1002.
- P. Fitzpatrick, *On the key equation*, IEEE Trans. on Inf. Th. **41** (1995), no. 5, 1290–1302.
- M. Fujisawa and S. Sakata, *On a fast method of bounded-distance decoding based on Sudan’s algorithm for one-point algebraic geometry code*, Proc. of SITA2005, 2005, pp. 543–546.
- M. Fujisawa, H. Matsui, M. Kurihara, and S. Sakata, *With a higher probability one can correct error up to the designed distance for primal codes from curves*, Proc. of SITA2006, 2006, pp. 101–104.
- O. Geil, *Algebraic geometry codes from order domains*, this volume, 2009, pp. 121–141.
- V. D. Goppa, *Codes on algebraic curves*, Soviet Math. Dokl. **24** (1981), no. 1, 170–172.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick’s approach to recent developments*, this volume, 2009, pp. 197–218.
- V. Guruswami and M. Sudan, *Improved decoding of Reed–Solomon and algebraic geometric codes*, IEEE Trans. on Inf. Th. **45** (1999), no. 6, 1757–1767.
- T. Høholdt, J. H. van Lint, and R. Pellikaan, *Algebraic geometry of codes*, Handbook of coding theory, vols. I, II (V. S. Pless and W.C. Huffman, eds.), North-Holland, Amsterdam, 1998, pp. 871–961.
- J. Justesen and T. Høholdt, *A course in error-correcting codes*, EMS textbooks in mathematics, EMS, 2004.
- J. Justesen, Larsen K. J., Jensen H. E., A. Havemose, and T. Høholdt, *Construction and decoding of a class of algebraic geometry codes*, IEEE Trans. on Inf. Th. **35** (1989), no. 4, 811–821.
- J. Justesen, K. J. Larsen, H. E. Jensen, and T. Høholdt, *Fast decoding of codes from algebraic plane curves*, IEEE Trans. on Inf. Th. **38** (1992), no. 1, 111–119.
- R. Kötter, *On decoding of algebraic-geometric and cyclic codes*, Ph.D. thesis, Linköping University, 1996.
- R. Kötter and A. Vardy, *Algebraic soft-decision decoding of Reed–Solomon codes*, Trans. on Inf. Th. **49** (2003), no. 11, 2809–2825.
- K. Lee and M. E. O’Sullivan, *Sudan’s list decoding of RS codes from a Gröbner basis perspective*, preprint, 2006, [arXiv:math/0601022](https://arxiv.org/abs/math/0601022).
- K. Lee and M. E. O’Sullivan, *List decoding of Reed–Solomon codes from a Gröbner basis perspective*, J. Symbolic Comput. **43** (2008), no. 9, 645–658.
- D. A. Leonard, *A tutorial on AG code construction from a Gröbner basis perspective*, this volume, 2009a, pp. 93–106.
- D. A. Leonard, *A tutorial on AG code decoding from a Gröbner basis perspective*, this volume, 2009b, pp. 187–196.
- J. B. Little, *Automorphisms and encoding of AG and order domain codes*, this volume, 2009, pp. 107–120.
- M. G. Marinari, H. M. Möller, and T. Mora, *Gröbner bases of ideals defined by functionals with an application to ideals of projective points*, AAECC **4** (1993), no. 2, 103–145.
- H. M. Möller and B. Buchberger, *The construction of multivariate polynomials with preassigned zeros*, LNCS, vol. **144**, Springer, Berlin, 1982, pp. 24–31.
- T. Mora, *The FGLM problem and Möller’s algorithm on zero-dimensional ideals*, this volume, 2009a, pp. 27–45.
- T. Mora, *Gröbner technology*, this volume, 2009b, pp. 11–25.
- Y. Numakami, M. Fujisawa, and S. Sakata, *Fast interpolation methods for list decoding of RS codes*, IEICE Trans. Fundamentals **J83** (2000), 1309–1317.

- H. O’Keeffe and P. Fitzpatrick, *Gröbner bases solutions of constrained interpolation problems*, Linear algebra and its applications **351–352** (2002), 533–551.
- H. O’Keeffe and P. Fitzpatrick, *Gröbner basis approach to list decoding of algebraic geometry codes*, AAECC **18** (2007), no. 5, 445–466.
- M. E. O’Sullivan, *Decoding of codes defined by a single point on a curve*, IEEE Trans. on Inf. Th. **41** (1995), no. 6, 1709–1719, part 1.
- F. Parvaresh and A. Vardy, *Correcting errors beyond the Guruswami–Sudan radius in polynomial time*, Proc. of IEEE FOCS2005, IEEE Computer Society, 2005, pp. 285–294.
- R. Pellikaan, *On a decoding algorithm for codes on maximal curves*, IEEE Trans. on Inf. Th. **35** (1989), no. 6, 1228–1232.
- R. Pellikaan, *On the efficient decoding of algebraic-geometric codes*, Eurocode ’92, CISM courses and lectures, vol. **339**, Springer, Berlin, 1993, pp. 231–253.
- S. C. Porter, B. Z. Shen, and R. Pellikaan, *Decoding geometric Goppa codes using an extra place*, IEEE Trans. on Inf. Th. **38** (1992), no. 6, 1663–1676.
- K. Saints and C. Heegard, *Algebraic-geometric codes and multidimensional cyclic codes: a unified theory and algorithms for decoding using Gröbner bases*, IEEE Trans. Inform. Theory **41** (1995), no. 6, 1733–1751.
- S. Sakata, *Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array*, J. Symbolic Comput. **5** (1988), no. 3, 321–337.
- S. Sakata, *n-dimensional Berlekamp–Massey algorithm for multiple arrays and construction of multivariate polynomials with preassigned zeros*, LNCS, vol. **357**, Springer, Berlin, 1989, pp. 356–376.
- S. Sakata, *Extension of the Berlekamp–Massey algorithm to N dimensions*, Inform. and Comput. **84** (1990), no. 2, 207–239.
- S. Sakata, *Finding a minimal polynomial vector set of a vector of nD arrays*, LNCS, vol. **539**, Springer, Berlin, 1991, pp. 414–425.
- S. Sakata, *On fast interpolation method for Guruswami–Sudan list decoding of one-point AG codes*, LNCS, vol. **2227**, Springer, Berlin, 2001, pp. 172–181.
- S. Sakata, *Efficient factorization methods for list decoding of code from curves*, Proc. of ISIT2003, 2003, pp. 363–363.
- S. Sakata, *A comparison between WB algorithm and BM algorithm*, Proc. of ISITA2006, 2006, pp. 244–247.
- S. Sakata, *The BMS algorithm*, this volume, 2009, pp. 143–163.
- S. Sakata and M. Fujisawa, *WB-like decoding algorithm of one-point codes from curves*, Proc. of SITA2006, 2006, pp. 93–96.
- S. Sakata, H. E. Jensen, and T. Høholdt, *Generalized Berlekamp–Massey decoding of algebraic-geometric codes up to half the Feng–Rao bound*, IEEE Trans. on Inf. Th. **41** (1995a), no. 6, 1762–1768, part 1.
- S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Høholdt, *A fast decoding method of AG codes from Miura–Kamiya curves  $C_{ab}$  up to half the Feng–Rao bound*, Finite Fields Appl. **1** (1995b), no. 1, 83–101.
- S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Høholdt, *Fast decoding of algebraic-geometric codes up to the designed minimum distance*, IEEE Trans. on Inf. Th. **41** (1995c), no. 6, 1672–1677, part 1.
- B. Z. Shen, *Algebraic-geometric codes and their decoding algorithm*, Ph.D. thesis, Eindhoven Univ. Tech., 1992.
- M. A. Shokrollahi and H. Wasserman, *List decoding of algebraic-geometric codes*, IEEE Trans. on Inf. Th. **45** (1999), no. 2, 432–437.
- A. N. Skorobogatov and S. G. Vlăduț, *On the decoding of algebraic-geometric codes*, IEEE Trans. on Inf. Th. **36** (1990), no. 5, 1051–1060.
- H. Stichtenoth, *A note on Hermitian codes over  $GF(q^2)$* , IEEE Trans. on Inf. Th. **34** (1988), 1345–1348.
- M. Sudan, *Decoding of Reed–Solomon codes beyond the error correction bound*, J. of Complexity **13** (1997), 180–193.
- L. R. Welch and E. R. Berlekamp, *Error correction for algebraic block codes*, U.S. Patent No. 4633470, 1986.

# A Tutorial on AG Code Decoding from a Gröbner Basis Perspective

Douglas A. Leonard

## 1 Introduction

The notation and AG code description were set up in Leonard (2009). While syndrome decoding of RS codes dates back at least to the 60's, the syndrome decoding of AG codes should be viewed in terms of Sakata's generalization of the Berlekamp–Massey algorithm (see Sakata 2009a, 2009b) and Feng and Rao's (1993) majority voting scheme to decode up to the designed minimum distance.

Sudan popularized list-decoding, but the important follow-up papers are Høholdt and Nielsen (1999) (for a much more readable introduction to Sudan's ideas), and Roth and Ruckenstein (2000), Wu and Siegel (2001), and Augot and Pecquet (2000) for three differently flavored views on implementing these ideas.

## 2 Functional Decoding of RS Codes and AG Codes Using Syndromes and Error-Locator Ideals

Moving from RS codes to AG codes means moving theoretically from *univariate polynomial rings* to *multivariate polynomial rings*. In univariate polynomial rings all *ideals* are *principal* (that is, have a single generator), so finding generators for the *error-locator ideals* is equivalent to finding a single *error-locator polynomial*. The generalization to multivariate polynomial rings is to finding *Gröbner bases* (Buchberger 1970, 1985, 1998, 2009) for the error-locator ideals, that is ideal bases with leading monomials that divide leading monomials of any elements in the ideal.

Syndrome decoding algorithms recursively compute such Gröbner bases that are consistent with the initial part of the sequence of syndromes; that is,  $\sum_{i=0}^t \sigma_i s_{i+j} = 0$ . If there is an error of smallest weight  $t \leq e$ , then the  $\Delta$ -set produced by such algorithms will have size  $t$ , and the *variety* of the ideal will also have size  $t$ .

---

D.A. Leonard

Department of Mathematics and Statistics, Auburn University, Auburn, AL 36849, USA  
e-mail: [leonada@auburn.edu](mailto:leonada@auburn.edu)

Let  $\mathbb{F}_{2^4} := \mathbb{F}_2[\gamma]/\langle 1 + \gamma + \gamma^4 \rangle$ . Consider the example syndrome vector (consisting of functions  $\underline{e}\text{Eval}^T = \underline{r}\text{Eval}^T$ ) of the supposed error  $\underline{e}$  of weight  $t$  at most 3:

$$\underline{s} := (0 \ \gamma^1 \ \gamma^1 \ \gamma^2 \ \gamma^{13} \ \gamma^1)$$

relative to the underlying functions  $(x^i : 0 \leq i \leq 5)$ , which is a shiftable shorthand for the (backshifted or Hankel) syndrome matrix

$$S := \text{Eval}\Delta(\underline{r})\text{Eval}^T = \begin{pmatrix} 0 & \gamma^1 & \gamma^1 & \gamma^2 & \gamma^{13} & \gamma^1 \\ \gamma^1 & \gamma^1 & \gamma^2 & \gamma^{13} & \gamma^1 & \gamma^1 \\ \gamma^1 & \gamma^2 & \gamma^{13} & \gamma^1 & \gamma^1 & \gamma^1 \\ \gamma^2 & \gamma^{13} & \gamma^1 & \gamma^1 & \gamma^1 & \gamma^1 \\ \gamma^{13} & \gamma^1 & \gamma^1 & \gamma^1 & \gamma^1 & \gamma^1 \\ \gamma^1 & & & & & \end{pmatrix}$$

with  $(i, j)$ -th entry clearly of the form  $\sum_P x^i(P)r(P)x^j(P)$ . (In matrix terminology, an error-locator polynomial corresponds to a linear dependence  $\underline{\sigma}$  among the rows of  $S$ ; which can be gotten (inefficiently) by simple row-reduction techniques.

The *Berlekamp–Massey algorithm*, the *extended Euclidean algorithm*, or even the standard matrix row-reduction will produce recursively at least the sequence  $1+0x, \gamma^4+x+x^2$ , and  $\sigma(x) := \gamma^4 + \gamma^{14}x + \gamma^7x^2 + x^3$ , consistent with initial parts of the total sequence of syndromes, with the former two being more efficient, in that they take advantage of the back-shifted nature of the matrix to avoid recalculating intermediate results.

The factorization  $\sigma(x) = (x + \gamma^0)(x + \gamma^1)(x + \gamma^3)$  gives the variety  $\{\gamma^0, \gamma^1, \gamma^3\}$  of *error positions* from which various other algorithms can be used to produce the *error magnitudes*.

Consider an example for the *Hermitian code* with affine definition given by the single generator  $f := x_2^4 + x_2 - x_1^5$  having  $1 + 16 + 2 \cdot 6\sqrt{16} = 65 > 1 + 16$  projective points (equal to the Hasse–Weil bound) rational over  $\mathbb{F}_{16}$ , and genus  $g = 6$ .

Since  $f_4 := x_1$  has pole order 4 and  $f_5 := x_2$ , pole order 5 at the projective point  $P_\infty := (1 : 0 : 0)$  at which these rational functions have all their poles, there are functions of the form  $f_5^{i_5} f_4^{i_4}$  of every pole order other than the  $g = 6$  values 1, 2, 3, 6, 7, 11.

$\mathcal{L}(m \cdot P_\infty)$  is given by an  $\overline{\mathbb{F}_2}[f_4]$ -module basis  $(1, f_5, f_5^2, f_5^3)$ , and the curve  $\mathcal{X}$  is defined by the *quotient ring*

$$\mathcal{Q} := \overline{\mathbb{F}_2}[f_5, f_4]/\mathcal{I}, \quad \mathcal{I} := \langle f_5^4 + f_4^5 + f_5 \rangle$$

with the monomial order a weighted total-degree order relative to the weights  $(5, 4)$ , the pole orders of the variables.

Consider the original example Feng and Rao (1993) used to exemplify majority voting to determine extra syndromes (also used by Leonard (1995) to introduce the idea of an error-locator ideal and the computation of a basis for same). The syndrome “vector” (now a 2-dimensional array, given that there are two variables

involved)

$$\underline{s} := \begin{pmatrix} \gamma^1 & \gamma^{14} & \gamma^{11} & \gamma^4 & \gamma^0 & \gamma^1 & \gamma^5 & \gamma^{12} & \gamma^2 & \gamma^6 \\ \gamma^2 & \gamma^4 & \gamma^{11} & \gamma^6 & \gamma^{12} & \gamma^7 & \gamma^1 & \gamma^{14} & & \\ \gamma^9 & \gamma^{10} & \gamma^8 & \gamma^5 & \gamma^1 & \gamma^7 & 0 & & & \\ \gamma^5 & \gamma^2 & \gamma^{10} & \gamma^4 & \gamma^9 & \gamma^0 & & & & \\ & & & & & & & & & \\ \gamma^5 & \gamma^8 & \gamma^0 & \gamma^3 & \gamma^4 & & & & & \\ \gamma^0 & & & & & & & & & \end{pmatrix}$$

with entries  $\sum_P r(P) f_5^i(P) f_4^j(P)$  relative to the underlying functions  $h_{i,j} := f_5^i f_4^j$ :

$$\begin{pmatrix} 1 & f_4 & f_4^2 & f_4^3 & f_4^4 & f_4^5 & f_4^6 & f_4^7 & f_4^8 & f_4^9 \\ f_5 & f_5 f_4 & f_5 f_4^2 & f_5 f_4^3 & f_5 f_4^4 & f_5 f_4^5 & f_5 f_4^6 & f_5 f_4^7 & f_5 f_4^8 & \\ f_5^2 & f_5^2 f_4 & f_5^2 f_4^2 & f_5^2 f_4^3 & f_5^2 f_4^4 & f_5^2 f_4^5 & f_5^2 f_4^6 & f_5^2 f_4^7 & f_5^2 f_4^8 & \\ f_5^3 & f_5^3 f_4 & f_5^3 f_4^2 & f_5^3 f_4^3 & f_5^3 f_4^4 & f_5^3 f_4^5 & f_5^3 f_4^6 & f_5^3 f_4^7 & f_5^3 f_4^8 & \\ f_5^4 & f_5^4 f_4 & f_5^4 f_4^2 & f_5^4 f_4^3 & f_5^4 f_4^4 & f_5^4 f_4^5 & f_5^4 f_4^6 & f_5^4 f_4^7 & f_5^4 f_4^8 & \\ f_5^5 & & & & & & & & & \end{pmatrix}$$

is a shorthand (shifttable in both directions) for the (almost backshifted) syndrome matrix  $S := \text{Eval}\Delta(\underline{r})\text{Eval}^T = \text{Eval}\Delta(\underline{e})\text{Eval}^T$ :

$$\begin{array}{ccccccccccccccccccccccccc} \gamma^1 & \dots & \gamma^{14} & \gamma^2 & \dots & \gamma^{11} & \gamma^4 & \gamma^9 & \gamma^4 & \gamma^0 & \gamma^{11} & \gamma^{10} & \gamma^5 & \gamma^0 & \gamma^6 & \gamma^8 & \gamma^2 & \gamma^1 & \gamma^{12} & \gamma^5 & \gamma^{10} & \gamma^5 \\ \dots & & \dots & \dots & & \dots \\ \gamma^{14} & \dots & \gamma^{11} & \gamma^4 & \dots & \gamma^4 & \gamma^{11} & \gamma^{10} & \gamma^0 & \gamma^6 & \gamma^8 & \gamma^2 & \gamma^1 & \gamma^{12} & \gamma^5 & \gamma^{10} & \gamma^5 \\ \gamma^2 & \dots & \gamma^4 & \gamma^9 & \dots & \gamma^{11} & \gamma^{10} & \gamma^5 & \gamma^6 & \gamma^8 & \gamma^2 & \gamma^5 & \gamma^{12} & \gamma^5 & \gamma^{10} & \gamma^8 & & & & & & \\ \dots & & \dots & \dots & & \dots & & & & & & & \\ \gamma^{11} & \dots & \gamma^4 & \gamma^{11} & \dots & \gamma^0 & \gamma^6 & \gamma^8 & \gamma^1 & \gamma^{12} & \gamma^5 & \gamma^{10} & \gamma^5 \\ \gamma^4 & \dots & \gamma^{11} & \gamma^{10} & \dots & \gamma^6 & \gamma^8 & \gamma^2 & \gamma^{12} & \gamma^5 & \gamma^{10} & \gamma^8 \\ \gamma^9 & \dots & \gamma^{10} & \gamma^5 & \dots & \gamma^8 & \gamma^2 & \gamma^5 & \gamma^5 & \gamma^{10} & \gamma^8 & & \\ \dots & & \dots & \dots & & \dots & \dots & \dots & \dots & \dots & \dots & & \\ \gamma^4 & \dots & \gamma^4 & \gamma^{11} & \dots & \gamma^1 & \gamma^{12} & \gamma^5 & \gamma^5 \\ \gamma^{11} & \dots & \gamma^{11} & \gamma^{10} & \dots & \gamma^{12} & \gamma^5 & \gamma^{10} & \\ \gamma^{10} & \dots & \gamma^{10} & \gamma^5 & \dots & \gamma^5 & \gamma^{10} & \gamma^8 \\ \gamma^5 & \dots & \gamma^5 & \gamma^0 & \dots & \gamma^{10} & \gamma^8 \\ \gamma^4 & \dots & \gamma^1 & \gamma^{12} & \dots & \gamma^5 \\ \gamma^{11} & \dots & \gamma^{12} & \gamma^5 & \dots & \\ \gamma^{10} & \dots & \gamma^5 & \gamma^{10} & \dots & \\ \gamma^5 & \dots & \gamma^{10} & \gamma^8 & & \\ \gamma^1 & \dots & \gamma^5 & & & \\ \gamma^{12} & \dots & & & & \\ \gamma^5 & \dots & & & & \\ \gamma^{10} & \dots & & & & \end{array}$$

with  $(4i+k, 4j+l)$ -th entry clearly of the form  $\sum_P f_5^k(P) f_4^{i-k}(P) r(P) f_5(P)^l \times f_4^{j-l}(P)$ .

The algorithm producing the following computations of pairs,  $(\sum_h \sigma_{f,h} h, \sum_h \sigma_{f,h} s_h)$ , is simply a multi-dimensional row-reduction and shifting algorithm,

a version of the *Berlekamp–Massey–Sakata algorithm* discussed in Sakata (2009a, 2009b):

$\gamma^0$	$\gamma^1$
$\gamma^{13} \gamma^0$	$0 \gamma^0$
$\gamma^7 \gamma^1$	$0 \ 0$
$\gamma^0$	$\gamma^5$
$\gamma^3 \gamma^{14} \gamma^0$	$0 \ 0 \ 0 \ 0 \gamma^{12}$
$0$	$0 \ 0 \ 0$
$0$	$0 \ 0$
$0$	$0$
$0 \ \gamma^0 \ \gamma^0$	$0 \ 0 \ 0 \ 0 \ 0 \ 0$
$\gamma^0 \ \gamma^0$	$0 \ 0 \ 0 \ 0 \ 0 \ 0$
$0 \ \gamma^0 \ \gamma^4$	$0 \ 0 \ 0 \ 0 \ 0 \ 0$
$\gamma^0 \ \gamma^1$	$0 \ 0 \ 0 \ 0 \ 0 \ 0$
$\gamma^0$	$0 \ 0 \ 0 \ 0 \ 0 \ 0$
$0 \ 0 \ \gamma^3 \ \gamma^{14} \ \gamma^0$	$0 \ 0 \ 0 \ \gamma^{12}$
$0 \ 0 \ 0$	$0 \ 0$
$0 \ 0$	$0$
$0 \ \gamma^{12} \ 0 \ 0 \ \gamma^0$	$0 \ 0 \ 0 \ 0 \ 0$
$\gamma^{11} \ 0 \ 0 \ 0$	$0 \ 0 \ 0$
$0 \ 0 \ 0$	$0 \ 0$
$0 \ 0$	$0$

The *minimal, (unreduced) Gröbner basis* for the error-locator ideal  $\mathcal{I}$  can be read off from the left-hand side entries, with corresponding right-hand side zero as:

$$f_5 f_4 + f_4^2 + f_5 + f_4, \quad f_5^2 + \gamma^1 f_5 f_4 + \gamma^4 f_4^2 + \gamma^0 f_5 + \gamma^0 f_4, \quad f_4^5 + f_4^4 + \gamma^{11} f_5 + \gamma^{12} f_4$$

relative to the implicit *weighted total-degree* order induced by the pole orders. This is consistent with the syndromes computed from the received word (or those computed, given the extra assumption that the error weight and hence the rank of  $S$  is at most 6) in the sense that  $\sum_h \sigma_{f,h} s_h = 0$  for each  $\sigma_f := \sum_h \sigma_{f,h} h$  in the basis.

A factored lex basis

$$f_4(f_4 + 1)(f_4^4 + f_4^3 + 1) \cdot 1, \quad (f_4 + 1) \cdot (f_5 + f_4), \quad 1 \cdot (f_5^2 + \gamma^4 f_5 + f_4^2 + \gamma^4 f_4)$$

can be used to find the *variety* (of *error positions*)  $P_j$  with

$$(f_5(P_j), f_4(P_j)) \in \{(0, 0), (\gamma, 1), (\gamma^7, \gamma^7), (\gamma^{14}, \gamma^{14}), (\gamma^{13}, \gamma^{13}), (\gamma^{11}, \gamma^{11})\},$$

$$\{\gamma^7, \gamma^{14}, \gamma^{13}, \gamma^{11}\} \text{ being the set of roots of } x^4 + x^3 + 1.$$

The following similar example in the handbook (Høholdt et al. 1998) is originally due to Sakata: The syndrome “vector” is

$$s := \begin{pmatrix} \gamma^9 & \gamma^{14} & \gamma^5 & \gamma^7 & \gamma^2 & \gamma^5 & \gamma^0 \\ 0 & \gamma^9 & \gamma^{14} & \gamma^{12} & \gamma^5 & \gamma^5 \\ \gamma^9 & \gamma^{11} & 0 & \gamma^{12} \\ \gamma^6 & \gamma^4 & \gamma^7 \\ \hline \gamma^5 & \gamma^7 \\ \gamma^6 \end{pmatrix}$$

with

$$S = \begin{array}{ccccccccccccccccccccc} \gamma^9 & \dots & \gamma^{14} & 0 & \dots & \gamma^5 & \gamma^9 & \gamma^9 & \dots & \gamma^7 & \gamma^{14} & \gamma^{11} & \gamma^6 & \gamma^2 & \gamma^{12} & 0 & \gamma^4 & \gamma^5 & \gamma^5 & \gamma^{12} & \gamma^7 & \gamma^0 & \gamma^5 \\ \dots & & \dots & \dots & & \dots & \dots & \dots & & \dots \\ \gamma^{14} & \dots & \gamma^5 & \gamma^9 & \dots & \gamma^7 & \gamma^{14} & \gamma^{11} & \dots & \gamma^2 & \gamma^{12} & 0 & \gamma^4 & \gamma^5 & \gamma^5 & \gamma^{12} & \gamma^7 & \gamma^0 & \gamma^5 \\ 0 & \dots & \gamma^9 & \gamma^9 & \dots & \gamma^{14} & \gamma^{11} & \gamma^6 & \dots & \gamma^{12} & 0 & \gamma^4 & \gamma^5 & \gamma^5 & \gamma^{12} & \gamma^7 & \gamma^7 & \gamma^5 \\ \dots & & \dots & \dots & & \dots & \dots & \dots & & \dots \\ \gamma^5 & \dots & \gamma^7 & \gamma^{14} & \dots & \gamma^2 & \gamma^{12} & 0 & \dots & \gamma^5 & \gamma^5 & \gamma^{12} & \gamma^7 & \gamma^0 & \gamma^5 \\ \gamma^9 & \dots & \gamma^{14} & \gamma^{11} & \dots & \gamma^{12} & 0 & \gamma^4 & \dots & \gamma^5 & \gamma^{12} & \gamma^7 & \gamma^7 & \gamma^5 \\ \gamma^9 & \dots & \gamma^{11} & \gamma^6 & \dots & 0 & \gamma^4 & \gamma^5 & \dots & \gamma^{12} & \gamma^7 & \gamma^7 & \gamma^6 \\ \dots & & \dots & \dots & & \dots & \dots & \dots & & \dots & \dots & \dots & \dots \\ \gamma^7 & \dots & \gamma^2 & \gamma^{12} & \dots & \gamma^5 & \gamma^5 & \gamma^{12} & \dots & \gamma^0 & \gamma^5 \\ \gamma^{14} & \dots & \gamma^{12} & 0 & \dots & \gamma^5 & \gamma^{12} & \gamma^7 & \dots & \gamma^5 \\ \gamma^{11} & \dots & 0 & \gamma^4 & \dots & \gamma^{12} & \gamma^7 & \gamma^7 & \dots \\ \gamma^6 & \dots & \gamma^4 & \gamma^5 & \dots & \gamma^7 & \gamma^7 & \gamma^6 \\ \gamma^2 & \dots & \gamma^5 & \gamma^5 & \dots & \gamma^0 & \gamma^5 \\ \gamma^{12} & \dots & \gamma^5 & \gamma^{12} & \dots & \gamma^5 \\ 0 & \dots & \gamma^{12} & \gamma^7 & \dots \\ \gamma^4 & \dots & \gamma^7 & \gamma^7 & \dots \\ \gamma^5 & \dots & \gamma^0 & \gamma^5 \\ \gamma^5 & \dots & \gamma^5 \\ \gamma^{12} & \dots & \gamma^7 & \dots \\ \gamma^7 & \dots & \gamma^0 & \dots \\ \gamma^5 & \dots & \gamma^5 \end{array}$$

The row-reduction computations are:

$\gamma^0$	$\gamma^9$
$\gamma^5 \gamma^0$	$0 \gamma^8$
$\gamma^6 \gamma^1$	$0 \quad 0$
$\gamma^0$	$\gamma^{13}$
$\gamma^6 \gamma^{11} \gamma^0$	$0 \quad 0 \quad \gamma^5$
$\gamma^3$	$0$
$\gamma^1 \gamma^{14} \quad 0$	$0 \quad 0 \quad 0 \quad \gamma^5$
$\gamma^{13} \gamma^0$	$0 \quad 0$
	$0$
$\gamma^{11} \gamma^{13} \quad 0$	$0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$
$\gamma^{13} \gamma^{10}$	$0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$
$\gamma^0$	$0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$
	$0 \quad 0 \quad 0$
$\gamma^{10} \quad 0 \quad \gamma^5 \quad \gamma^0$	$0 \quad 0 \quad 0$
$\gamma^{14} \quad \gamma^3$	$0 \quad \gamma^5$
$0$	
$\gamma^4 \quad \gamma^7 \quad \gamma^3 \quad 0$	$0 \quad 0 \quad 0 \quad 0 \quad 0$
$\gamma^1 \quad \gamma^3 \quad \gamma^0$	$0 \quad 0 \quad 0$
$0$	$0 \quad 0$
	$0$
$\gamma^{13} \quad \gamma^3 \quad 0 \quad \gamma^3 \quad \gamma^0$	$0 \quad 0 \quad 0 \quad 0 \quad 0$
$\gamma^2 \quad \gamma^9 \quad 0 \quad \gamma^3$	$0$
$0 \quad 0$	
$0$	
$\gamma^{11} \quad 0 \quad 0 \quad 0 \quad 0 \quad \gamma^0$	$0 \quad 0 \quad 0 \quad 0 \quad 0$
$\gamma^{11} \quad 0 \quad 0 \quad 0$	$0 \quad 0 \quad 0$
$0 \quad 0 \quad 0$	$0 \quad 0$
$0 \quad 0$	$0$

so a minimal, reduced Gröbner basis for the error-locator ideal  $\mathcal{I}$  is

$$\begin{aligned} f_5^2 + \gamma^{10}f_5f_4 + \gamma^{13}f_5 + \gamma^{13}f_4 + \gamma^{11}, \\ f_5f_4^2 + \gamma^4f_5f_4 + \gamma^3f_4^2 + \gamma^1f_5 + \gamma^7f_4 + \gamma^3, \quad f_4^5\gamma^{11}f_5 + \gamma^{11}. \end{aligned}$$

A factored lex basis is

$$\begin{aligned} 1 \cdot (f_4 + 1)(f_4 + \gamma^1)(f_4 + \gamma^2)(f_4 + \gamma^5)(f_4 + \gamma^8)(f_4 + \gamma^{11})(f_4 + \gamma^{14}), \\ (f_5 + \gamma^4f_4^5 + 1) \cdot 1 \end{aligned}$$

and the variety (of error positions) is

$$(1, \gamma), (\gamma, \gamma^7), (\gamma^2, \gamma^3), (\gamma^5, \gamma^3), (\gamma^8, \gamma^3), (\gamma^{11}, \gamma^3), (\gamma^{14}, \gamma^3).$$

### 3 Interpolation to Do List Decoding for RS Codes and AG Codes

Sudan was first to suggest *list decoding* of a  $k$ -dimensional functionally-encoded RS code, by treating  $r$  and  $x$  as having weights  $k - 1$  and 1, respectively, in the polynomial ring  $\mathbb{F}[r, x]$ , interpolating the received pairs  $(r_i, x_i)$ , and finding factors (linear in the variable  $r$ ) of some resulting polynomial. For an example of such, consider the received word:

$$\underline{r} = (\gamma^{14}, 0, \gamma^6, \gamma^{11}, 0, \gamma, \gamma^3, \gamma^6, \gamma^{10}, \gamma^6, \gamma^{10}, \gamma^2, \gamma^{11}, 1, \gamma^3, \gamma^2)$$

indexed by the elements of  $\mathbb{F}_{16} := \mathbb{F}_2[\gamma]/(1 + \gamma + \gamma^4)$ :

$$\underline{x} = (1, \gamma^1, \gamma^2, \gamma^3, \gamma^4, \gamma^5, \gamma^6, \gamma^7, \gamma^8, \gamma^9, \gamma^{10}, \gamma^{11}, \gamma^{12}, \gamma^{13}, \gamma^{14}, 0)$$

for a functionally-encoded RS code with  $k = 4$  (and  $n = 16$ ). The MAGMA code<sup>1</sup> is:

```
F16<C>:=FiniteField(16);
P<r,x>:=PolynomialRing(F16,2,"weight", [3,1,3,0]);
R:=[c^14,0 ,c^6,c^11,0 ,c^1,c^3,c^6,c^10,c^6,c^10,c^2 ,c^11,c^0 ,c^3 ,c^2];
X:=[c^0 ,c^1,c^2,c^3 ,c^4,c^5,c^6,c^7,c^8 ,c^9,c^10,c^11,c^12,c^13,c^14,0];
L<u,t>:=PolynomialRing(F16,2,"grevlex");
hpl:=function(i) return hom<P->L|R[i]+u,X[i]+t>; end function;
hlp:=function(i) return hom<L->P|u-R[i],t-X[i]>; end function;
tt:=function(f) return TrailingTerm(f); end function;
f_0_0:=(P!1)@hpl(1);0,0,1,tt(f_0_0);
f_0_1:=(f_0_0*t)@hpl(1)@hpl(2);0,1,2,tt(f_0_1);
f_0_2:=(f_0_1*t)@hpl(2)@hpl(3);0,2,3,tt(f_0_2);
f_0_3:=(f_0_2*t)@hpl(3)@hpl(4);0,3,4,tt(f_0_3);
f_1_0:=(((f_0_0*u)@hpl(1)@hpl(2)-c^14/c^4*f_0_1)@hpl(2)@hpl(3)
-c^13/c^13*f_0_2)@hpl(3)@hpl(4)-c^3/c^14*f_0_3)@hpl(4)@hpl(5);
1,0,5,tt(f_1_0);
f_0_4:=((f_0_3*t)@hpl(4)@hpl(5)-c^3/c^2*f_1_0)@hpl(5)@hpl(6);
```

---

<sup>1</sup>For information on MAGMA see MAGMA et al. (2008), Bosma et al. (1997), Cannon and Playoust (1996).

```

0,4,6,tt(f_0_4);
f_1_1:=((f_1_0*t)@hlp(5)@hpl(6)-c^13/c^5*f_0_4)@hlp(6)@hpl(7);
1,1,7,tt(f_1_1);
f_0_5:=((f_0_4*t)@hlp(6)@hpl(7)-c^10/c*f_1_1)@hlp(7)@hpl(8);
0,5,8,tt(f_0_5);
f_1_2:=((f_1_1*t)@hlp(7)@hpl(8)-c^3/c*f_0_5)@hlp(8)@hpl(9);
1,2,9,tt(f_1_2);
f_0_6:=((f_0_5*t)@hlp(8)@hpl(9)-c^13/c^14*f_1_2)@hlp(9)@hpl(10);
0,6,10,tt(f_0_6);
f_1_3:=((f_1_2*t)@hlp(9)@hpl(10)-1/c^8*f_0_6)@hlp(10)@hpl(16);
1,3,16,tt(f_1_3);
f_2_0:=((((((f_1_0*u)@hlp(5)@hpl(6)-c^6/c^5*f_0_4)@hlp(6)@hpl(7)
-c^11/c*f_1_1)@hlp(7)@hpl(8)-c^7/c*f_0_5)@hlp(8)@hpl(9)
-c^9/c^14*f_1_2)@hlp(9)@hpl(10)-c/c^8*f_0_6)@hlp(10)@hpl(16)
-c^3/c^3*f_1_3)@hlp(16);2,0,Factorization(f_2_0);
f_0_7:=(f_0_6*t)@hlp(10)@hpl(11);0,7,11,tt(f_0_7);
f_1_4:=(f_1_3*t)@hlp(16);1,4,Factorization(f_1_4);
f_0_8:=(f_0_7*t)@hlp(11)@hpl(12);0,8,12,tt(f_0_8);
f_0_9:=(f_0_8*t)@hlp(12)@hpl(13);0,9,13,tt(f_0_9);
f_0_10:=(f_0_9*t)@hlp(13)@hpl(14);0,10,14,tt(f_0_10);
f_0_11:=(f_0_10*t)@hlp(14)@hpl(15);0,11,15,tt(f_0_11);
f_0_12:=((f_0_11*t)@hlp(15)@hpl(16)-c^5/c^3*f_1_3)@hpl(16);
0,12;

```

produces output (slightly edited for readability)

$$f_{2,0} = r^2 + crx^3 + c^6x^6 + c^{10}rx^2 + c^{12}x^5 + c^3rx + c^{12}x^4 + c^6r + c^7x^3 \\ + c^{11}x^2 + c^7x + c^5;$$

$$f_{1,4} := rx^4 + c^{14}x^7 + c^{13}rx^3 + c^2x^6 + c^2rx^2 + c^{10}x^5 + c^{12}rx + c^3x^4 \\ + c^{13}x^3 + c^2x^2 + x;$$

$$f_{0,12} := x^{12} + rx^8 + c^2x^{11} + c^{11}rx^7 + c^5x^{10} + c^9rx^6 + c^7x^9 + crx^5 + c^{12}x^8 \\ + c^4rx^4 + c^{11}x^7 + c^4x^6 + crx^2 + c^{12}x^5 + c^7rx + c^3x^4 + c^{12}r \\ + c^8x^3 + c^{14}$$

with weighted total degrees 6, 7, and 12 respectively. These form a Gröbner basis for the interpolating ideal. And any message with codeword at most 4 errors away from the received word must be a common root of the first two. Indeed,

$$f_{2,0} = (r + c^7x^3 + c^6x^2 + c^7x + c^2)(r + c^{14}x^3 + c^7x^2 + c^4x + c^3) \\ f_{1,4} = x(x + c^3)(x + c^4)(x + c^5)(r + c^{14}x^3 + c^7x^2 + c^4x + c^3)$$

with common root  $M(x) = \gamma^{14}x^3 + \gamma^7x^2 + \gamma^4x + \gamma^3$ ; which interpolates all the pairs except possibly the four with  $x \in \{\gamma^3, \gamma^4, \gamma^5, 0\}$ . In general, it is necessary to use interpolation to some depth  $s$  greater than 1 to get *lists* of such messages that can correspond to nearest codewords, allowing decoding beyond the standard minimum distance bound  $e < d/2$ .

To generalize this to AG codes, as first suggested by Sudan and Guruswami, let  $(f_0, f_\rho, \dots, f_m)$  be a canonical vector-space basis (of size  $k$ ) for  $\mathcal{L}(m \cdot P_\infty)$  with increasing pole sizes  $0, \rho, \dots, m$  at  $P_\infty$  (and for  $m > 2(g - 1)$ , this means  $m + 1 = k + g$ ). It may be possible to directly recover a message  $M := \sum_j m_j f_j$

from the received word  $\underline{r} = (r_1, \dots, r_n)$  by interpolation techniques. First extend the weighted total-degree ordering on  $\mathcal{L}(m \cdot P_\infty)$  given by the matrix  $A$  to  $\overline{A} := \begin{pmatrix} m & A \\ 1^T & 0 \end{pmatrix}$  to extend it to an extra variable  $r$  representing the received word.

Since these  $f$  have all their poles at  $P_\infty$ , the Laurent series at points  $P_j \neq P_\infty$  must be just *power series*. So map  $r \mapsto r(P_j) + u$  and each  $f \in \mathcal{L}(m \cdot P_\infty)$  to its power series expansion, truncated to  $t_j^i, i < s$  at  $P_j$ . *Depth-s interpolation* means finding functions with images having total degree (in  $u$  and  $t_j$ ) at least  $s$ .

If  $M(f_m, \dots, f_0)$  is encoded as a codeword  $\underline{c}$  at distance at most  $e$  from the received word  $\underline{r}$  and  $H(M(f_m, \dots, f_0), f_m, \dots, f_0) \in \mathcal{L}(((n-e)s-1) \cdot P_\infty)$ , then  $H(M(f_m, \dots, f_0), f_m, \dots, f_0)$  is a rational function with fewer than  $(n-e)s$  poles but at least that many zeros. But that means that  $H(M(f_m, \dots, f_0), f_m, \dots, f_0) \equiv 0$ , so  $r - M(f_m, \dots, f_0) | H(r, f_m, \dots, f_0)$ . Thus any  $M$  that encodes to a word at most  $e$  errors away from  $r$ , will correspond to a common linear factor  $r - M$  of all such  $H$ .

At each point  $P_j$  there are  $\binom{s+1}{2}$  “bad” trailing terms  $u^i t_j^\ell$ , with  $i + \ell < s$ , so a total  $\Delta$ -set of size  $n \binom{s+1}{2}$ . The smallest element  $H(r, f_m, \dots, f_0)$  of the interpolating Gröbner basis will be a combination of the first  $1 + n \binom{s+1}{2}$  monomials in the order described by  $\overline{A}$ . There will be at least one such good function  $H$  guaranteed, interpolating  $n - e$  of the  $n$  points, if interpolation to depth  $s$  is done, and there are more monomials  $f \in \mathcal{L}(((n-e)s-1) \cdot P_\infty)$  than  $n \binom{s+1}{2}$ , the number of elements in the  $\Delta$ -set. Simple combinatorial arguments can be used to determine the depth  $s$  needed to correct  $e$  errors using this method. And some lists will have more than one entry when  $e \geq d/2$ , as in the example below with  $d = 4$  and  $e = 2$ . (Initial papers on list decoding spent far too much time on this combinatorial aspect of the topic to the detriment of the more interesting interpolation and ideal-theoretic aspects.)

Consider the example from Høholdt and Nielsen (1999) using the Hermitian curve with  $q = 2$ . This has 8 rational points  $P_j := (x_2(P_j) : x_1(P_j) : 1)$  over  $\mathbb{F}_4 := \mathbb{F}_2[\alpha]/\langle 1 + \alpha + \alpha^2 \rangle$  other than  $P_\infty := (1 : 0 : 0)$ . Let  $f_2 := x_1$  and  $f_3 := x_2$  to reflect the respective pole sizes at  $P_\infty$ . Consider the values:

$f_2(P_j)$	0	0	1	$1\alpha$	$\alpha$	$\alpha^2$	$\alpha^2$
$f_3(P_j)$	0	1	$\alpha$	$\alpha^2$	$\alpha$	$\alpha^2$	$\alpha$
$r(P_j)$	$\alpha^2$	0	0	$\alpha^2$	0	0	0
$c_1(P_j)$	0	0	0	0	0	0	0
$c_2(P_j)$	$\alpha^2$	$\alpha^2$	$\alpha^2$	$\alpha^2$	0	0	0

for  $M_1 := 0$  and  $M_2 := \alpha^2(1 + f_2 + f_2^2)$ .  $A := \begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix}$ , so

$$\overline{A} := \begin{pmatrix} 4 & 3 & 2 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

for  $m = 4$ .

For  $n = 8$  and  $e = 2$ , the size of the  $\Delta$ -set and the number of monomials for various values of  $s$  are  $\Delta(1) = 8 > 6$ ,  $\Delta(2) = 24 > 21$ ,  $\Delta(3) = 48 > 45$ ,

$\Delta(4) = 80 > 78$ ,  $\Delta(5) = 120 = 120$ , and  $\Delta(6) = 168 < 171$ . For instance the standard monomials of weight less than 6 are  $f_0, f_2, f_3, f_2^2, r, f_3 f_2$ , and the others of weight less than 12 are

$$f_2^3, rf_2, f_3 f_2^2, rf_3, f_2^4, rf_2^2, r^2, f_3 f_2^3, rf_3 f_2, f_2^5, rf_2^3, r^2 f_2, f_3 f_2^4, rf_3 f_2^2, r^2 f_3.$$

So interpolating to depth 6 is guaranteed to produce at least  $171 - 168 = 3$  functions  $H(r, f_3, f_2)$  for list decoding.

The functions  $f_2$  and  $f_3$  with pole orders 2 and 3 respectively at the point  $P_\infty$  at infinity, have series expansions  $f_2 = x_1(P_j) + t_j$  and  $f_3 = x_2(P_j) + \sum_{\ell=0}^{\infty} (x_1(P_j)^2 t_j + x_1(P_j) t_j^2 + t_j^3)^{2^\ell}$  at each other point  $P_j$ . But with  $x_1(P_j) \in \mathbb{F}_4$  (so that  $x_1(P_j)^4 + x_1(P_j) = 0$ ) and working mod  $t_j^6$ , this reduces to  $f_3 \equiv x_2(P_j) + x_1(P_j)^2 t_j + t_j^3$ . So map  $r$  to  $r(P_j) + u$  as well, and ask for functions with images having total degree (in  $u$  and  $t_j$ ) at least 6. In this example there are  $5 > 3$   $H(r, f_3, f_2)$ 's, all with common roots  $r = 0$  and  $r = \alpha^2(1 + f_2 + f_2^2)$ , the two messages listed above.

*Remark 1* Similar ideas could work with Order Domain codes (Geil 2009). For a different approach to list-decoding, see Guerrini and Rimoldi (2009), Augot and Stepanov (2009) and Beelen and Brander (2009).

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

## References

- D. Augot and L. Pecquet, *A Hensel lifting to replace factorization in list-decoding of algebraic-geometric and Reed–Solomon codes*, IEEE Trans. on Inf. Th. **46** (2000), no. 7, 2605–2614.
- D. Augot and M. Stepanov, *A note on the generalisation of the Guruswami–Sudan list decoding algorithm to Reed–Muller codes*, this volume, 2009, pp. 359–398.
- P. Beelen and K. Brander, *Decoding folded Reed–Solomon codes using Hensel lifting*, this volume, 2009, pp. 389–394.
- W. Bosma, J. Cannon, and C. Playoust, *The MAGMA algebra system. the user language*, J. Symbolic Comput. **24** (1997), nos. 3–4, 235–265.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebräischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- J. Cannon and C. Playoust, *MAGMA: a new computer algebra system*, Euromath Bull. **2** (1996), no. 1, 113–144.
- G. L. Feng and T. R. N. Rao, *Decoding algebraic-geometric codes up to the designed minimum distance*, IEEE Trans. on Inf. Th. **39** (1993), no. 1, 37–45.
- O. Geil, *Algebraic geometry codes from order domains*, this volume, 2009, pp. 121–141.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick’s approach to recent developments*, this volume, 2009, pp. 197–218.

- T. Høholdt and R. R. Nielsen, *Decoding Hermitian codes with Sudan's algorithm*, LNCS, vol. **1719**, Springer, Berlin, 1999, pp. 260–270.
- T. Høholdt, J. H. van Lint, and R. Pellikaan, *Algebraic geometry of codes*, Handbook of coding theory, vols. I, II (V. S. Pless and W.C. Huffman, eds.), North-Holland, Amsterdam, 1998, pp. 871–961.
- D. A. Leonard, *Error-locator ideals for algebraic-geometric codes*, IEEE Trans. on Inf. Th. **41** (1995), no. 3, 819–824.
- D. A. Leonard, *A tutorial on AG code construction from a Gröbner basis perspective*, this volume, 2009, pp. 93–106.
- MAGMA, J. J. Cannon, W. Bosma (eds.), *Handbook of MAGMA functions*, edition 2.15, 2008.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- R. Roth and G. Ruckenstein, *Efficient decoding of Reed–Solomon codes beyond half the minimum distance*, IEEE Trans. on Inf. Th. **46** (2000), 246–257.
- S. Sakata, *The BMS algorithm*, this volume, 2009a, pp. 143–163.
- S. Sakata, *The BMS algorithm and decoding of AG codes*, this volume, 2009b, pp. 165–185.
- X. W. Wu and P. H. Siegel, *Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes*, IEEE Trans. on Inf. Th. **47** (2001), no. 6, 2579–2587.

# FGLM-Like Decoding: from Fitzpatrick's Approach to Recent Developments

Eleonora Guerrini and Anna Rimoldi

**Abstract** Many decoding problems in algebraic coding theory can be solved by the computation of a suitable Gröbner basis. The Gröbner basis can often be computed via the FGLM algorithm or a related algorithm (like the Buchberger–Möller algorithm). In this tutorial we describe how this has been done in the literature from a historical point of view, starting from Fitzpatrick's seminal 1995 paper, and covering recent developments for list decoding.

**Keywords** FGLM · Gröbner basis · Algebraic coding theory · Decoding · List decoding

## 1 Introduction

Algebraic decoding of error correcting codes has been studied for a long time, at least since the 1950 paper by Hamming (1950). A breakthrough came with the now-called “Berlekamp–Massey”(BM) algorithm (Massey 1969), with which any cyclic code can be decoded up to its BCH bound. Extensions to other types of corrections (erasure-and-error decoding, burst decoding, etc.) have appeared, as well as generalizations to other families of codes, such as the Algebraic-Geometry (AG) codes (Leonard 2009; Sakata 2009b). Among the numerous alternatives that have been proposed in the literature, some of them rely on Gröbner basis computations for suitable polynomial ideals or polynomial modules (Buchberger 1965, 1970, 1985, 1998, 2006). Other chapters of this book treat some of them (Mora 2009a; Mora and Orsini 2009).

In this tutorial we describe a specific family of decoding algorithms depending on the computation of a Gröbner basis, that is, we describe those algorithms where the basis can be obtained by an FGLM-like algorithm (or similar techniques, see e.g. the Buchberger–Möller algorithm Mora 2009b) in a polynomial module. We will call such algorithms “FGLM-like algorithms”.

---

E. Guerrini · A. Rimoldi  
Department of Mathematics, University of Trento, Trento, Italy

E. Guerrini  
e-mail: [guerrini@science.unitn.it](mailto:guerrini@science.unitn.it)

A. Rimoldi  
e-mail: [rimoldi@science.unitn.it](mailto:rimoldi@science.unitn.it)

First, in Sect. 2, we give some versions of the “functional approach to the computation of Gröbner bases”, which are convenient for our later discussions. We remark that this approach has been detailed in Mora (2009b) in this volume.

In Sect. 3 we provide the first instance ever published of FGLM-like algorithms, which appears in Fitzpatrick (1995).

In Sect. 4 we present some variations and improvements.

In Sect. 5 we show a first application to some AG codes.

In Sect. 6 we detail how errors and erasures can be simultaneously decoded by FGLM-like algorithms.

In Sect. 7 we explain how FGLM-like algorithms can perform list decoding of both Reed–Solomon (RS) codes and one-point AG codes. Here we present the original Sudan approach, the solution by Kötter and Vardy, the solution by O’Keeffe and Fitzpatrick. Moreover, we show how also list-decoding with soft-information can be performed with an FGLM-like algorithm.

Finally, in Sect. 8 we draw some conclusions.

## 2 Iterative Computation of Gröbner Basis

Let  $\mathbb{F}$  be a field,  $\mathcal{P} = \mathbb{F}[x_1, \dots, x_n]$  be a polynomial ring and  $L \geq 1$  be a natural number.

Recall that any term in  $\mathcal{P}^L$  is of the form  $t = \phi \mathbf{e}_k$ ,  $1 \leq k \leq L$  where  $\phi$  is a term in  $\mathcal{P}$ , and  $\{\mathbf{e}_1, \dots, \mathbf{e}_L\}$  is the canonical basis of  $\mathcal{P}^L$ , and that the set of terms in  $\mathcal{P}^L$  is denoted by  $\mathcal{T}^{(L)}$ .

Let  $S$  and  $S'$  be  $\mathcal{P}$ -submodules of a free module  $\mathcal{P}^L$  such that:

- $S' \subseteq S$
- $S' = \{a \in S \mid \lambda(a) = 0\}$ , where  $\lambda : S \rightarrow \mathbb{F}$  is an  $\mathbb{F}$ -homomorphism.

Since  $S' = \ker(\lambda)$ ,  $\forall a, b \in S \setminus S'$ , the elements

$$\left( b - \frac{\lambda(b)}{\lambda(a)}a \right) \quad \text{and} \quad x_i \left( b - \frac{\lambda(b)}{\lambda(a)}a \right)$$

are in  $S'$ . So, we have the following equality  $\lambda(x_i b) / \lambda(b) = \lambda(x_i a) / \lambda(a)$ .

It follows that, for any  $x_i$ , there exists  $\beta_i \in \mathbb{F}$  such that, for all  $c \in S$ ,

$$\lambda((x_i - \beta_i)c) = 0, \tag{1}$$

that is,  $(x_i - \beta_i)c \in S'$  (for example  $\beta_i = \lambda(x_i b) / \lambda(b)$ ).

Suppose that we know an ordered Gröbner basis  $\mathcal{G} = \{g_1, \dots, g_r\}$  of  $S$  with respect to a certain term ordering  $<$ . We want to determine a Gröbner basis  $\mathcal{G}'$  of  $S'$ . It is shown<sup>1</sup> in O’Keeffe and Fitzpatrick (2002) that such a Gröbner basis consists

---

<sup>1</sup>Of course, this is closely related to a version of the algorithm presented in Mora (2009b, Fig. 1) but we report it for later convenience. See also O’Keeffe and Fitzpatrick (2007).

of three parts:

$$\mathcal{G}' = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$$

These parts can be constructed as follows. If  $\lambda(g_h) = 0$  for all  $1 \leq h \leq r$ , then  $\mathcal{G} \subseteq S' \subseteq S$  and so  $S' = S$ . (In this case,  $\mathcal{G}_1 = \mathcal{G}'$  and  $\mathcal{G}_2 = \mathcal{G}_3 = \emptyset$ .) Otherwise, let  $\lambda(g_h) = \alpha_h$  for  $1 \leq h \leq r$  and let  $h_*$  be the least  $h$  such that  $\alpha_{h_*} \neq 0$ . We have:

- $\mathcal{G}_1 = \{g_h \mid 1 \leq h < h_*\}$ ; note  $\mathcal{G}_1 \subseteq S'$ .
- Clearly,  $g_{h_*} \notin S$  but it follows from (1) that, for any integer  $i$ ,

$$(x_i - \beta_i)g_{h_*} = \left( x_i - \frac{\lambda(x_i g_{h_*})}{\alpha_{h_*}} \right) g_{h_*} \in S'.$$

So we take

$$\mathcal{G}_2 = \left\{ (x_i - \beta_i)g_{h_*} \mid 1 \leq i \leq n \right\} \quad (\text{note } \mathcal{G}_2 \subseteq S').$$

- For  $h > h_*$ , we have  $g_h - \frac{\alpha_h}{\alpha_{h_*}} g_{h_*} \in S'$  and so we take

$$\mathcal{G}_3 = \left\{ g_h - \frac{\alpha_h}{\alpha_{h_*}} g_{h_*} \mid h_* < h \leq 1 \right\} \quad (\text{again } \mathcal{G}_3 \subseteq S').$$

In fact, by assumption of ordered Gröbner basis, it follows that the leading term of an element  $a \in S'$  is divisible by the leading term of an element of  $\mathcal{G}'$ , if  $h \neq h_*$ . Then, we may suppose that the leading term of  $g_{h_*}$  is the only leading term of the basis elements  $g_h$  that divides the leading term of  $a$  and prove that  $x_n \mathbf{T}(g_{h_*})$  also divides  $\mathbf{T}(a)$  for some  $n$ .

It is possible to apply the previous result as an incremental step of a more general situation. Let  $M_0 \supset \cdots \supset M_\ell \supset \cdots \supset M_N$  be submodules of a  $\mathcal{P}$ -module  $M$ . Let  $\theta_\ell : M_\ell \rightarrow \mathbb{F}$  be a  $\mathbb{F}$ -homomorphism such that

$$\ker(\theta_\ell) = M_{\ell+1}. \tag{2}$$

Let  $H : \mathcal{P}^L \rightarrow M$  be an  $\mathbb{F}$ -linear function such that, for any  $1 \leq i \leq n$ , there exists  $\gamma_i \in \mathbb{F}$  satisfying

$$H(x_i \mathbf{b}) = (x_i + \gamma_i)H(\mathbf{b}) \tag{3}$$

where  $\mathbf{b} = (b_1, \dots, b_L) \in \mathcal{P}^L$ . Suppose that our submodules  $S$  and  $S'$  are, respectively, the sets of elements satisfying the following congruences

$$H(\mathbf{b}) \equiv 0 \pmod{M_\ell}$$

and

$$H(\mathbf{b}) \equiv 0 \pmod{M_{\ell+1}}.$$

If we have a ordered Gröbner basis  $\mathcal{G} = \{g_1, \dots, g_r\}$  of  $S$ , the Gröbner basis  $\mathcal{G}'$  of  $S'$  will be determined as before

$$\mathcal{G}' = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$$

defining the  $\mathbb{F}$ -homomorphism  $\lambda : S \rightarrow \mathbb{F}$  by  $\lambda = \theta_\ell \circ H$ .

Observe that  $\mathcal{G}_2$  can be written as

$$\mathcal{G}_2 = \{(x_i - (\beta_i + \gamma_i))g_{h_*} \mid 1 \leq i \leq n\}.$$

Moreover, a further generalization of the previous result can be considered.

Let  $T \subset \mathbb{N}$ ,  $|T| < \infty$ . Suppose that the sequences

$$M_0^{(k)} \supset \cdots \supset M_\ell^{(k)} \supset \cdots \supset M_{N_k}^{(k)} = M^{(k)}$$

satisfy the previous conditions for any  $k \in T$ , i.e. there exists an  $\mathbb{F}$ -homomorphism such that

$$\theta_\ell^k : M_\ell^{(k)} \rightarrow \mathbb{F}, \quad \ker(\theta_\ell^k) = M_{\ell+1}^{(k)} \tag{4}$$

and  $H^k : \mathcal{P}^L \rightarrow M^{(k)}$ . We can construct  $\hat{H} : \mathcal{P}^{kL} \rightarrow M^{(1)} \times \cdots \times M^{(k)}$  such that  $\hat{H}(a_1, \dots, a_k) = (H^{(1)}(a_1), \dots, H^{(k)}(a_k))$  and consider the corresponding submodules  $S$  and  $S'$  as before.

In the next theorem we give a sufficient condition for submodules of  $\mathcal{P}^L$  which leads to the construction of a descending chain of modules and the corresponding functions.

**Theorem 1** *Let  $M$  be a submodule of  $\mathcal{P}^L$  with a finite codimension. Let  $t_0, \dots, t_{N-1}$  be the terms in  $\mathbf{N}(M)$ , ordered by some term ordering  $<'$  (not necessarily  $<$ ). Define  $M_N = M$  and  $M_\ell = \mathbb{F}t_\ell + M_{\ell+1}$ ,  $0 \leq \ell \leq N-1$ . For  $\mathbf{b} \in M_\ell$  define  $\theta_\ell(\mathbf{b})$  to be the coefficient of  $t_\ell$  in  $\mathbf{b}$ . The subsets  $M_\ell$  form a descending chain of submodules satisfying condition (2) with  $M_0 = \mathcal{P}^L$ .*

Clearly, it is possible to apply this result when  $H$  is a  $\mathcal{P}$ -homomorphism (i.e., when  $\gamma_s = 0$  for all  $s$ ), and when  $M = \mathcal{P}$  and  $M_\ell, M_{\ell+1}$  are ideals.

The well-known Vanishing Ideal Problem on a finite set of points

$$P^{(k)} = (p_1^{(k)}, \dots, p_s^{(k)}) \in \mathbb{F}^s$$

can be addressed in a natural way as a special case<sup>2</sup> of the previous general algorithm. In order to do this, we define, for  $f \in \mathcal{P}$ , an  $\mathbb{F}$ -linear function

$$H^{(k)}(f) = f(x_1 - p_1^{(k)}, \dots, x_s - p_s^{(k)})$$

such that  $H^{(k)}(x_i f) = (x_i - p_i^{(k)})H^{(k)}(f)$ , where  $i = 1, \dots, s$ .

---

<sup>2</sup>Again, also this is reported fully in Mora (2009b).

The ideal  $I = \{f \in \mathcal{P} \mid f(P^k) = 0, k = 1, \dots, s\}$  is the solution set of a sequence of congruences

$$H^{(k)}(f) \equiv 0 \pmod{\langle x_1, \dots, x_s \rangle},$$

where  $k = 1, \dots, p$ .

In the rest of the paper we will see that many solution sets  $S$  and  $S'$  arising for applications require particular conditions on their degrees.

### 3 The Key Equation for Alternant Codes

A well-known problem in decoding  $t$ -error correcting alternant codes consists of the determination of  $\omega, \sigma \in \mathcal{P} = \mathbb{F}[x]$  satisfying the key equation

$$\omega \equiv \sigma h \pmod{x^{2t}} \quad (5)$$

where  $h$  is the syndrome polynomial and  $\sigma$  and  $\omega$  represent the error locator and error evaluation polynomials, respectively. There are some constraints on the degree of these polynomials, that is,  $\deg(h) \leq 2t - 1$ ,  $\deg(\omega) < \deg(\sigma) \leq t$  and  $\gcd(\omega, \sigma) = 1$ . To deal with this problem, in Fitzpatrick (1995) proposed two techniques for finding particular solutions of the congruence

$$a \equiv bg \pmod{x^n} \quad (6)$$

for a given polynomial  $g$  such that  $\deg(g) \leq n - 1$ , satisfying the following degree constraints:

$$\deg(a) \leq l, \quad \deg(b) \leq m, \quad l + m < n, \quad \gcd(a, b) = 1. \quad (7)$$

The main point is to identify the required solution by its degree constraints and then to define the term order so that the required solution becomes the minimal element in the solution module with respect to this order. Since this minimal element must appear in a Gröbner basis with respect to this order, it can be determined by finding a Gröbner basis of the solution module.<sup>3</sup> It is easy to show that the solution set  $M$  of the congruence (6) is a submodule of  $\mathcal{P}^2$ . Thus, it is possible to apply the algorithm presented in the previous section in order to solve the sequence of partial problems  $a \equiv bg \pmod{x^k}$  (where  $0 \leq k \leq n$ ), determining Gröbner bases of the modules  $M_0, \dots, M_n$  with respect to the special term ordering  $<_r$  defined as:

$$(x^i, 0) <_r (x^{i'}, 0) \quad \text{if } i < i' \\ (0, x^j) <_r (0, x^{j'}) \quad \text{if } j < j'$$

---

<sup>3</sup>In Flynn–Fitzpatrick (1992) the idea is more general: the required solution is not necessarily minimal but it is forced to lie in a Gröbner basis anyway by good choice of  $<$ .

$$(x^i, 0) <_r (0, x^j) \quad \text{if } i \leq j + r.$$

As proved in Fitzpatrick and Jennings (1998), using this technique for decoding alternant codes rather than the B-M algorithm, the designer has the same computational complexity. The number of multiplications used by this algorithm is at most  $2t^2$ , instead of the  $2t^2 - 2t + 1$  multiplications of the B-M algorithm. Both algorithms use  $2t$  divisions but it is trivial to convert the former algorithm into a division free algorithm.

The second technique, proposed in Fitzpatrick (1997) to solving the key equation (6) consists of a direct application of the FGLM algorithm. Since it is easy to show that  $\mathcal{G} = \{(g, 1), (x^n, 0)\}$  is a Gröbner basis of  $M$  with respect to the order  $<_{\deg(g)}$ , using FGLM we can determine a Gröbner basis  $\mathcal{G}'$  with respect to the term ordering  $<_r$ . The computation is equivalent to the Euclidean algorithm technique for decoding alternant codes.

## 4 Variations

Remark that, if, with the notation of Sect. 2, we are given  $\rho$  functionals  $\lambda_i : S \rightarrow \mathbb{F}$ ,  $1 \leq i \leq \rho$ , and we denote  $S^* := \{a \in S | \lambda_i(a) = 0, 1 \leq i \leq \rho\}$  necessarily  $\dim_{\mathbb{F}}(S^*) \leq \rho$  and, given any  $\rho + 1$  elements  $a_0, a_1, \dots, a_\rho \in S$ , necessarily there is an  $\mathbb{F}$ -linear relation  $\sum_{i=0}^\rho c_i a_i \in S^*$ .

The specialization of this general remark performed by Fitzpatrick in 1995 to the key equation perfectly illustrates the general scheme<sup>4</sup>: denoting  $S := \mathcal{P}^2$ , given the  $\rho := n$  functionals which impose (6) and the  $\lambda + \mu + 2 = \rho + 1 = n + 1$  elements

$$\begin{aligned} a_0 &:= (1, 0), a_1 := (x, 0), \dots, a_\lambda := (x^\lambda, 0), \\ a_{\lambda+1} &:= (0, 1), a_{\lambda+2} := (0, x), \dots, a_{\lambda+\mu+1} := (0, x^\mu), \end{aligned}$$

we obtain

$$S^* \ni \sum_{i=0}^\rho c_i a_i = \sum_{i=0}^\lambda c_i (x^i, 0) + \sum_{i=0}^\mu c_{\lambda+1+i} (0, x^i),$$

i.e., setting  $a := \sum_{i=0}^\lambda c_i x^i$  and  $b := \sum_{i=0}^\mu c_{\lambda+1+i} x^i$ , the relation (6) with the degree constraints (7).

The same scheme, of course, can be applied to more general setting. Decoding multidimensional cyclic codes, for instance, can be obtained via the following theorem.

**Theorem 2** (Little et al. 2003) *Denote with  $\tau(p)$  the total degree of a generic polynomial  $p$ , assume that  $|N| + |D| \geq |\mathbf{N}(I)| + 1$ , with  $N = \{\mathbf{x}^\alpha : |\alpha| \leq t_1\}$  and  $D = \{\mathbf{x}^\beta : |\beta| \leq t_2\}$ . Then a Gröbner basis for  $M$  w.r.t.  $<_\tau$  contains an element  $(a, b)$  such that  $\tau(a) < \tau(b) \leq t_2$ , if such a solution exists in  $M$ .*

---

<sup>4</sup>As well as the specialization performed by the FGLM Algorithm.

As Little et al. point out, these hypotheses are still limiting. However, it is sufficient, as it was suggested in Farr and Gao (2005), to consider any term ordering  $\prec$  on  $S := \mathcal{P}^2$ , enumerate the elements in  $\mathcal{T}^{(2)}$  as  $\mathbf{x}^{\alpha_1} \mathbf{e}_{i_1}, \mathbf{x}^{\alpha_2} \mathbf{e}_{i_2}, \dots$ , fix a value  $t$  and set  $J_1 := \{j \leq t + 1 : i_j = 1\}$  and  $J_2 := \{j \leq t + 1 : i_j = 2\}$  in order to obtain the following theorem.

**Theorem 3** (Farr and Gao 2005) *For any  $f$  in  $\mathcal{P}$  there is a pair of polynomials  $(a, b) \in \mathcal{P}^2$ , not both zero, satisfying*

$$bf \equiv a \pmod{I}, \quad a = \sum_{i \in J_1} a_i \mathbf{x}^{\alpha_i}, \quad b = \sum_{i \in J_2} b_i \mathbf{x}^{\alpha_i}.$$

Furthermore,  $(a, b)$  is contained in the reduced Gröbner basis of  $M$ .

As shown Farr and Gao (2005), we can allow  $N := \{\mathbf{x}^{\alpha_i} \mid i \in J_1\}$  and  $D := \{\mathbf{x}^{\alpha_i} \mid i \in J_2\}$  to take a much wider variety of shapes than has been studied previously (e.g., the triangular ones and the rectangular shapes), simply by playing freely with  $\prec$  and its underlining term-ordering  $<$  on  $\mathcal{T}$ .

## 5 Some Applications to AG Codes

An interesting application arises from a method of decoding certain AG codes proposed in Shen's (1992) thesis (see also Porter et al. 1992). Without going into the details of the algebraic geometric background, it is possible to interpret their technique as follows.

Let  $f \in \mathbb{F}[x, y]$  such that the affine curve  $f = 0$  is regular. For a certain geometric Goppa code  $C$  defined from  $f$  it is shown in Porter et al. (1992) that there exists a polynomial  $g$  such that the decoding problem for  $C$  is equivalent to the determination of a solution  $(a, b)$  of the congruence

$$a \equiv bh \pmod{I}, \tag{8}$$

where  $I$  is the zero dimensional ideal  $I = \langle f, g \rangle$  of  $\mathbb{F}[x, y]$  in which  $\deg(b)$  is minimal subject to  $\deg(a) - \deg(b) \leq s$  for a fixed positive integer  $s$  (associated with the curve), and  $h$  is a given polynomial in normal form relative to a Gröbner basis of  $I$ .

To solve this problem with the techniques presented in Sect. 2, we refer to Fitzpatrick (1997) defining a term order  $<_2$  in  $\mathcal{T}^{(2)}$  as follows. Let  $<$  be the total degree lexicographic order in  $\mathcal{T}$  with  $x < y$  and let  $<_u$  be the term order in  $\mathcal{T}$  defined in Mora (2009a). Let  $u = (u_1, \dots, u_n) \in \mathbb{N}^n$ , define  $\phi \mathbf{e}_i <_2 \psi \mathbf{e}_i$ , for  $i = 1, 2$ , if and only if  $\phi <_u \psi$ , and define  $\phi \mathbf{e}_1 <_2 \psi \mathbf{e}_2$  if either  $\deg(\psi) - \deg(\phi) \leq s$  or  $(\deg(\psi) - \deg(\phi)) = s$  and  $\phi <_u \psi$  and  $\psi \mathbf{e}_2 <_2 \phi \mathbf{e}_1$  otherwise.

## 6 Errors and Erasures for Alternant Codes

In Sect. 2 we have seen how the theory of Gröbner bases of modules could be applied to the solution of the key equation for BCH codes to derive an algorithm whose computational complexity is the same as the Berlekamp–Massey algorithm, but which has a more regular structure and certain hardware advantage. In Fitzpatrick (1995) it is also shown how the algorithm could be adapted to solve the errors-and-erasures problem using the modified syndrome polynomial. However it is well-known that this approach can be improved in the case of the Berlekamp–Massey algorithm by initializing it with the erasure locator polynomial, thereby removing the need to calculate the modified syndromes. In this section we describe how it is possible to apply this corresponding simplification in the Fitzpatrick’s algorithm. For this section, we refer to Fitzpatrick (1999).

We consider the following more general interpolation problem: given  $c, g \in \mathcal{P}$  and  $n$  a non-negative integer, determine a pair  $(a, b) \in \mathcal{P}^2$  such that  $c|b$ , satisfy both (6) and the degree constraint

$$\deg(a) \leq l, \quad \deg(b) \leq m, \quad l + m < n + \deg(c). \quad (9)$$

### 6.1 Errors and Erasures

We recall the classical equations for errors and erasures. We consider a  $t$ -error correcting BCH code  $C$  of length  $n$  over the field  $\mathbb{F}$ , and denote the error and erasure polynomials by

$$e(x) = \sum_{i \in I} e_i x^i, \quad E(x) = \sum_{j \in J} E_j x^j \quad (10)$$

respectively, where  $I, J$  are disjoint subsets of  $\{1, \dots, n-1\}$ . We may assume that  $\deg(e) = |I|$ ,  $\deg(E) = |J|$  satisfy  $2\deg(e) + \deg(E) \leq 2t$ . The corresponding error locator and erasure locator polynomials are

$$\sigma(x) = \sum_{i \in I} (1 - \alpha^i x), \quad \Sigma(x) = \sum_{j \in J} (1 - \alpha^j x), \quad (11)$$

where  $\alpha$  is a primitive  $n$ -th root of unity in some extension of  $\mathbb{F}$  and, for simplicity, we assume that the code is defined by the roots  $\alpha^k$ ,  $k = 1, \dots, 2t$ . The syndrome polynomial is

$$\begin{aligned} S(x) &= \sum_{k=0}^{2t-1} (e(\alpha^{k+1}) + E(\alpha^{k+1})) x^k \\ &= \sum_{k=0}^{2t-1} \left[ \sum_{i \in I} e_i \alpha^{i(k+1)} + \sum_{j \in J} E_j \alpha^{j(k+1)} \right] x^k \end{aligned}$$

$$= \sum_{i \in I} \frac{e_i \alpha^i}{1 - \alpha^i x} + \sum_{j \in J} \frac{E_j \alpha^j}{1 - \alpha^j x} \bmod x^{2t}.$$

Multiplying by the product  $\sigma(x)\Sigma(x)$  we obtain

$$\begin{aligned} \sigma(x)\Sigma(x)S(x) &\equiv \Sigma(x) \sum_{i \in I} e_i \alpha^i \prod_{\substack{i' \in I \\ i' \neq i}} (1 - \alpha^{i'} x) \\ &\quad + \sigma(x) \sum_{j \in J} E_j \alpha^j \prod_{\substack{j' \in J \\ j' \neq j}} (1 - \alpha^{j'} x) \bmod x^{2t}. \end{aligned} \quad (12)$$

The right hand side of this congruence will be denoted  $\Omega(x)$  and we note that it is relatively prime to  $\sigma\Sigma$  and its degree is at most  $\deg(e) + \deg(E) - 1$ . Hence, taking  $g = S, n = 2t, a = \Omega, b = \sigma\Sigma, c = \Sigma$ , we have

$$2\deg(b) = 2\deg(e) + 2\deg(E) \leq 2t + \deg(E)$$

and

$$\deg(b) \leq \left\lfloor \frac{n + \deg(c)}{2} \right\rfloor.$$

Also,

$$2\deg(a) \leq 2\deg(e) + 2\deg(E) - 2 \leq 2t + \deg(E) - 2$$

and hence

$$\deg(a) \leq \left\lfloor \frac{n + \deg(c)}{2} \right\rfloor - 1.$$

Taking

$$l = \left\lfloor \frac{n + \deg(c)}{2} \right\rfloor, \quad m = \left\lfloor \frac{n + \deg(c)}{2} \right\rfloor - 1, \quad (13)$$

we have  $l + m < 2t + \deg(c)$ , so conditions (9) hold and the errors-and-erasures decoding problem is a special case of (6) with  $c|b$  and conditions (9).

Notice that the errors-only case is recovered when  $c$  is a constant.

## 6.2 Solutions Using Gröbner Bases

We define the sequence  $\mathcal{P}^2 = M_0, M_1, \dots, M_n = M$  of submodules of  $\mathcal{P}^2$ , where  $M_k$  is the set of all pairs  $(a, b) \in \mathcal{P}^2$  satisfying

$$a \equiv bg \bmod x^k \text{ and } c|b. \quad (14)$$

We want to determine a Gröbner basis of  $M_n = M$  relative to  $<_r$  and the main point is whether the sequence of Gröbner bases of the  $M_k$  can be initialized at some point other than  $k = 0$ . The next result shows that the polynomial  $c$  can be used to define the initialization. Let  $\bar{f}$  denote the reduction of  $f$  modulo  $x^k$  (where the value of  $k$  will be clear from the context) and define

$$\mathcal{B}_k = \begin{cases} \{(\bar{c}g, c), (x^k, 0)\} & \text{if } k \geq 0 \\ \{(0, c), (1, 0)\} & \text{if } k < 0 \end{cases} \quad (15)$$

**Lemma 1** *Let  $k = \deg(c) + r + 1$ . Then  $\mathcal{B}_k$  is a Gröbner basis with respect to  $<_r$  of  $M_k$  if  $k \geq 0$ , and of  $M_0$  if  $k \leq 0$ .*

It is now clear that, keeping to the conventions of Fitzpatrick (1995), the set

$$\mathcal{B}_{\deg(c)+r+1} = \{(x^{\deg(c)+r+1}, 0), (\bar{c}g, c)\}$$

in which the first element has leading term on the left and the second element is minimal, can be used to initialize the algorithm in Fitzpatrick (1995). Also, as in the Berlekamp–Massey algorithm, we can avoid computing the left hand components. Thus the algorithm can be initialized with the pair  $(0, c)$  and continued through  $n - \deg(c) - r - 1$  iterations. Furthermore, the previous lemma allows us to avoid initialization of the parameter  $d$  by adopting the convention that coefficients of negative powers of  $x$  are zero when they arise.

---

### Algorithm 1 Fitzpatrick (1999)

---

**Require:**  $c, g, n, r$

**Ensure:**  $b_i$

```

 $b_0, k \leftarrow 0, b_1 \leftarrow c, \alpha_0 \leftarrow -1, i, i', j, d \leftarrow 1$ 
while  $k < n - \deg(c) - r - 1$  do
     $\alpha_j \leftarrow (b_j g)_{k+\deg(c)+r+1}$ 
     $k \leftarrow k + 1$ 
    if  $\alpha_i \neq 0$  then
         $b_{i'} \leftarrow b_{i'} - \frac{a_{i'}}{a_i} b_i$ 
         $b_i \leftarrow x b_i$ 
         $j \leftarrow i', d \leftarrow d - 1$ 
        if  $d = 0$  then
             $i \leftarrow i', d \leftarrow 1$ 
        end if
    else
         $b_{i'} \leftarrow x b_{i'}$ 
         $j \leftarrow i, d \leftarrow d + 1$ 
    end if
end while

```

---

**Table 1** Intermediate steps in locator computation

$k$	$b_0$	$b_1$	$\alpha_0$	$\alpha_1$	$i$	$j$	$d$
0	0	$\alpha^{13}x^2 + \alpha^9x + 1$	1	$\alpha^3$	1	1	1
1	$\alpha^{10}x^2 + \alpha^6x + \alpha^{12}$	$\alpha^{13}x^3 + \alpha^9x^2 + x$	$\alpha^5$	$\alpha^3$	0	0	1
2	$\alpha^{10}x^3 + \alpha^6x^2 + \alpha^{12}x$	$\alpha^{13}x^3 + \alpha^{12}x^2 + \alpha x + \alpha^{10}$	$\alpha^5$	$\alpha^4$	1	1	1
3	$\alpha^{11}x^3 + x^2 + \alpha^7x + \alpha^{11}$	$\alpha^{13}x^4 + \alpha^{12}x^3 + \alpha x^2 + \alpha^{10}x$	$\alpha^9$	$\alpha^4$	0	0	1
out	$\alpha^{11}x^4 + x^3 + \alpha^7x^2 + \alpha^{11}x$	$\alpha^{13}x^4 + \alpha^4x^3 + \alpha^8x^2 + \alpha^4x + \alpha^6$	$\alpha^9$	$\alpha^4$	1	1	1

*Example 1* Consider the  $(15, 9)$  RS code over  $\mathbb{F}_{16}$  defined by the roots  $\alpha^k$ ,  $k = 1, \dots, 6$  where  $\alpha$  is a primitive field element satisfying  $\alpha^4 + \alpha + 1 = 0$ . Suppose that the erasure locator polynomial is  $\sum = \alpha^{13}x^2 + \alpha^9x + 1$ , corresponding to erasures in locations  $\alpha^2, \alpha^{11}$ , and the syndrome polynomial is  $g = \alpha^{14}x^5 + x^4 + \alpha^2x^3 + \alpha^5x^2 + \alpha^5x + \alpha^{12}$ . From (13) the value of  $r$  is  $-1$ , so we iterate from  $k = 0$  to  $k = 3$ . The output is given in Table 1

Thus

$$\sigma \Sigma = \alpha^{-6}(\alpha^{13}x^4 + \alpha^4x^3 + \alpha^8x^2 + \alpha^4x + \alpha^6) \quad (16)$$

$$= \alpha^7x^4 + \alpha^{13}x^3 + \alpha^2x^2 + \alpha^{13}x + 1 \quad (17)$$

from which we derive the error locations  $\alpha, \alpha^8$ . The erasure and error values can be obtained from (12) by the usual argument.

## 7 List Decoding Problem

When the number of errors in a received word exceeds half the minimum distance of the code, there may be more than one codeword consistent with the error vector. As an alternative to finding a particular codeword, the decoder may attempt to generate a list of consistent codewords, and then choose among these according to some criterion. Ideally, such a list would be short and have only one element on most occasions. This is known as *list decoding*.

Sudan (1997) proposed a polynomial-time technique (for low rate codes) which performs list decoding of RS codes. The essential idea consists of two steps:

1. use a received word to create a set of points and construct a 2-variable polynomial interpolating these points,
2. factorize the computed polynomial to yield the required list of codewords.

This approach was extended in 1999 to RS codes of all rates by Guruswami and Sudan (1999). They also applied it to 1-point AG codes. An improved interpolation step was presented by Høholdt and Nielsen (2000); Sakata also provided an interpolation improvement by using a Gröbner basis approach and the Berlekamp–Massey–Sakata algorithm (Sakata 2009a, 2009b).

Kötter and Vardy (2003) used an approach similar to Guruswami–Sudan for the soft-decision decoding of RS and AG codes. Recently, Pellikaan and Wu (2004) showed that Reed Muller codes of certain orders can be described by 1-point AG codes and list decoding can be performed using AG list decoding techniques.

O’Keeffe and Fitzpatrick (2002, 2007) applied their general Gröbner basis algorithm to solving constrained interpolation problems in hard and soft-decision decoding of RS codes. Moreover, they addressed the interpolation problems for 1-point AG codes arising in Guruswami–Sudan and in Kötter–Vardy, starting from their general algorithm.

## 7.1 Sudan’s Approach

A RS code of dimension  $K$  and length  $N = q - 1$  (Augot et al. 2009) can viewed as the evaluation of polynomials in  $\mathbb{F}_q[x]$  with degree less than  $K$  at the non-zero elements  $\{\alpha_1, \dots, \alpha_N\} \subseteq \mathbb{F}_q$ , that is, the RS code is

$$\mathcal{C}_q(N, K) = \{(f(\alpha_1), \dots, f(\alpha_N)) \mid f \in \mathbb{F}_q[x], \deg(f) < K\}.$$

The minimum distance of  $\mathcal{C}_q(N, K)$  is  $N - K + 1$ . The size of the base field is a limit on the length of such a code.

Let the transmitted codeword correspond to  $f \in \mathbb{F}_q[x]$  and the received word be  $(\beta_1, \dots, \beta_N)$ . If  $\tau$  errors occurred then  $\tau = |\{i \mid f(\alpha_i) \neq \beta_i\}|$ . When  $\tau \leq \lfloor \frac{N-K}{2} \rfloor$  a unique closest codeword can be found; otherwise uniqueness cannot be guaranteed.

The approaches to list decoding of RS codes, based on Sudan’s method, seek a polynomial  $Q(x, y) \in \mathbb{F}_q[x, y]$ . Polynomial  $Q$  interpolates a set of points  $(\alpha_i, \beta_i)$ . The univariate polynomials corresponding to the candidate codewords are among the factors  $y - f(x)$  of  $Q$ . The conditions on the original Sudan algorithm confine its applicability to low rate codes. Guruswami and Sudan (1999) extended the algorithm to codes of all rates and required that the polynomial  $Q(x, y)$  have some derivatives equal to zero (i.e. have multiple zeros) at the interpolating points. We give a sketch of their algorithm.

1. Interpolation step. Given the received vector  $(\beta_1, \dots, \beta_n)$ , the decoder constructs a two variable polynomial

$$Q(x, y) = \sum_{i,j} a_{i,j} x^i y^j$$

such that  $Q$  has a given multiplicity  $m^5$  at any point  $(\alpha_i, \beta_i)$  and for which the  $(1, K - 1)$  weighted degree of  $Q(x, y)$  is as small as possible.

---

<sup>5</sup>We say that  $Q(x, y) = \sum_{i,j} a_{i,j} x^i y^j \in F[x, y]$  has a zero of multiplicity (or order)  $m$  at  $(0, 0)$  if  $Q(x, y)$  involves no term of total degree less than  $m$ , i.e.,  $a_{i,j} = 0$  if  $i + j < m$ .

2. Factorization step. The decoder then finds all factors of  $Q(x, y)$  of the form  $y - f(x)$ , where  $f(x)$  is a polynomial of degree  $K - 1$  or less. Let

$$L = \{f_1(x), \dots, f_{|L|}(x)\}$$

be the list of polynomials produced by this step. The polynomials (codewords)  $f(x) \in L$  are of three possible types: the transmitted codeword, codewords with Hamming distance  $\leq \tau_m$  from the received vector or codewords with distance  $> \tau_m$  from the received vector.

We recall the following (over any  $\mathbb{F}$ ). Let  $<$  be a fixed monomial ordering:

$$1 = \phi_0(x, y) < \phi_1(x, y) < \phi_2(x, y) < \dots$$

With respect to  $<$  any nonzero polynomial in  $\mathbb{F}[x, y]$  can be expressed uniquely as

$$Q(x, y) = \sum_{j=1}^J a_j \phi_j(x, y)$$

for suitable coefficients  $a_j \in \mathbb{F}$  with  $a_J \neq 0$ . Monomial  $\phi_J$  is the leading monomial of  $Q(x, y)$ .

As proposed in McEliece (2003), we can rewrite the Guruswami–Sudan algorithm. Given an  $(N, K)$  RS code over  $\mathbb{F}$ , with support set  $(\alpha_1, \dots, \alpha_N)$ , and a positive integer  $m$ , the Guruswami–Sudan decoder accepts a vector  $\beta = (\beta_1, \dots, \beta_N) \in \mathbb{F}^N$  in input, and produces a list of polynomials  $\{f_1, \dots, f_L\}$  as output. In particular, it constructs a non zero two-variable polynomial

$$Q(x, y) = \sum_{j=0}^J a_j \phi_j(x, y)$$

where  $\phi_0 < \phi_1 < \dots$  is  $(1, v)$ -revlex monomial order,<sup>6</sup> such that  $Q(x, y)$  has a zero of order  $m$  at each of the  $N$  points  $(\alpha_i, \beta_i)$  for  $i = 1, \dots, N$ . The output of the algorithm is the list of  $y$ -roots of  $Q(x, y)$ , i.e.

$$L = \{f(x) \in F[x] : (y - f(x)) \mid Q(x, y)\}.$$

It is easy to prove the following theorem:

**Theorem 4** *The list  $L$  contains every polynomial of degree  $\leq v$  such that*

$$|\{i \mid f(\alpha_i) = \beta_i\}| \geq \max\{\deg_{1,v} \phi_i(x, y) : i = 0, \dots, C\}.$$

---

<sup>6</sup>I.e. the refinement of the weight  $(1, v)$  with the revlex ordering induced by  $x < y$ .

Furthermore, the number of polynomials in the list is at most

$$L_m = \left\lfloor \sqrt{\frac{N}{v}m(m+1) + \left(\frac{v+2}{2v}\right)^2} - \frac{v+2}{2v} \right\rfloor.$$

The construction of AG codes is analogous to that of RS codes. It is based on evaluating rational functions at points on algebraic curves. Practical decoding of AG codes was introduced in Justesen et al. (1989), and faster decoding was presented in Justesen et al. (1992). We will only consider the special case of 1-point (AG) codes. Drawing on Algebraic Geometry, longer codes can be created by analogy with the description of RS codes above. Rational functions on a curve are evaluated at  $\mathbb{F}_q$ -rational points of that curve, where the pole order of these functions at a single point plays the role corresponding to polynomial degree in the case of RS codes.

Let  $\chi$  be an absolutely irreducible curve of genus  $g$  over  $\mathbb{F}_q$ . Denote  $n+1$   $\mathbb{F}_q$ -rational points on  $\chi$  by  $P_1, \dots, P_n, P_\infty$  (where  $P_\infty$  is the point at infinity). Define as usual  $\mathcal{L}(l(P_\infty))$  to be the set of rational functions in  $\chi$  at  $P_\infty$  whose pole order at  $P_\infty$  is at most  $l$ . For  $2g-1 \leq k < n$ , a 1-point code  $\mathcal{C}_\chi(k, P_\infty)$  can be defined as the  $\mathbb{F}_q$ -vector space

$$\{(f(P_1), \dots, f(P_n)) \mid f \in \mathcal{L}(kP_\infty)\}. \quad (18)$$

For any  $l \geq 2g-1$  there are functions  $\phi_{1,\infty}, \dots, \phi_{l-g+1,\infty}$ , with increasing pole orders, that form a (vector space) basis of  $\mathcal{L}(lP_\infty)$ . Thus, a code defined by (18) has length  $N=n$  and dimension  $K=k-g+1$ .

It is well-known (Kötter and Vardy 2000; Kötter and Vardy 2003; Guruswami and Sudan 1999; Høholdt and Nielsen 2000) that for any of the points  $P_i \neq P_\infty$ , there is also a basis for  $\mathcal{L}(lP_\infty)$  of functions  $\phi_{1,i}, \dots, \phi_{l-g+1,i}$  with increasing zero order at  $P_i$ . There is a set of basis conversion constants

$$\{\delta_{i,j_2,j_3} \in \mathbb{F}_q \mid i \in [n], j_2, j_3 \in [l-g+1]\}$$

such that for any  $i, j_2$

$$\phi_{j_2,\infty} = \sum_{j_3} \delta_{i,j_2,j_3} \phi_{j_3,i}.$$

These observations lead to the extension of Sudan's algorithm to 1-point codes (Guruswami and Sudan 1999), which we do not detail.

## 7.2 Improvements on the Interpolation Steps for the RS Codes

There have been some improvements on the interpolation steps in RS list decoding using a module approach:

1. Kötter (1996) present an algorithmic which outputs the minimal element in a specified submodule;

2. Kötter and Vardy (2000, 2003) give a similar algorithm for the more general case when soft-information are available;
3. McEliece (2003) rewrites the two previous algorithm showing that they actually compute a basis for the submodule and recognize some Gröbner aspects (El-Khamy and McEliece 2005);
4. O'Keeffe and Fitzpatrick (2002) give a Gröbner computation an a similar module.

Let  $\mathbb{F}_L[x, y]$  denote the set of polynomials in  $\mathbb{F}[x, y]$  whose  $y$ -degree is  $\leq L$ , i.e. those of the form

$$Q(x, y) = \sum_{k=0}^L q_k(x) y^k$$

where each  $q_k(x) \in \mathbb{F}[x]$ . We note that  $\mathbb{F}_L[x, y]$  is an  $\mathbb{F}[x]$ -module, since if  $Q(x, y) \in \mathbb{F}_L[x, y]$ , and  $f(x) \in \mathbb{F}[x]$ , then  $f(x)Q(x, y) \in \mathbb{F}_L[x, y]$ , as well.

Let  $D_1, \dots, D_C$  be  $C$  linear functionals defined on  $\mathbb{F}_L[x, y]$  and let  $K_1, \dots, K_C$  be the corresponding kernels, i.e.

$$K_i = \{Q(x, y) \in \mathbb{F}_L[x, y] : D_i(Q) = 0\}.$$

The cumulative kernels  $\bar{K}_0, \dots, \bar{K}_C$  are defined as follows:  $\bar{K}_0 = \mathbb{F}_L[x, y]$  and for  $i = 1, \dots, C$ ,

$$\begin{aligned} \bar{K}_i &= \bar{K}_{i-1} \cap K_i = K_1 \cap \cdots \cap K_i \\ &= \{Q(x, y) \in \mathbb{F}_L[x, y] : D_1(Q) = \cdots = D_i(Q) = 0\}. \end{aligned}$$

The key point here is that any  $K_i$  is a  $\mathbb{F}[x, y]$ -module. Kötter and Vardy decode by iterative computation of Gröbner basis for sub-modules  $\bar{K}_i$ , as detailed in Algorithm 3. Algorithm 3 is seen by McEliece as an instance of a more general algorithm (Algorithm 2 already present in Kötter's thesis). However, nor Kötter, nor Vardy, nor McEliece notice that the two algorithms are actually computing a Gröbner basis (see the following remark).

*Remark 1* Algorithm 2 is related to O'K-F. In fact we can rewrite

$$g_j := \Delta x f - D_i(xf) f$$

where  $f = g_{h^*}$  and  $\Delta = \alpha_{h_*}$ , which gives

$$\begin{aligned} \alpha_{h_*}(x g_{h^*}) - \lambda(x g_{h^*}) g_{h^*} &= \alpha_{h_*}(x g_{h^*}) - \beta_i \lambda(g_{h^*}) g_{h^*} \\ &= \beta_i \alpha_{h^*} = \alpha_{h^*}(x - \beta_i) g_{h_*}. \end{aligned}$$

*Remark 2* We would like also to report on a recent result by Lee and O'Sullivan (2008). They consider the same submodule and the same term-order used by Kötter, but they do not apply an FGLM-like algorithm to compute the Gröbner basis.

---

**Algorithm 2** Kötter algorithm. (Complexity  $O(C^2)$ )

---

**Require:**  $L, (D_i)_{i=1}^C$ , arbitrary monomial order

```

for  $j = 0$  to  $L$  do
     $g_j := y^j$ 
end for
for  $i = 1$  to  $C$  do
    for  $j = 0$  to  $L$  do
         $\Delta_j := D_i(g_j)$ 
         $J := \{j : \Delta_j \neq 0\}$ 
    end for
    if  $j \neq 0$  then
         $j^* := \operatorname{argmin}\{\Delta_j : \Delta_j \neq 0\}$ 
         $f := g_{j^*}; \Delta := \Delta_{j^*}$ 
        for  $j \in J$  do
            if  $j \neq j^*$  then
                 $g_j := \Delta g_j - \Delta_j f$ 
            else
                 $g_j := \Delta x f - D_i(xf) f$ 
            end if
        end for
    end if
end for
end for
 $Q_0(x, y) := \min_{j=0}^L \{g_j(x, y)\}$ 
```

---

Instead, they apply directly a specialized form of the Buchberger algorithm.

According to the authors, the complexity of their approach is essentially the same as that of the FGLM-like technique.

### 7.3 Method in Sect. 12.2 Applied to List Decoding for AG Codes

For this subsection we refer to O’Keeffe and Fitzpatrick (2002) (but see also Fitzpatrick and O’Keeffe 2002).

Let  $R = \bigcup_{l=0}^{\infty} \mathcal{L}(lP_{\infty})$  and let  $z$  be transcendental over  $\mathbb{F}_q(\chi)$ . Consider the polynomial  $Q \in R[z]$  where

$$Q(z) = \sum_{j_1=0}^b \sum_{j_2=1}^a q_{j_1, j_2} z^{j_1} \phi_{j_2, \infty}.$$

Here the expansion of the polynomial with respect to the zero basis at  $P_i$  plays the role of the shifting of the first indeterminate by  $x_i$  in the RS case. By associ-

**Algorithm 3** Kötter Interpolation Algorithm.

---

**Require:**  $L, (\alpha_i, \beta_i)_{i=1}^n, (m_i)_{i=1}^n, (1, k-1)$ wdeg arbitrary monomial order

```

for  $j = 0$  to  $L$  do
     $g_j := y^j$ 
end for
for  $i = 1$  to  $n$  do
    for  $(r, s) = (0, 0)$  to  $(m_i - 1, 0)$  do
        for  $j = 0$  to  $L$  do
             $\Delta_j := D_{r,s} g_j(\alpha_1, \beta_i)$ 
             $J := \{j : \Delta_j \neq 0\}$ 
        end for
        if  $J \neq 0$  then
             $j^* := \operatorname{argmin}\{g_j : j \in J\}$ 
             $f := g_{j^*}; \Delta := \Delta_{j^*}$ 
            for  $j \in J$  do
                if  $j \neq j^*$  then
                     $g_j := \Delta g_j - \Delta_j f$ 
                else
                     $g_j := \Delta(x - \alpha_i) f$ 
                end if
            end for
        end if
    end for
end for
end for
 $Q_0(x, y) := \min_j\{g_j(x, y)\}$ 
```

---

ating  $\phi_{j_2, \infty}$  with  $\mathbf{e}_{j_2}$ , we can view  $Q$  as an element  $\mathbf{Q}_M$  of the free  $\mathbb{F}_q[z]$ -module  $M = \mathbb{F}_q[z]^a$ , where each component has degree  $\leq b$ .

Let  $Q^{(i, \gamma)}(z) = Q(z + \gamma)$ . We can expand  $Q^{(i, \gamma)}(x)$  around the basis elements  $\phi_{1,i}, \dots, \phi_{l,i}$  at  $P_i$ . By associating  $\phi_{j_2, i}$  with  $\mathbf{e}_{j_2}$  in this expansion,  $Q^{(i, \gamma)}(z)$  can be viewed as an element  $\mathbf{Q}_M^{(i, \gamma)}$  of  $M$ . The function which maps  $Q$  to  $Q^{(i, \gamma)}$  depends only on  $\gamma$  and

$$\{\delta_{i, j_2, j_3} \in \mathbb{F}_q \mid i \in \{1, \dots, n\}, j_2, j_3 \in \{1, \dots, a\}\}.$$

We define as its counterpart the function  $H^{(i, \gamma)} : M \rightarrow M$  that maps  $\mathbf{Q}_M$  to  $\mathbf{Q}_M^{(i, \gamma)}$ . This can be represented graphically as follows

$$\begin{array}{ccc} Q & \longrightarrow & Q^{(i, \gamma)} \\ \downarrow & & \downarrow \\ Q_M & \longrightarrow & Q_M^{(i, \gamma)} \end{array}$$

We see that  $H^{(i,\gamma)}$  is  $\mathbb{F}$ -linear and  $H^{(i,\gamma)}(z\mathbf{b}) = (z + \gamma)H^{(i,\gamma)}(\mathbf{b})$ . Thus  $H^{(i,\gamma)}$  satisfies (3).

Polynomial  $Q$  is said to have a *zero of multiplicity at least  $m$*  at  $(P_i, \gamma)$  if the coefficients of the terms  $\phi_{j_2, i} z^{j_1}$  of  $Q^{(i,\gamma)}(z)$  are zero when  $j_1 + (j_2 - 1) < m$ . Equivalently,  $\mathbf{Q}_M$  has a zero of multiplicity at least  $m$  at  $(P_i, \gamma)$  if the coefficients of the terms  $z_{j_2} \mathbf{e}_{j_2}$  of  $\mathbf{Q}_M^{(i,\gamma)}$  are zero when  $j_1 + (j_2 - 1) < m$ . Thus a module sequence satisfying (4) can be constructed as in Theorem 1.

The following problems have interpolations at their core. The parameters for these interpolations are chosen so as to guarantee the existence of interpolating polynomials whose factors provide the list of valid codewords. These parameters also curtail the search space for polynomials so that efficient techniques can be introduced. In particular, this module description may lead to practical algorithms.

## 7.4 Hard-Decision List Decoding and List Decoding with Soft Information

Let  $(z_1, \dots, z_n)$  be the received word. Define  $s = \lfloor \frac{l-g}{k+g-1} \rfloor$ . A polynomial of the form

$$Q(z) = \sum_{j_1=0}^s \sum_{j_2=1}^{l-g+1-(k+g-1)j_1} q_{j_1 j_2} \phi_{j_2, \infty} z^{j_1}$$

is sought which has a zero of multiplicity at least  $m$  at each point  $(P_i, z_i)$ ,  $1 \leq i \leq n$ . Using the module description, there exists a solution  $\mathbf{Q}_M \in \mathbb{F}[z]^{l-g+1}$  whose terms satisfy

$$(1, k+g-1) - \deg(z^{j_1} \mathbf{e}_{j_2}) = (k+g-1)j_1 + (j_2 - 1) < l - g + 1.$$

All solutions whose terms have this property are contained in  $\mathbb{F}[z]^{l-g+1}$ . A fortiori there is a minimal solution with respect to  $<_{k+g-1, (0, 1, 2, \dots, l-g)}$  in  $\mathbb{F}[z]^{l-g+1}$ . This element will be the first of an ordered Gröbner basis, with respect to this order, of the solution module

$$\{\mathbf{b} \in \mathbb{F}_q[z]^{l-g+1} \mid H^{i, y_i}(\mathbf{b}) \equiv 0 \pmod{M_m}, i = 1, \dots, n\}$$

where

$$\begin{aligned} M_m = \{\mathbf{f} \in \mathcal{P}^u \mid & \text{the coefficients of terms } \mathbf{t} \text{ of } \mathbf{f} \text{ are } 0 \ \forall \mathbf{t} = z^{j_1} \mathbf{e}_{j_2} \\ & \text{with } j_1 + (j_2 - 1) < m\} \end{aligned}$$

Thus, all the requirements of the general algorithm in Sect. 2 are satisfied and the minimal element produced is a solution to the interpolation problem.

We now consider the polynomial

$$Q(z) = \sum_{j_2=0}^a \sum_{j_1=0}^b q_{j_1, j_2} \phi_{j_2, \infty} z^{j_1}.$$

We will say  $Q$  has a *zero of multiplicity at least  $m$*  at  $(P_i, \gamma)$  if the coefficients of the terms  $\phi_{j_2, i} z^{j_1}$  of  $Q^{(i, \gamma)}(z)$  are zero when  $j_1 + j_2 < m$ .

Let  $\gamma_1, \dots, \gamma_q$  be the elements of  $\mathbb{F}_q$ . The task is to find a polynomial  $Q(z)$  with minimal leading term with respect to  $k j_1 + j_2$ , which has a multiplicity at least  $m_{ij} \neq 0$  for each point  $(P_j, \gamma_j)$  with  $j \in [n]$  and  $\gamma_i \in \mathbb{F}_q$ , when  $m_{ij} \neq 0$ . Again, the existence of this polynomial is guaranteed by the problem parameters. The cost  $C(M)$  of the multiplicity matrix  $M$  is defined as  $\frac{1}{2} \sum_{i=1}^q \sum_{j=1}^n m_{ij}(m_{ij} + 1)$ . If we choose  $v$  to be the minimum value such the dimension of the vector space of terms where  $k j_1 + j_2 \leq v$  is greater than  $C(M)$  and we can confine our search to  $\mathbb{F}_q[z]^L$  where  $L = v + 1$ .

We associate  $\phi_{j, i}$  with  $\mathbf{e}_{j+1}$ . Similarly,  $\mathbf{Q}_M$  has a zero of multiplicity at least  $m$  at  $(P_i, \gamma)$  if the coefficients of the terms  $z^{j_1} \mathbf{e}_{j_2}$  of  $\mathbf{Q}_M^{(i, \gamma)}$  are zero when  $j_1 + (j_2 - 1) < m$ . In this way, it is possible to generate a descending module sequence. A minimal element of the solution submodule with respect to  $<_{k, w}$  where  $w = (0, 1, 2, \dots, L - 1)$  corresponds to the required solution and thus, it is possible to apply the general algorithm.

While an algorithm for the hard-decision interpolation problem is an immediate consequence of the general algorithm in O'Keeffe and Fitzpatrick (2002), this problem can also be considered as a special case of the soft-decision interpolation and a common algorithm can be used to solve both.

We can create a “multiplicity” matrix from the received word  $(z_1, \dots, z_n)$ . Let  $m_{ij} = m$  when  $z_i = \gamma_j$  and  $m_{ij} = 0$  otherwise. Set  $L = l - g + 1$  and  $K = k + g - 1$ .

Both problems can be solved using Algorithm 5.2 in O'Keeffe and Fitzpatrick (2007). We just observe that the basis element in  $\mathcal{G}_2$  can be discarded by the *ord* function if the  $(1, K)$ -deg of its leading term is less than  $L$ .

*Remark 3* A different approach is provided by Kötter (1998). Let  $Q$  be a rational point on a smooth and absolutely irreducible curve  $C$  over a finite field  $\mathbb{F}_q$ . Let  $R$  be the ring of functions on  $C$  with poles only at  $Q$ . Let  $\gamma$  be the smallest positive pole order of functions in  $R$  and let  $x \in R$  have pole order  $\gamma$ . Kötter's algorithm treats the ring  $R$  as a free module over the polynomial ring  $\mathbb{F}_q[x]$ . The Kötter algorithm finds a basis for the ideal  $I$  of functions vanishing at the error locations (which is also a free module over  $\mathbb{F}_q[x]$ ). The advantage of Kötter's algorithm is that the decoder is more regular in structure. The updating of the polynomials involves multiplication by  $x$  only, and is therefore simple to implement using linear shift registers. More recently, O'Sullivan (2004) showed that Kötter's algorithm may also be used to compute error evaluator polynomials. One can use the update polynomials and the derivators of the locators to compute error values, thus he avoids computing error evaluator polynomials.

*Remark 4* For more decoding of AG codes, see Sakata (2009b) in this volume.

## 8 Conclusions

Virtually any non-trivial algebraic decoding algorithm has at its core an interpolation step. Since (Mora 2009b) interpolation is an applications of Gröbner bases, especially for techniques like FGLM or Buchberger–Möller, it is hardly surprising that many “FGLM-like” decoding algorithms (as we called them) have arisen in the past.

However, it is a historic fact that the first FGLM-like decoding appeared only in Fitzpatrick (1995), since clearly it took time for the Gröbner basis theory to spread among the coding research community. After 1995 this approach has received more and more attention and when new research directions started in coding theory (notably, the list decoding problem), they have nearly immediately been followed by researchers proposing Gröbner basis approaches.

Now that the Gröbner basis theory is part of the background of most algebraic coding theorists, we think that in a near future we will see more and more applications of these techniques, as for example to oder domain codes (Geil 2009).

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences and RISC, Johannes Kepler University, Linz, Austria.

The authors would like to thank their supervisor: M. Sala.

For many helpful comments and suggestions, the authors heartily thank H. O’Keeffe and P. Fitzpatrick.

This work has been partially supported by STMicroelectronics contract: “Complexity Issues in Algebraic Coding Theory and Cryptography”.

## References

- D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, *Aequationes Math.* **4** (1970), 374–383.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, *London Math. Soc. LNS* **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger’s PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, *J. Symb. Comput.* **41** (2006), nos. 3–4, 475–511.
- M. El-Khamy and R. J. McEliece, *Interpolation multiplicity assignment algorithms for algebraic soft-decision decoding of Reed Solomon codes*, Algebraic coding theory and information theory, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. **68**, AMS, Providence, 2005, pp. 99–120.
- J. B. Farr and S. Gao, *Gröbner bases and generalized Padé approximation*, *Mathematics of Computation* **75** (2005), 461–473.
- P. Fitzpatrick, *On the key equation*, *IEEE Trans. on Inf. Th.* **41** (1995), no. 5, 1290–1302.

- P. Fitzpatrick, *Solving a multivariable congruence by change of term order*, J. Symb. Comput. **11** (1997), 505–510.
- P. Fitzpatrick, *Errors and erasures decoding of BCH codes*, IEE Proc. Commun. **146** (1999), no. 2, 79–81.
- P. Fitzpatrick and J. Flynn, *A Gröbner technique for Padé approximation*, J. Symb. Comput. **13** (1992), 133–138.
- P. Fitzpatrick and S. M. Jennings, *Comparison of two algorithms for decoding alternant codes*, AAECC **9** (1998), no. 3, 211–220.
- P. Fitzpatrick and H. O’Keeffe, *Hard and soft-decision list decoding of 1-point codes as solutions of constrained interpolation problems*, Proc. of ISIT 2002, 2002, p. 308.
- O. Geil, *Algebraic geometry codes from order domains*, this volume, 2009, pp. 121–141.
- V. Guruswami and M. Sudan, *Improved decoding of Reed–Solomon and algebraic geometric codes*, IEEE Trans. on Inf. Th. **45** (1999), no. 6, 1757–1767.
- R. W. Hamming, *Error detecting and error correcting codes*, Bell Systems Technical Journal **29** (1950), 147–160.
- T. Høholdt and R. R. Nielsen, *Decoding Hermitian codes with Sudan’s algorithm*, LNCS, vol. **1719**, Springer, Berlin, 2000, pp. 260–270.
- J. Justesen, K. J. Larsen, H. E. Jensen, A. Havemose, and T. Høholdt, *Construction and decoding of a class of algebraic geometry codes*, IEEE Trans. on Inf. Th. **35** (1989), no. 4, 811–821.
- J. Justesen, K. J. Larsen, H. E. Jensen, and T. Høholdt, *Fast decoding of codes from algebraic plane curves*, IEEE Trans. on Inf. Th. **38** (1992), no. 1, 111–119.
- R. Kötter, *On decoding of algebraic-geometric and cyclic codes*, Ph.D. thesis, Linköping University, 1996.
- R. Kötter, *A fast parallel implementation of a Berlekamp–Massey algorithm for algebraic-geometric codes*, IEEE Trans. on Inf. Th. **44** (1998), no. 4, 1353–1368.
- R. Kötter and A. Vardy, *Algebraic soft-decision decoding of Reed–Solomon codes*, Tech. report, Proc. of ISIT 2000, 2000.
- R. Kötter and A. Vardy, *Algebraic soft-decision decoding of Reed–Solomon codes*, Trans. on Inf. Th. **49** (2003), no. 11, 2809–2825.
- K. Lee and M. E. O’Sullivan, *List decoding of Reed–Solomon codes from a Gröbner basis perspective*, J. Symb. Comput. **43** (2008), no. 9, 645–658.
- D. A. Leonard, *A tutorial on AG code construction from a Gröbner basis perspective*, this volume, 2009, pp. 93–106.
- J. B. Little, D. Ortiz, R. Ortiz-Rosado, R. Pablo, and K. Ri’os-Soto, *Some remarks on Fitzpatrick and Flynn’s Gröbner basis technique for Padé approximation*, J. Symb. Comput. **35** (2003), 451–461.
- J. L. Massey, *Shift-register synthesis and BCH decoding*, IEEE Trans. on Inf. Th. **15** (1969), 122–127.
- R. J. McEliece, *The Guruswami–Sudan decoding algorithm for Reed–Solomon codes*, Tech. report, IPN Progress Report, 42-153, 2003.
- T. Mora, *Gröbner technology*, this volume, 2009a, pp. 11–25.
- T. Mora, *The FGLM problem and Möller’s algorithm on zero-dimensional ideals*, this volume, 2009b, pp. 27–45.
- T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, this volume, 2009, pp. 69–91.
- H. O’Keeffe and P. Fitzpatrick, *Gröbner bases solutions of constrained interpolation problems*, Linear Algebra and its Applications **351–352** (2002), 533–551.
- H. O’Keeffe and P. Fitzpatrick, *Gröbner basis approach to list decoding of algebraic geometry codes*, AAECC **18** (2007), no. 5, 445–466.
- M. E. O’Sullivan, *On Kötter’s algorithm and the computation of error values*, Des., Codes and Crypt. **31** (2004), 169–188.
- R. Pellikaan and X.-W. Wu, *List decoding of  $q$ -ary Reed–Muller codes*, IEEE Trans. on Inf. Th. **50** (2004), no. 4, 679–682.
- S. C. Porter, B. Z. Shen, and R. Pellikaan, *Decoding geometric Goppa codes using an extra place*, IEEE Trans. on Inf. Th. **38** (1992), no. 6, 1663–1676.

- S. Sakata, *The BMS algorithm*, this volume, 2009a, pp. 143–163.
- S. Sakata, *The BMS algorithm and decoding of AG codes*, this volume, 2009b, pp. 165–185.
- B. Z. Shen, *Algebraic-geometric codes and their decoding algorithm*, Ph.D. thesis, Eindhoven Univ. Tech., 1992.
- M. Sudan, *Decoding of Reed–Solomon codes beyond the error correction bound*, J. of Complexity **13** (1997), 180–193.

# An Introduction to Ring-Linear Coding Theory

Marcus Greferath

**Abstract** This contribution gives an introduction to algebraic coding theory over rings. We will start with a historical sketch and then present basics on rings and modules. Particular attention will be paid to weight functions on these, before some foundational results of ring-linear coding will be discussed. Among these we will deal with code equivalence, and with MacWilliams' identities about the relation between weight enumerators. A further section is devoted to existence bounds and code optimality. An outlook will then be presented on the still unsolved problem of the construction of large families of ring-linear codes of high quality.

## 1 Introduction and History

Ring-linear coding theory is a discipline of algebraic coding theory where the underlying alphabet does not carry the structure of a finite field but merely of a finite ring or, more generally, of a module. Such a setup was considered much earlier than widely assumed: in their contribution *Error-Correcting Codes: an Axiomatic Approach*, Assmus and Mattson (1963) first mention rings as possible alphabets for linear codes. It took however considerable time for ring-linear coding theory to develop from these origins to nowadays' state-of-the-art. For an introduction to linear and cyclic codes over fields, see Augot et al. (2009).

In the seventies of the previous century, Blake (1972, 1975) presented linear codes first over semi-simple, later over primary integer residue rings. Analogs of Hamming, Reed–Solomon and BCH Codes were also introduced. Spiegel (1977, 1978) pursued a group-algebraic approach to linear codes over  $\mathbb{Z}_m$ . Like Blake, he used the Chinese Remainder Theorem to investigate BCH Codes over these rings. Shankar (1979) presented a polynomial approach to cyclic codes over integer residue rings which enabled notions of generator polynomials for cyclic codes. Later, Satyanarayana (1979) investigated linear codes over integer residue rings equipped with the Lee weight. Constant weight codes and Reed–Muller type codes were presented as well. It was common to most of these early papers to consider alphabets equipped with the Hamming weight. Although the Lee metric was used by Satyanarayana, a significant change in the metrics used would occur only much later.

In the eighties Klemm (1987) considered linear codes over integer residue rings and proved MacWilliams' weight enumerator theorem. His investigations were

---

M. Greferath

School of Mathematical Sciences, University College Dublin, Dublin, Ireland

e-mail: [marcus.greferath@ucd.ie](mailto:marcus.greferath@ucd.ie)

based on a suitable weight function to obtain his result. Nechaev (1991) discovered that all Kerdock codes can be understood as cyclic linear codes over  $\mathbb{Z}_4$ . While Klemm introduced a novelty regarding the metrical aspect, Nechaev's result did not involve a statement regarding metrics. The latter paper predates however the breakthrough that came with the paper by Hammons et al. (1994) in the nineties.

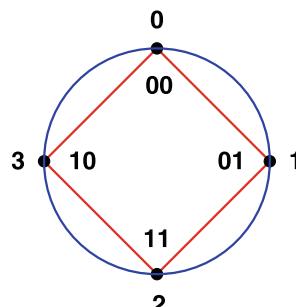
Traditional finite-field coding theory had shown that considering linear codes promises significant advantages over the non-linear counterparts when it comes to complex tasks like encoding and decoding. Although very good linear codes were known, it was recognized early (Preparata 1968; Kerdock 1972a, 1972b) that the class of binary block codes contained excellent codes, which were however not linear. This was the case for the families of Preparata, Kerdock codes, Goethals and Goethals–Delsarte codes. Apart from their quality, these families showed formal duality properties in terms of their distance enumerators that resembled those among linear codes and their duals.

This phenomenon went unexplained for a long time. A breakthrough in the understanding of this behavior came in the 1990's when Nechaev (1991) and, independently, Hammons et al. (1994) discovered that these families allow a representation in terms of  $\mathbb{Z}_4$ -linear codes.

The central insight in this  $\mathbb{Z}_4$ -linear representation came from the fact that the alphabet was equipped with an alternative weight function, the Lee weight. The Hamming weight distinguishes only whether an element is zero or not. The Lee weight is finer, giving the zero-divisor 2 a weight different from that of the other ring elements.

Figure 1 depicts the so-called Gray isometry which is a bijection between  $(\mathbb{Z}_4, w_{\text{Lee}})$  and  $(\mathbb{Z}_2^2, w_H)$  that preserves the weight. Extending this isometry coordinatewise to  $\mathbb{Z}_4^n$  we obtain a binary code of length  $2n$  from each  $\mathbb{Z}_4$ -code of length  $n$ . It is exactly this way that the above code families found their  $\mathbb{Z}_4$ -linear representations.

To be more specific: Let  $f \in \mathbb{F}_2[x]$  be a primitive monic polynomial of degree  $r$ . This polynomial divides the polynomial  $x^n - 1 \in \mathbb{F}_2[x]$  where  $n = 2^r - 1$ . Hensel's lemma shows that there exists a monic divisor  $\tilde{f}$  of  $x^n - 1 \in \mathbb{Z}_4[x]$  which is a preimage of  $f$  under the natural epimorphism  $v : \mathbb{Z}_4 \longrightarrow \mathbb{F}_2$ . Taking this polynomial  $\tilde{f}$  as generator polynomial for a cyclic  $\mathbb{Z}_4$ -linear code, and extending this code by an overall parity check, we end up with a linear code  $P_r$  which for odd  $r$  has parameters



**Fig. 1** Gray isometry between  $\mathbb{Z}_4$  and  $\mathbb{Z}_2^2$

$[2^r, 2^r - 1 - r, 6]$  with respect to the Lee distance on  $\mathbb{Z}_4$ . The binary image of this code under the above Gray isometry is a code that has the same parameters as the Preparata code of order  $r$ .

The above observations suggest to expand the traditional theory of linear codes in at least two directions. On the one hand, it seems obvious that the next more general algebraic structure that might serve as an alphabet for linear coding is that of finite rings, or slightly more general, finite modules. On the other hand, the appropriateness of the Lee weight for  $\mathbb{Z}_4$ -linear coding shows that the metric component of a generalized coding theory also requires a generalisation.

## 2 Rings and Modules

As this book is interested only in the commutative case, we will assume rings to be commutative, associative and unital in the sequel. Note however, that most of the concepts and results that we present here have a non-commutative counterpart. The unit group of a ring  $R$ , i.e. the set of its multiplicatively invertible elements, will be denoted by  $R^\times$ . Talking about a module  $_R M$  we will always assume that  $1 \cdot m = m$  for all  $m \in M$ .

For a subset  $X$  of a module  $_R M$  we denote by  $\text{Ann}_R(X)$  the annihilator of  $X$  in  $_R R$ , which is an ideal of  $R$ . The module  $_R M$  is called faithful if  $\text{Ann}_R(M) = 0$ , and an element  $x \in M$  is called free if  $\text{Ann}_R(\{x\}) = 0$ .

If the kernel of any epimorphism onto  $_R M$  is a direct summand in the source module, then  $_R M$  is called a projective module. Alternatively the class of all projective modules can be characterized as the class of all direct summands of free modules. We call a module  $_R M$  injective, if its image under any monomorphism is a direct summand in the target module. If  $R$  as a module over itself is injective then we call  $R$  self-injective.

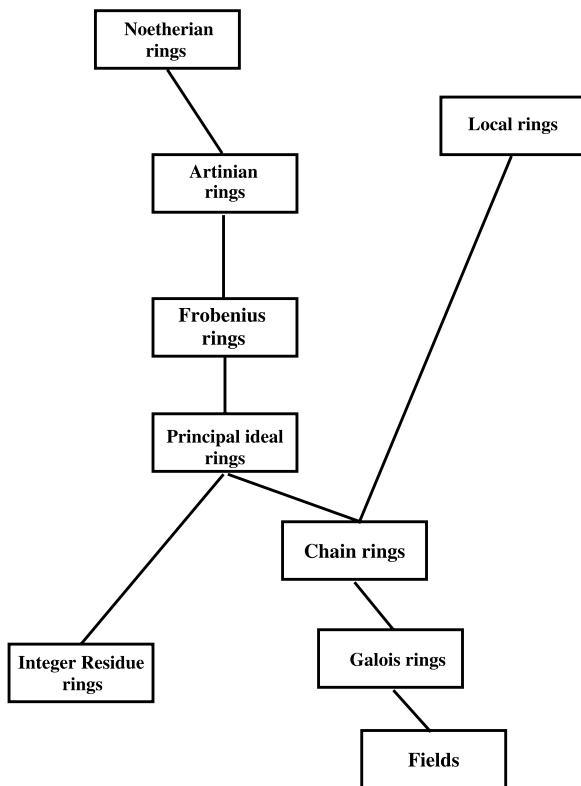
The intersection of all maximal proper submodules of a module  $_R M$  is called the Jacobson radical of  $_R M$  and is denoted by  $\text{rad}(_R M)$ . The sum of all nonzero minimal submodules of  $_R M$  is called the socle of  $_R M$  and is denoted by  $\text{soc}(_R M)$ . Particular attention will be paid to  $\text{rad}(R)$  and  $\text{soc}(R)$ .

### 2.1 Some Classes of Rings

Various kinds of rings are important for ring-linear coding theory, among them also classes of non-finite rings (cf. Calderbank and Sloane 1995). But even in the context of traditional finite-field coding theory this is not surprising if one thinks of convolutional codes as linear block codes over polynomial rings.

We will now discuss some classes of rings considering their logical dependencies. We will start with the class of all Noetherian rings which is defined by what is called the ascending chain condition: every ascending chain of ideals has a largest

**Fig. 2** Classes of commutative rings and their logical dependencies



element. The class of Noetherian rings contains  $\mathbb{Z}$ , but also all (multivariate) polynomial rings  $\mathbb{F}[x_1, \dots, x_n]$  over fields as prominent examples. A subclass of this class is that of all Artinian rings, which is defined by the so-called descending chain condition: every descending chain of ideals has a smallest element.

A Noetherian ring is called a Frobenius ring if it is self-injective. It can be seen that a ring  $R$  of this class is Artinian at the same time, and that  $(R/\text{rad}(R)) \cong \text{soc}(R)$  as  $R$ -modules (cf. Rowen 1991, p. 347f, Lam 1999, Chap. 15). The class of all Frobenius rings contains all Artinian chain rings (as defined below) as subclasses. For some aspects of our presentation it is important that within the class of all finite rings exactly the Frobenius rings are those which possess a free character module. Equivalently, these rings are exactly those which possess a cyclic socle (cf. Wood 1999 and Honold 2001).

A further important class of commutative rings is that of all local (commutative) rings, i.e. those rings which possess a unique maximal ideal. It can be shown that every Artinian ring is a ring-direct product of local rings. As mentioned earlier, we need to emphasize the chain rings. A ring is called a chain ring if the set of its ideals forms a chain. For a classification of all chain rings see (Al-Khamees 1995; Clark and Liang 1973; Clark and Drake 1973).

Last, but not least, there is a large class of finite chain rings which are called Galois rings. For this let  $p$  be a prime number,  $k, r$  positive integers, and let  $f \in \mathbb{Z}_{p^k}[x]$  be a monic polynomial of degree  $r$  that is irreducible modulo  $\mathbb{Z}_p$ . Then the chain ring  $\text{GR}(p^k, r) := \mathbb{Z}_{p^k}[x]/(f)$  is—up to isomorphism—uniquely determined by  $p^k$  and  $r$ , and is called Galois ring of characteristic  $p^k$  and rank  $r$  (cf. Krull 1923; Clark and Liang 1973). Its Jacobson radical is generated by the number  $p$ , and its residual field is isomorphic to  $\mathbb{F}_{p^r}$ .

### 3 Weight Functions on Finite Rings and Modules

For traditional algebraic coding theory over finite fields the Hamming weight plays a dominant role which stems from the fact that non-zero elements of a field  $F$  cannot be distinguished as vectors in the space  $_F F$ . This does not hold in a module based theory of linear codes. An important property of finite rings and modules is their non-trivial lattice of submodules and consequently the possibility to distinguish module elements in an order-theoretic way.

In search for a weight function on a finite module  $_R M$  that plays the same role for ring-linear coding theory as the Hamming weight for finite-field coding theory a first contribution was due to Constantinescu (1995) who coined the notion of homogeneous weights on integer residue rings. These weight functions are prominent for two reasons: firstly, homogeneous weights are constant on classes of associate elements of  $\mathbb{Z}_m$ ; secondly the average weight of every ideal of  $\mathbb{Z}_m$  is constant.

The existence of homogeneous weights on a finite module can easily be proved using Möbius inversion on certain function spaces. This has been done first in Greferath and Schmidt (2000) for all finite rings, and later in Greferath et al. (2004) for arbitrary finite modules. For preparation recall the inversion calculus on partially ordered sets (cf. Rota 1964; Stanley 1997, Chap. 3.6; Aigner 1997, Chap. IV): If  $P$  is a locally finite partially ordered set, the Möbius function  $\mu : P \times P \rightarrow \mathbb{R}$  is defined by  $\mu(x, x) := 1$  and  $\mu(x, y) := 0$  for all  $x \not\leq y$ , and implicitly by the relation  $\sum_{x \leq t \leq y} \mu(x, t) = 0$  for all  $x < y$ . This function induces the following equivalence for arbitrary pairs  $f, g$  of real-valued functions on  $P$ :

$$g(y) = \sum_{x \leq y} f(x) \quad \text{for all } y \in P \iff f(y) = \sum_{x \leq y} g(x)\mu(x, y) \quad \text{for all } y \in P.$$

Let  $R$  be a finite ring in the following, and let  $_R M$  be a finite module. Finally, let  $\mu$  denote the Möbius function on the set of all  $\{Rx \mid x \in M\}$  which is clearly partially ordered by set inclusion, and let  $R^\times x$  denote the set of all generating elements of the submodule  $Rx$  of  $_R M$ .

Call a weight  $w : M \rightarrow \mathbb{R}$  homogeneous, if  $w(0) = 0$  and the following hold:

- (H1) For all  $x, y \in M$  the equality  $Rx = Ry$  implies  $w(x) = w(y)$ .
- (H2) There exists a real number  $\gamma$  such that for all  $x \in M \setminus \{0\}$  there holds  $\sum_{y \in Rx} w(y) = \gamma |Rx|$ .

We then have the following characterization of homogeneous weights on finite modules (cf. Greferath et al. 2004).

**Theorem 1** *A weight  $w$  on the finite module  $R M$  is homogeneous if and only if the following holds:*

$$(H) \text{ There exists a positive real number } \gamma \text{ such that } w(x) = \gamma \left(1 - \frac{\mu(0, Rx)}{|R^x x|}\right).$$

The reader should note further work on weight functions which is in close connection to the homogeneous weight that we have introduced here. A first paper to mention is the one by Carlet (1998) who generalised the above-mentioned concept of Gray isometry for the rings  $\mathbb{Z}_{2k}$ . Such an isometry will generally be onto a suitable binary first order Reed–Muller code, and hence not onto the ambient space of the latter code. A further result dealing with such isometries can be found in Salagean (1999).

## 4 Linear and Cyclic Codes

A module theoretic generalisation of the traditional setup of algebraic coding theory is quite simple. It only requires to acknowledge the possible non-existence of bases and complements.

If  $R M$  is a finite module and  $n$  a natural number then the submodules of  $R M^n$  will be called  $R$ -linear codes of length  $n$  over  $M$ . This is the most general context for ring-linear coding theory. In most contributions  $M = R$  assuming a restriction on  $R$  like being a finite Frobenius ring. Note however that often such a restriction can be given up by exchanging the ring by a suitable module.

Let  $C$  be an  $R$ -linear code over  $R$ . We say that  $C$  is splitting, if it possesses a complement in  $R R^n$ . If  $C$  possesses a basis, we will call it a free code. A  $(k \times n)$ -matrix  $G$  with coefficients in  $R$  is called a generator matrix for  $C$  if  $C = \{xG \mid x \in R^k\}$ . Such a matrix will be called a check matrix for  $C$  if  $C = \{x \in R^n \mid xG^t = 0\}$ .

Note that if  $R$  is a finite Frobenius ring then all  $R$ -linear codes possess parity-check matrices. For obvious reasons statements on the number of rows of generator and check matrices, as well as standard forms for these matrices, can be proved only under suitable additional conditions.

For the following consider the standard inner product

$$R^n \times R^n \longrightarrow R, \quad (x, y) \mapsto xy := \sum_{i=1}^n x_i y_i.$$

If  $C$  is a linear code of length  $n$  over  $R$  then the code  $C^\perp := \{x \in R^n \mid cx = 0 \text{ for all } c \in C\}$  is again linear, and will be called the dual code of  $C$ . Let us call the left  $R$ -linear code  $C$  self-dual, if  $C = C^\perp$ .

## 4.1 Cyclic Linear Codes

It is well known that cyclic linear codes of length  $n$  over a (finite) field  $\mathbb{F}$  can be characterized in terms of the divisors of the polynomial  $x^n - 1$  in  $\mathbb{F}[x]$ .

This has been generalised in Greferath (1997). There are only a few algebraic facts which need to be observed in order to obtain a complete characterization of a large class of cyclic linear codes over finite rings. One of these is that if  $gh = x^n - 1$  for some  $g, h \in R[x]$ , then it can then be seen that  $R[x]h$  is a free  $R$ -module, and that  $R[x]g$  is a direct summand of the  $R$ -module  $R[x]$ . Further preparation can be looked up in most standard texts on abstract algebra: if  $R$  is a finite ring, then  $\text{rad}(R)[x]$  is a negligible submodule of the  $R$ -module  $R[x]$ , which means for any  $R$ -submodule  $U$  of  $R[x]$  with  $\text{rad}(R)[x] + U = R[x]$  there follows  $U = R[x]$ .

**Theorem 2** *For a cyclic linear code  $C$  of length  $n$  over a finite ring  $R$  the following are equivalent:*

- (a)  *$C$  is a splitting code.*
- (b) *There exists a divisor  $g$  of  $x^n - 1$  in  $R[x]$  such that  $C = R[x]g/(x^n - 1)$ .*

Another interesting result deserves to be mentioned here. It has been first observed in Calderbank and Sloane (1995, Theorem 6) (see also Kanwar and López-Permouth 1997).

**Theorem 3** *Let  $R$  be a Galois ring of characteristic  $p^m$ , and let  $C \subseteq R^n$  be a cyclic code of length  $n$  where  $p \nmid n$ . Then there exists a polynomial  $f \in R[x]$  such that*

$$C = (fR[x] + (x^n - 1))/(x^n - 1).$$

In summary, we can state that, among the cyclic codes of length  $n$  over a Galois ring  $R$ , exactly the free codes are those which possess a generator polynomial which is a divisor of  $x^n - 1 \in R[x]$ .

It is plausible that cyclic codes that do not possess a generator polynomial should allow to be dealt with using Gröbner bases (Buchberger 1965, 1985, 2006). Among various papers approaching this question we mention the work by Salagean and Norton (2001a, 2001b, 2002, 2003). Further work, particularly in the context of multivariate polynomials rings was done by Martinez-Moro et al. (2006). For a Gröbner basis decoding, see Byrne and Mora (2009).

## 5 A Foundational Result: Code Equivalence

Let  $R$  be a finite ring that is equipped with a weight function  $w : R \longrightarrow \mathbb{R}$ . As usual, we extend this function additively to the  $R^n$ , and the natural question arises if all  $w$ -isometries  $\varphi : C \longrightarrow R^n$ , i.e. of all those linear mappings for which  $w(\varphi(c)) = w(c)$  for all  $c \in C$  can be characterised in a nice way.

A basic theorem achieving this in the context of traditional finite-field linear coding theory is MacWilliams' (1962) equivalence theorem. To understand its terms, let a Hamming isometry of  $C$  into  $R^n$  be an  $R$ -linear map that preserves the Hamming weight. A monomial transformation of  $R^n$  is a mapping  $\psi : R^n \longrightarrow R^n$  with

$$\psi(x) = x P D$$

for all  $x \in R^n$ , where  $P$  is a (coordinate) permutation matrix and  $D$  is an invertible diagonal matrix in  $M_n(R)$ .

MacWilliams' equivalence theorem then can be stated as follows:

**Theorem 4** *If  $\mathbb{F}$  is a finite field and  $C \leq \mathbb{F}^n$  a linear code, then every Hamming isometry  $C \longrightarrow \mathbb{F}^n$  can be extended to a monomial transformation of  $\mathbb{F}^n$ .*

This means that the usual textbook definition of monomial code equivalence is justified for the class of all linear codes and linear Hamming isometries.

It was Wood (1999) who first suggested that finite Frobenius rings form an appropriate class of rings for ring-linear coding theory, in that MacWilliams' equivalence theorem holds for linear codes over finite Frobenius rings.

### Theorem 5

- (a) *If  $R$  is a finite Frobenius ring and  $C \leq R^n$  a linear code, then every Hamming isometry  $C \longrightarrow R^n$  can be extended to a monomial transformation of  $R^n$ .*
- (b) *If a finite (commutative) ring  $R$  satisfies, that all Hamming isometries between linear codes over  $R$  allow for monomial extensions, then  $R$  is a Frobenius ring.*

Another recent work (Greferath et al. 2004) shows that when the character module  $\hat{R} := \text{Hom}(R, \mathbb{Q}/\mathbb{Z})$  is used as alphabet in lieu of the ring  $R$  itself, then MacWilliams' equivalence theorem holds without imposing any hypotheses on the underlying ring  $R$ .

So far, we have only discussed Hamming isometries and their monomial representation. The immediate question now is if also other weight functions on finite Frobenius rings lead to results like MacWilliams' equivalence theorem. This question has not been answered yet in general, but there at least one result that should be quoted here. A homogeneous isometry will be a linear isomorphism that preserves the homogeneous weight as discussed in a previous section.

**Theorem 6** *If  $R$  is a finite Frobenius ring and  $C \leq R^n$  a linear code, then every homogeneous isometry  $C \longrightarrow R^n$  can be extended to a monomial transformation of  $R^n$ .*

Regarding a complete characterization of all those weight functions for which isometries between linear codes can be extended to monomial transformations of the ambient space there has appeared work by Wood (1997). It characterises such weight functions when the underlying ring is a chain ring.

**Theorem 7** Let  $R$  be a finite chain ring. Then every weight function  $w : R \rightarrow \mathbb{R}$  for which  $\sum_{r \in \text{soc}(R)} w(r)$  does not vanish, allows a MacWilliams' equivalence theorem.

## 6 Weight Enumerators and MacWilliams' Identity

One of the most important computational tools in traditional algebraic coding theory is based on a theorem by MacWilliams' that establishes a connection between the weight enumerators of a linear code and its dual.

Let  $C \leq R^n$  be a linear code. The integer polynomial

$$W_C(x) = \sum_{i=0}^n A_i x^i \quad \text{where } A_i := \#\{c \in C \mid w(c) = i\} \text{ for } i \in \{0, \dots, n\}$$

is called the weight enumerator of  $C$ .

As stated earlier the dual code  $C^\perp$  is again an  $R$ -linear code, and there arises the question in how far the weight enumerator of  $C$  determines that of  $C^\perp$ .

MacWilliams' famous theorem on weight enumerators states an answer to this question and may be quoted as follows (cf. MacWilliams and Sloane 1977):

**Theorem 8** Let  $\mathbb{F}_q$  be the finite field of  $q$  elements, and let  $C \leq \mathbb{F}_q^n$  be a linear code with Hamming weight enumerator  $W_C(x)$ . Then the Hamming weight enumerator of  $C^\perp$  is given by

$$W_{C^\perp}(x) = \frac{1}{|C|} [1 + (q - 1)x]^n W_C\left(\frac{1 - x}{1 + (q - 1)x}\right).$$

Again, it was J. Wood to observe that this theorem remains true when  $\mathbb{F}$  is exchanged by any finite Frobenius ring  $R$ . This stems from the fact that the statement allows for a formulation in a much more general context. For this let  $x = (x_s)_{s \in R}$  be a family of indeterminates and  $\mathbb{Z}[x]$  be the multivariate polynomial ring. Define the complete weight enumerator of  $C$  as the integer polynomial

$$\text{CWE}_C(x) = \sum_{c \in C} \prod_{i=1}^n x_{c_i}.$$

**Theorem 9** Let  $R$  be a finite Frobenius ring and let  $C \leq R^n$  be a linear code with complete weight enumerator  $\text{CWE}_C(x)$ . Then the complete weight enumerator of  $C^\perp$  is given as

$$\text{CWE}_{C^\perp}(x) = \frac{1}{|C|} \text{CWE}_C(Mx)$$

where  $M$  is the matrix with entry  $M_{s,t} = \chi(st)$  and  $\chi$  is a generating character of  $R$ .

For a proof of this theorem see Wood (1999). From the same source we cite the following result which states that MacWilliams' weight enumerator theorem allows for a generalisation to symmetrized versions. The first to mention is the one dealing with Hamming weight enumerators.

**Theorem 10** *Let  $R$  be a finite Frobenius ring, and let  $C \leq R^n$  be a linear code with Hamming weight enumerator  $W_C(x)$ . Then the Hamming weight enumerator of  $C^\perp$  is given by*

$$W_{C^\perp}(x) = \frac{1}{|C|} [1 + (|R| - 1)x]^n W_C\left(\frac{1-x}{1+(|R|-1)x}\right).$$

For a more general symmetrisation let  $U$  be a subgroup of  $R^\times$ . Let  $S := R/U$  denote the set of all  $U$ -associate classes of elements in  $R$ . Consider the polynomial ring  $\mathbb{Z}[x_{Ur} \mid Ur \in S]$ , and define the symmetrized weight enumerator of the  $R$ -linear code  $C \leq R^n$  as

$$\text{SWE}_C^U(x) = \sum_{c \in C} \prod_{Ur \in S} x_{UR}^{n_{Ur}(c)}, \quad \text{where } n_{Ur}(c) = \#\{1 \leq i \leq n \mid c_i \in Ur\}.$$

Then we have the following theorem (cf. Wood 1999):

**Theorem 11** *Let  $R$  be a finite Frobenius ring and let  $C \leq R^n$  be a linear code with symmetrized weight enumerator  $\text{SWE}_C^U(x)$ . Then the symmetrized weight enumerator of  $C^\perp$  is given as*

$$\text{SWE}_{C^\perp}^U(x) = \frac{1}{|C|} \text{SWE}_C^U(N^\dagger x)$$

where  $N$  is the matrix with entry  $N_{Us,Ut} = \sum_{u \in U} \chi(ust)$ .

It was clear early that other weight functions on a finite ring, such as the homogeneous weight, cannot give rise to a symmetrised version of the above theorem in the traditional way, since the partition induced by the homogeneous weight is not compatible with the Fourier transform in general. A recent article by Byrne et al. (2007) however shows that even in this case something can be done.

For that, let  $E(R)$  denote the set of all equivalence relations on the finite Frobenius ring  $R$ . Given an arbitrary equivalence relation  $\theta \in E(R)$ , we write  $R/\theta$  for the set of equivalence classes of  $\theta$  and write  $r\theta$  for the equivalence class containing the element  $r \in R$ . Note that  $E(R)$  is a partially ordered set, with respect to the relation given by  $\theta \leq \theta'$  if and only  $r\theta s$  implies  $r\theta' s$  for all  $r, s \in R$ . Equivalently, we say that the partition induced by  $\theta$  is a refinement of that induced by  $\theta'$  if and only if  $\theta \leq \theta'$ . Note that  $E(R)$  has a least element  $\Delta$  which is the equality relation on  $R$ .

We consider the map  $\Phi : E(R) \rightarrow E(R)$  that assigns to each  $\theta \in E(R)$  the relation  $\Phi(\theta)$  defined by

$$s\Phi(\theta)s' \quad \text{if and only if} \quad \sum_{t \in r\theta} \chi(ts) = \sum_{t \in r\theta} \chi(ts') \quad \text{for all } r \in R.$$

It is easily seen that  $\Phi(\theta)$  is indeed an equivalence relation on  $R$  and that  $\Phi$  is an order preserving mapping on  $E(R)$ . Moreover, the image of  $\Phi$  is contained in the interval  $[\Delta, \theta_H]$ , where  $\theta_H$  is the equivalence relation induced by the Hamming weight on  $R$ .

As presented above, Wood (1999) derived MacWilliams identities for three choices of  $\theta$  being fixed by  $\Phi$ : firstly  $\theta = \Delta$  (the unsymmetrized case relating complete weight enumerators), then  $\theta = \theta_H$  (the relation induced by the Hamming weight on  $R$ ), and finally the relation  $\theta_U$ , which is induced by the subgroup  $U$  of  $R^\times$ .

The following example now shows that the partition induced by the homogeneous weight is not invariant under  $\Phi$ .

*Example 1* On the ring  $\mathbb{Z}_8$ , the homogeneous weight (with  $\gamma = 1$ ) satisfies  $0 \mapsto 0$ ,  $4 \mapsto 2$  and  $r \mapsto 1$  for all remaining  $r \in \mathbb{Z}_8$ . The relation  $\theta_{\text{hom}}$  induced by this weight has therefore the classes  $\{0\}$ ,  $\{4\}$  and  $\{1, 2, 3, 5, 6, 7\}$ . The classes of  $\Phi(\theta_{\text{hom}})$  are given by  $\{0\}$ ,  $\{2, 4, 6\}$  and  $\{1, 3, 5, 7\}$  which shows that  $\Phi(\theta_{\text{hom}}) \neq \theta_{\text{hom}}$  in general. Note however that here  $\Phi(\Phi(\theta_{\text{hom}})) = \theta_{\text{hom}}$ .

For applications of the MacWilliams identities to obtaining bounds on the size of a code it is important to know how an equivalence relation  $\theta$  on a finite ring needs to be chosen such that  $\Phi(\theta) = \theta_{\text{hom}}$ . That has been settled at least in the case of a local Frobenius ring. It is conjectured that it can be proven in general.

**Proposition 1** *Let  $R$  be a finite local Frobenius ring. Let  $\theta \in E(R)$  be the equivalence relation with partition  $R/\theta = [\{0\}, \text{rad}(R) \setminus \{0\}, R^\times]$ . Then*

$$\Phi(\theta) = \theta_{\text{hom}}.$$

For an equivalence relation  $\theta$  on  $R$ , consider the corresponding family of indeterminates  $y = (y_{r\theta})_{r\theta \in R/\theta}$ . Let  $\tau \in E(R)$  be given such that  $\tau \leq \Phi(\theta)$ , and let the family of indeterminates  $z = (z_{r\tau})_{r\tau \in R/\tau}$  belong to the classes of  $\tau$ .

We will now establish an identity relating the  $\theta$ -symmetrized weight enumerator

$$S_{C^\perp}(y) := \sum_{c \in C^\perp} \prod_{i=1}^n y_{c_i \theta} \quad \text{of } C^\perp$$

to the  $\tau$ -symmetrized weight enumerator

$$S_C(z) := \sum_{c \in C} \prod_{i=1}^n z_{c_i \tau} \quad \text{of } C$$

which is given in the following theorem.

**Theorem 12** (MacWilliams' Identity) *Let  $\theta, \tau$  be equivalence relations on  $R$  such that  $\tau \leq \Phi(\theta)$ . Let  $C \subseteq R^n$  be a linear code with  $\tau$ -symmetrized weight enumerator*

$S_C(z)$ , and let  $C^\perp \leq R^n$  denote its dual with  $\theta$ -symmetrized weight enumerator  $S_{C^\perp}(y)$ . Then

$$S_{C^\perp}(y) = \frac{1}{|C|} N(S_C(z)),$$

where  $N$  is defined as the natural extension of  $N(z_{s\tau}) = \sum_{r\theta \in R/\theta} y_{r\theta} [\sum_{t \in r\theta} \chi(ts)]$ .

The reader might understand that for computational reasons the case  $\tau = \Phi(\theta)$  is of particular interest.

We finally remark that the mentioned work (Greferath et al. 2004) again shows that using the character module  $\hat{R}$  as alphabet rather than  $R$  itself, many of the above theorems hold without imposing any hypotheses on the underlying ring  $R$ .

## 7 Code Optimality: Bounds on the Parameters of Codes

Among the traditional bounds on the parameters of block codes over finite fields the most important are probably the sphere-packing bound, the Singleton bound, the Gilbert–Varshamov bound, the linear programming bound, the Plotkin and Elias bounds and the Griesmer bound. With the exception of the Gilbert–Varshamov bound, which asserts the existence of codes having parameters above a given bound, all these bounds restrict the existence of codes with given parameters, and give rise to various notions of code optimality.

In the traditional context (see Augot et al. 2009)  $\mathbb{F}_q$  is the finite field with  $q$  elements, and  $A_q(n, d)$  denotes the maximal number of words that a block code of length  $n$  over  $\mathbb{F}_q$  with minimum Hamming distance  $d$  can have. It is clear that the sphere of radius  $t$  in the Hamming space  $\mathbb{F}^n$  has volume

$$\text{vol}_q(n, t) = \sum_{i=0}^t \binom{n}{i} (q-1)^i.$$

Now the sphere-packing bound states that

$$A_q(n, d) \leq \frac{q^n}{\text{vol}_q(n, \lfloor \frac{d-1}{2} \rfloor)}.$$

The Singleton bound states that

$$A_q(n, d) \leq q^{n+1-d}.$$

Abbreviating  $\gamma = \frac{q-1}{q}$  and assuming  $\gamma n < d$  there holds the Plotkin bound which says that

$$A_q(n, d) \leq \frac{d}{d - \gamma n}.$$

The Elias bound, an extensive refinement of the Plotkin bound, states that for every  $t \in \mathbb{R}$  with  $t < \gamma n$  and  $t^2 - 2t\gamma n + d\gamma n > 0$  there holds

$$A_q(n, d) \leq \frac{\gamma n d}{t^2 - 2t\gamma n + d\gamma n} \cdot \frac{q^n}{\text{vol}_q(n, t)}.$$

Finally, there is the famous linear programming bound which says that

$$A_q(n, d) \leq \max \sum_{i=1}^n A_i$$

where the  $(A_i)_{i=1 \dots n}$  are taken to be families of non-negative numbers with  $A_0 = 1$  and  $A_i = 0$  for all  $1 \leq i \leq d - 1$ , such that

$$\sum_{i=0}^n A_i P_k(i) \geq 0 \quad \text{for } k = 0, \dots, n.$$

Here

$$P_k(x) = \sum_{j=0}^n \binom{x}{j} \binom{n-x}{k-j} (-1)^j (q-1)^{k-j}$$

are the well-known Krawtchouk polynomials.

Regarding existence bounds for ring-linear codes it might not be surprising that on the algebraic side the class of all finite Frobenius rings is a class subject to most investigations. It is also clear that traditional bounds need to be generalized to the case where a finite ring is equipped with a weight function that differs from the Hamming weight.

Regarding rings equipped with the Hamming weight we mention an approach in Shiromoto (2000) for codes over finite commutative Frobenius rings (Singleton bound). In Shiromoto and Storme (2003) there can be found a generalisation of the Griesmer bound for codes over finite commutative Frobenius rings.

For other weight functions it turns out most of the classical bounds have natural counterparts, at least as far as the homogeneous weight as introduced before is given on the ring under consideration. The sphere-packing bound for example requires no modification, except that its version for a finite ring no longer involves a simple expression in terms of a binomial distribution.

Some recent work in Greferath and O'Sullivan (2004) provides a character based method to derive versions of the Plotkin and Elias bounds when the ring is equipped with a homogeneous weight.

Let  $R$  be a finite Frobenius ring and let  $w$  be a homogeneous weight on  $R$  of average value  $\gamma$ . Let  $A_{\text{hom}}(n, d)$  denote the maximal number of words  $a$  (not necessarily linear) code of length  $n$  over  $R$  that has minimum homogeneous distance  $d$ . Finally let  $\text{vol}_w(n, t)$  be the volume of the sphere of (homogeneous) radius  $t$  in the space  $R^n$ .

**Theorem 13** (Plotkin bound) *For every  $n, d$  with  $\gamma n < d$  there holds*

$$A_{\text{hom}}(n, d) \leq \frac{d}{d - \gamma n}.$$

Using a refinement in the technique of proof the mentioned work (Greferath and O'Sullivan 2004) establishes a version of the Elias bound.

**Theorem 14** (Elias Bound) *For every  $n, d, t$  with  $t \leq \gamma n$  and  $t^2 - 2t\gamma n + d\gamma n > 0$  there holds*

$$A_{\text{hom}}(n, d) \leq \frac{\gamma n d}{t^2 - 2t\gamma n + d\gamma n} \cdot \frac{|R|^n}{\text{vol}_w(n, t)}.$$

To derive a version of the linear programming bound we need some further notation and preparation. For  $\alpha \in \mathbb{N}^R$  we write  $|\alpha| = n$  in order to indicate that  $\sum_{r \in R} \alpha_r = n$ . It is also common to abbreviate  $x^\alpha := \prod_{r \in R} x_r^{\alpha_r}$ .

Let now  $\alpha, \beta \in \mathbb{N}^R$  be given with  $|\alpha| = n = |\beta|$ . We implicitly define  $P_\alpha(\beta)$  by

$$Mx^\beta = \sum_{\substack{\alpha \in \mathbb{N}^R \\ |\alpha|=n}} P_\alpha(\beta)x^\alpha.$$

Here  $M$  is the matrix that we introduced in Theorem 9, which means  $M_{s,t} = \chi(st)$  for all  $s, t \in R$  where  $\chi$  is a generating character of  $R$ .

For equivalence relations  $\tau, \theta$  on  $R$  with  $\tau \leq \Phi(\theta)$  we recall the homomorphism  $N : \mathbb{C}[z] \longrightarrow \mathbb{C}[y]$  that we derived from the symmetrizations with respect to  $\tau$  and  $\theta$ , respectively. For  $\alpha \in \mathbb{N}^{R/\theta}$  and  $\beta \in \mathbb{N}^{R/\tau}$  with  $|\alpha| = n = |\beta|$ , we will implicitly define  $Q_\alpha(\beta)$  by

$$Nz^\beta = \sum_{\substack{\alpha \in \mathbb{N}^{R/\theta} \\ |\alpha|=n}} Q_\alpha(\beta)y^\alpha.$$

It can be shown that  $P_\alpha$  and  $Q_\alpha$  are indeed polynomial functions in their arguments, satisfying particular orthogonality relations like those known for the traditional Krawtchouk polynomials. We may refer to these polynomials as *generalized Krawtchouk polynomials*. Summarizing, any pair of equivalence relation  $\theta, \tau$  on the finite Frobenius ring  $R$  with  $\tau \leq \Phi(\theta)$  gives rise to such a family of polynomials.

Let now  $w : R \longrightarrow \mathbb{R}$  be a weight function with  $w(r) = 0$  if  $r = 0$  and  $w(r) > 0$  otherwise. We will give the linear-programming bound on  $A_w(n, d)$  using the equivalence relations  $\theta$  and  $\tau = \Phi(\theta)$  in which we assume that  $w$  is constant on each class in  $R/\tau$ . This is particularly the case if  $\tau$  is induced by  $w$ , i.e.  $r\tau s$  if and only if  $w(r) = w(s)$  for all  $r, s \in R$ .

With the assumed compatibility, we can extend the definition of  $w$  to

$$w : \mathbb{N}^{R/\tau} \longrightarrow \mathbb{R}, \quad \alpha \mapsto \sum_{r\tau \in R/\tau} \alpha_{r\tau} w(r).$$

We emphasize the fact that we do not require any compatibility of  $w$  with the relation  $\theta$ .

**Theorem 15** (Linear programming bound) *Using the notation above, there holds:*

$$A_w(n, d) \leq \max_c \sum_{\substack{\beta \in \mathbb{N}^{R/\tau} \\ |\beta|=n}} c_\beta$$

where the maximum is taken over all  $c : \{\beta \in \mathbb{N}^{R/\tau} : |\beta|=n\} \rightarrow \mathbb{N}$  that satisfy

$$c_\beta \geq 0$$

$$c_\beta = \begin{cases} 1 & \text{if } \beta = n\delta_0 \\ 0 & \text{if } 0 < w(\beta) < d \end{cases} \quad \text{and}$$

$$\sum_{\substack{\beta \in \mathbb{N}^{R/\tau} \\ |\beta|=n}} c_\beta Q_\alpha(\beta) \geq 0 \quad \text{for all } \alpha \in \mathbb{N}^{R/\theta} \text{ with } |\alpha|=n.$$

## 8 Outlook: the Future of Ring-Linear Coding

As a first instance of a ring alphabet equipped with a non-Hamming metric the ring  $\mathbb{Z}_4$  turned out to cover a leading role in ring-linear coding theory. In fact it is probably not surprising that there is a strong emphasis on  $\mathbb{Z}_4$ -linear coding with respect to the Lee weight (cf. Bonnecaze et al. 1997, 2000; Dougherty et al. 2001; Langevin and Solé 2000). Moreover we may view  $\mathbb{Z}_4$  is the ring-linear coding analog of  $\mathbb{Z}_2$  in finite-field linear coding theory.

Apart from the  $\mathbb{Z}_4$ -linear series of codes that led to a discovery of the true role of rings in algebraic coding theory, we mention the so-called Calderbank–McGuire (1997) code, a sporadic  $\mathbb{Z}_4$ -linear example of a code whose Gray image is a binary code that has more codewords than any other known code of the same length and minimum distance.

Regarding other rings, we mention that using the rings  $\mathbb{Z}_8$  and  $\mathbb{Z}_9$  equipped with the homogeneous weights the papers (Duursma et al. 1999, 2001) yield a similar type of outperforming examples by Hensel-lifting the extended binary Golay code and the ternary  $[24, 12, 9]_3$  quadratic residue code. A further example is that the GR(4, 2)-linear code generated by a generator matrix of the  $[8, 4]_4$  Octacode can be used to produce an example of a non-linear code of length 32 over  $\mathbb{F}_4$  (cf. Muegge 2002), that has four times the number of codewords of any previously known  $\mathbb{F}_4$ -code of same length and distance.

Inspired by the success in providing an algebraic representation of good families of codes in terms of ring-linearity, there is a natural interest in further examples of good ring-linear codes. All the examples that we have given might be promising. They should however not deceive the reader about the fact that algebraic coding

theory over rings is in need for many more powerful examples of good ring-linear code and code families.

From the engineering aspect,  $\mathbb{Z}_4$ -linear coding on the basis of the Lee weight has been emphasized (cf. Dougherty et al. 2001; Langevin and Solé 2000; Bonnecaze et al. 1997, 2000), partly because many communications systems use so-called QPSK modulation which is well suited to the Lee weight. What is desirable in practice is to have a weight function that matches the error probabilities in the modulation scheme, that is, higher weight elements of the ring should correspond to errors that are less likely to occur. Finite rings provide a much better variety of weight functions for this task than finite fields.

Another practical aspect is the decoding problem. As to this question the paper (Hammons et al. 1994) gives an algebraic decoding algorithm for the  $\mathbb{Z}_4$ -linear Preparata and Kerdock codes. Further decoders have been developed by Helleseth and others (cf. Helleseth and Kumar 1995; Rong et al. 1999) for the  $\mathbb{Z}_4$ -linear versions of the Goethals and the Goethals–Delsarte codes.

There are a number of contributions (cf. Byrne and Fitzpatrick 2001, 2002) which mainly deal with algorithms decoding a given ring-linear code up to half of its Hamming minimum distance. There is a clear demand however for decoders realizing half of the minimum distance with respect to a given metric on the underlying alphabet. This would at least be a first step if we ignore the fact that for performance reasons contemporary coding theory is mostly interested in decoders that perform beyond half of the minimum distance of a given code.

## 9 Addendum: the Non-commutative Case

The reader might have noticed that restricting to the commutative case is somewhat artificial as for most of the results that we have discussed there are non-commutative analogues in place. The following statements will provide some insight in what changes need to be made in case a non-commutative ring is underlying.

### Rings and modules:

A first important class of rings is that of all Dedekind-finite rings which are defined by the property that these rings do not contain elements which are invertible only on one side. Of course every commutative ring is Dedekind-finite.

The class of all Dedekind-finite rings contains that of all left (or right) Noetherian rings, where we have the mentioned ascending chain condition on the left (or right) side. This class contains the class of all (left) Artinian rings, defined by the descending chain condition (on the left side).

A left Noetherian ring is called a quasi-Frobenius ring if it is left (or right) self-injective. It can be shown that rings of this class are always Artinian and self-injective on both sides (cf. Rowen 1991, p. 347f), but they are not Frobenius yet. A quasi-Frobenius ring  $R$  is called a Frobenius ring if  ${}_R(R/\text{rad}(R)) \cong {}_R\text{soc}({}_RR)$ ,

or, equivalently,  $(R/\text{rad}(R))_R \cong \text{soc}(R_R)_R$  in addition. Both the class of all quasi-Frobenius rings, and that of all Frobenius rings are closed under ring-direct products, group ring constructions and (full) matrix ring constructions. As mentioned earlier, the classes of all quasi-Frobenius rings and all Frobenius rings coincide in the commutative case (cf. Atiyah and MacDonald 1969, Theorem 8.7).

As before, within the class of all local Frobenius rings we emphasize the chain rings. A ring is called a left chain ring if the set of its left ideals forms a chain. It can be shown that every finite left chain ring is a right chain ring at the same time.

### Weight functions:

Call a weight  $w : R \rightarrow \mathbb{R}$  left homogeneous, if  $w(0) = 0$  and the following hold:

- (H1) For all  $x, y \in R$  the equality  $Rx = Ry$  implies  $w(x) = w(y)$ .
- (H2) There exists a real number  $\gamma$  such that for all  $x \in R \setminus \{0\}$  there holds  $\sum_{y \in Rx} w(y) = \gamma |Rx|$ .

It can then be shown that for all  $u \in R^\times$  and  $x \in R$  there holds  $w(xu) = w(x)$ , which means that both symmetry groups of homogeneous weights on rings are maximal. Honold (2001) showed that for a finite Frobenius ring  $R$  every left homogeneous weight on  $R$  is at the same time right homogeneous.

### Linear and cyclic codes:

The possible non-commutativity of the underlying ring  $R$  suggests that we distinguish between left and right  $R$ -linear codes. Note that if  $R$  is a finite quasi-Frobenius ring then all (left)  $R$ -linear codes possess parity-check matrices.

For a duality notion, consider the natural pairing

$${}_R R^n \times {}_{R_R} R^n \longrightarrow R, \quad (x, y) \mapsto xy := \sum_{i=1}^n x_i y_i.$$

If  $C$  is a left  $R$ -linear code of length  $n$  then the dual code  $C^\perp := \{x \in R^n \mid cx = 0 \text{ for all } c \in C\}$  will be right  $R$ -linear.

If  $* : R \rightarrow R$ ,  $x \mapsto x^*$  is an involutory antiautomorphism which is extended to a semilinear antiautomorphism  ${}_R R^n \rightarrow {}_{R_R} R^n$  then we find a one-to-one correspondence between all left linear codes of length  $n$  and all right linear codes of the same length. For all  $C \leq {}_R R^n$  we then have  $C^{\perp *} = C^{*\perp}$ . Let us call the left  $R$ -linear code  $C$  self-dual, if  $C^* = C^\perp$ .

*Example 2* Let  $R$  be the ring of all  $(2 \times 2)$ -matrices over the three element field and consider the code  $C$  defined by the generator matrix

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & 1 & \omega & -\omega \\ \mathbf{0} & 1 & 0 & \omega & 1 & \omega \\ \mathbf{0} & 0 & 1 & -\omega & \omega & 1 \end{bmatrix}, \quad \text{with } \omega = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix},$$

$$\mathbf{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

This code is free of rank 3 over  $R$  and self-dual with respect to the transposition automorphism on  $R$ .

The characterisation of all cyclic splitting codes stays valid also for finite non-commutative rings.

### Foundational results:

Regarding the equivalence notion for algebraic coding theory, there are natural non-commutative versions of the results that we have presented.

**Theorem 16** *For a finite ring  $R$  the following are equivalent:*

- (a)  *$R$  is a Frobenius ring.*
- (b) *Every Hamming isometry between left  $R$ -linear codes of length  $n$  can be extended to a monomial transformation of the ambient space  ${}_R R^n$ .*

Note that the monomial transformation will act from the right side in this case.

Similarly MacWilliams weight enumerator theorem will stay true with slight modifications. It was observed by Wood, for example, that if  $U$  is a subgroup of  $R^\times$  of  $R$  which is central in  $R$  (this means  $ur = ru$  for all  $u \in U$  and  $r \in R$ ), then the induced partition on  $R$  is compatible with the Fourier transform.

But even for  $U = R^\times$  which will not be central in general, we have the insight that one has to distinguish between the left and right actions.

*Example 3* Let  $\theta$  be defined by  $r\theta r'$  if and only if  $R^\times r = R^\times r'$  where  $R^\times$  is the group of units of  $R$ . Then  $s\Phi(\theta)s'$  if and only if  $sR^\times = s'R^\times$ .

### Existence bounds:

All results that we have stated in the respective section of this paper stay true in the non-commutative case.

## References

- M. Aigner, *Combinatorial theory*, Springer, Berlin, 1997.
- Y. Al-Khamees, *The enumeration of finite chain rings*, Panamer. Math. J. **5** (1995), no. 4, 75–81.
- E. F. Assmus Jr. and H. F. Mattson, *Error-correcting codes: An axiomatic approach*, Information and Control **6** (1963), 315–330.
- M. F. Atiyah and I. G. MacDonald, *Introduction to commutative algebra*, Addison–Wesley, London, 1969.
- D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- I. F. Blake, *Codes over certain rings*, Information and Control **20** (1972), 396–404.
- I. F. Blake, *Codes over integer residue rings*, Information and Control **29** (1975), 295–300.
- A. Bonnecaze, P. Solé, C. Bachoc, and B. Mourrain, *Type II codes over  $Z_4$* , IEEE Trans. on Inf. Th. **43** (1997), no. 3, 969–976.

- A. Bonnecaze, E. Rains, and P. Solé, *3-colored 5-designs and  $Z_4$ -codes*, J. Statist. Plann. Inference **86** (2000), no. 2, 349–368.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- E. Byrne and P. Fitzpatrick, *Gröbner bases over Galois rings with an application to decoding alternant codes*, J. Symbolic Comput. **31** (2001), no. 5, 565–584.
- E. Byrne and P. Fitzpatrick, *Hamming metric decoding of alternant codes over Galois rings*, IEEE Trans. on Inf. Th. **48** (2002), no. 3, 683–694.
- E. Byrne and T. Mora, *Gröbner bases over commutative rings and applications to coding theory*, this volume, 2009, pp. 239–261.
- E. Byrne, M. Greferath, and M. E. O'Sullivan, *The linear programming bound for codes over finite Frobenius rings*, Des. Codes Cryptogr. **42** (2007), no. 3, 289–301.
- A. R. Calderbank and G. M. McGuire, *Construction of a  $(64, 2^{37}, 12)$  code via Galois rings*, Des. Codes Cryptogr. **10** (1997), no. 2, 157–165.
- A. R. Calderbank and N. J. A. Sloane, *Modular and  $p$ -adic cyclic codes*, Des. Codes Cryptogr. **6** (1995), no. 1, 21–35.
- C. Carlet,  *$F_{2^k}$ -linear codes*, IEEE Trans. on Inf. Th. **44** (1998), 1543–1547.
- W. E. Clark and D. A. Drake, *Finite chain rings*, Abh. Math. Sem. Univ. Hamburg **39** (1973), 147–153.
- W. E. Clark and J. J. Liang, *Enumeration of finite commutative chain rings*, J. Algebra **27** (1973), 445–453.
- P. M. Cohn, *Algebra I*, Wiley, New York, 1989.
- I. Constantinescu, *Lineare Codes über Restklassenringen ganzer Zahlen und ihre Automorphismen bezüglich einer verallgemeinerten Hamming-Metrik*, Ph.D. thesis, Technische Universität München, 1995.
- S. T. Dougherty, M. Harada, and P. Solé, *Shadow codes over  $\mathbb{Z}_4$* , Finite Fields Appl. **7** (2001), no. 4, 507–529.
- I. M. Duursma, M. Greferath, S. Litsy, and S. E. Schmidt, *A  $F_9$ -linear code inducing a ternary  $(72, 3^{25}, 24)$ -code*, Proc. of OC2001, 1999.
- I. M. Duursma, M. Greferath, S. N. Litsyn, and S. E. Schmidt, *A  $F_8$ -linear lift of the binary Golay code and a nonlinear binary  $(96, 2^{37}, 24)$ -code*, IEEE Trans. on Inf. Th. **47** (2001), no. 4, 1596–1598.
- M. Greferath, *Cyclic codes over finite rings*, Discrete Math. **177** (1997), nos. 1–3, 273–277.
- M. Greferath and M. E. O'Sullivan, *On bounds for codes over Frobenius rings under homogeneous weights*, Discrete Math. **289** (2004), nos. 1–3, 11–24.
- M. Greferath and S. E. Schmidt, *Finite-ring combinatorics and MacWilliams' equivalence theorem*, J. Combin. Theory Ser. A **92** (2000), no. 1, 17–28.
- M. Greferath, A. Nechaev, and R. Wisbauer, *Finite quasi-Frobenius modules and linear codes*, J. Algebra Appl. **3** (2004), no. 3, 247–272.
- A. R. Hammons Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, *The  $Z_4$ -linearity of Kerdock, Preparata, Goethals, and related codes*, IEEE Trans. on Inf. Th. **40** (1994), no. 2, 301–319.
- T. Helleseth and P. V. Kumar, *The algebraic decoding of the  $Z_4$ -linear Goethals codes*, IEEE Trans. on Inf. Th. **41** (1995), no. 6, 2040–2048, part 2.
- T. Honold, *Characterization of finite Frobenius rings*, Arch. Math. (Basel) **76** (2001), no. 6, 406–415.
- P. Kanwar and S. R. López-Permouth, *Cyclic codes over the integers modulo  $p^m$* , Finite Fields Appl. **3** (1997), no. 4, 334–352.
- A. M. Kerdock, *A class of low-rate nonlinear binary codes*, Information and Control **20** (1972a), 182–187.

- A. M. Kerdock, Information and Control **21** (1972b), 395.
- M. Klemm, *Über die Identität von MacWilliams für die Gewichtsfunktion von Codes*, Arch. Math. (Basel) **49** (1987), no. 5, 400–406.
- W. Krull, *Algebraische Theorie der Ringe II*, Math. Ann **91** (1923), 1–46.
- T. Y. Lam, *A first course in noncommutative rings*, Springer, New York, 1991.
- T. Y. Lam, *Lectures on modules and rings*, Springer, Berlin, 1999.
- P. Langevin and P. Solé, *Duadic Z<sub>4</sub>-codes*, Finite Fields Appl. **6** (2000), no. 4, 309–326.
- F. J. MacWilliams, *Combinatorial properties of elementary Abelian groups*, Ph.D. thesis, Radcliffe College, Cambridge MA, 1962.
- F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*, vols. I and II, North-Holland, Amsterdam, 1977.
- E. Martínez-Moro and I. F. Rúa, *Multivariable codes over finite chain rings: serial codes*, SIAM J. Discrete Math. **20** (2006), no. 4, 947–959.
- T. Muegge, *Existenzschranken für Blockcodes über endlichen Frobeniusringen*, Diplomarbeit, 2002.
- A. A. Nechaev, *Kerdock codes in cyclic form*, Discrete Math. Appl. **1** (1991), no. 4, 365–384.
- G. H. Norton and A. Sălăgean, *Strong Gröbner bases and cyclic codes over a finite-chain ring*, International Workshop on Coding and Cryptography (Paris, 2001), Electron. Notes Discrete Math., vol. **6**, Elsevier, Amsterdam, 2001a, p. 11.
- G. H. Norton and A. Sălăgean, *Strong Gröbner bases for polynomials over a principal ideal ring*, Bull. Austral. Math. Soc. **64** (2001b), no. 3, 505–528.
- G. H. Norton and A. Sălăgean, *Gröbner bases and products of coefficient rings*, Bull. Austral. Math. Soc. **65** (2002), no. 1, 145–152.
- G. H. Norton and A. Sălăgean, *Cyclic codes and minimal strong Gröbner bases over a principal ideal ring*, Finite Fields Appl. **9** (2003), no. 2, 237–249.
- F. P. Preparata, *A class of optimum nonlinear double-error correcting codes*, Information and Control **13** (1968), 466–473.
- C. Rong, T. Helleseth, and J. Lahtonen, *On algebraic decoding of the Z<sub>4</sub>-linear Calderbank-McGuire code*, IEEE Trans. on Inf. Th. **45** (1999), no. 5, 1423–1434.
- G.-C. Rota, *On the foundations of combinatorial theory. I. Theory of Möbius functions*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete **2** (1964), 340–368.
- L. H. Rowen, *Ring theory*, Academic, San Diego, CA, 1991.
- A. Sălăgean-Mandache, *On the isometries between Z<sub>p<sup>k</sup></sub> and Z<sub>p</sub><sup>k</sup>*, IEEE Trans. on Inf. Th. **45** (1999), no. 6, 2146–2148.
- C. Satyanarayana, *Lee metric codes over integer residue rings*, IEEE Trans. on Inf. Th. **25** (1979), 250–253.
- P. Shankar, *On BCH codes over arbitrary integer rings*, IEEE Trans. on Inf. Th. **25** (1979), no. 4, 480–483.
- K. Shiromoto, *Singleton bounds for codes over finite rings*, J. Algebraic Combin. **12** (2000), no. 1, 95–99.
- K. Shiromoto and L. Storme, *A Griesmer bound for linear codes over finite quasi-Frobenius rings*, Discrete Appl. Math. **128** (2003), no. 1, 263–274, WCC 2001.
- E. Spiegel, *Codes over Z<sub>m</sub>*, Information and Control **35** (1977), 48–51.
- E. Spiegel, *Codes over Z<sub>m</sub>, revisited*, Information and Control **37** (1978), 100–104.
- R. P. Stanley, *Enumerative combinatorics*, vol. 1, Cambridge University Press, Cambridge, 1997, With a foreword by Gian-Carlo Rota, Corrected reprint of the 1986 original.
- J. A. Wood, *Extension theorems for linear codes over finite rings*, Applied algebra, algebraic algorithms and error-correcting codes (Toulouse, 1997), Springer, Berlin, 1997, pp. 329–340.
- J. A. Wood, *Duality for modules over finite rings and applications to coding theory*, Amer. J. Math. **121** (1999), no. 3, 555–575.

# Gröbner Bases over Commutative Rings and Applications to Coding Theory

Eimear Byrne and Teo Mora

**Abstract** We give a survey of results and applications relating to the theory of Gröbner bases of ideals and modules where the coefficient ring is a finite commutative ring. For applications, we specialize to the case of a finite chain ring. We discuss and compare the main algorithms that may be implemented to compute Gröbner and (in the case of a chain ring) Szekeres-like bases. We give an account of a number of decoding algorithms for alternant codes over commutative finite chain rings.

**Keywords** Commutative rings · Finite chain rings · Galois ring · Gröbner bases · Szekeres-like bases · Buchberger's algorithm · Key-equation · Solution module · Berlekamp–Massey algorithm · FGLM algorithm · Alternant codes · Decoding algorithms · List decoding

## 1 Introduction

The theory of Gröbner bases was introduced by Buchberger in 1965 (Buchberger 1965, 1970, 1985, 1998, 2006). It has been widely studied and extended. A general introduction to the subject can be found in any of Becker and Weispfenning (1993), Cox et al. (1992), Mora (2005). Apart from Buchberger's original algorithm, there are now other algorithms for the computation of a Gröbner basis (cf. Gebauer and Möller 1988; Traverso and Donato 1989; Giovini et al. 1991; Faugère et al. 1993; Faugère 1999, 2002; Brickenstein 2005). Applications of the theory continues to grow. This can be particularly observed in coding theory and in cryptography. For example, in Fitzpatrick (1995) new algorithms corresponding to the Euclidean, Berlekamp–Massey, and Peterson–Gorenstein–Zierler algorithms were derived from the perspective of Gröbner bases, with each as efficient as its classical analogue (Fitzpatrick 1995; Fitzpatrick and Jennings 1998; Guerrini and Rimoldi 2009). The Gröbner basis approach has been applied to rational interpolation problems and to the solution of multivariable congruences (Fitz-

---

E. Byrne

School of Mathematical Sciences, University College, Dublin, Ireland  
e-mail: [ebyrne@ucd.ie](mailto:ebyrne@ucd.ie)

T. Mora

Department of Information Theory, University of Genoa, Genoa, Italy  
e-mail: [theomora@disi.unige.it](mailto:theomora@disi.unige.it)

patrick 1996, 1997, Guerrini and Rimoldi 2009). Very general decoding algorithms using Gröbner bases have been outlined in Mora et al. (2006), Orsini and Sala (2005, 2007), Mora and Orsini (2009). In recent times, the theory has been applied to algebraic cryptanalysis (Faugère and Joux 2003), which attacks cryptosystems based on hidden field equations and relies on solving systems of equations in many variables (Billet and Ding 2009).

Often Gröbner bases and their applications involve solving systems of polynomial equations over a field. They are, however, relevant in more general settings. In Spear (1977), Zacharias (1978), Gröbner bases in  $R[x_1, \dots, x_k]$  are considered for a Noetherian commutative ring  $R$ ; the specialized case where  $R$  is an Euclidean Ring was studied in Kandri-Rody and Kapur (1988), that where  $R$  is a domain in Pan (1989), that where  $R$  is a principal ideal ring in Möller (1988). For applications to coding theory we focus on *special* PIRs (cf. Zariski and Samuel 1958, p. 245), i.e. those PIRs that are commutative finite chain rings. The problem of solving a key equation arises in coding theory as part of a well-known algorithm for decoding an alternant code. Several papers have considered this problem for codes over rings (Interlando et al. 1997; Norton and Sălăgean 2000; Byrne and Fitzpatrick 2002; Byrne 2001, 2002). See Greferath (2009) for a survey on codes over rings. Both Byrne and Fitzpatrick (2002) and Byrne (2002) use Gröbner bases to determine a solution as a minimal element of a sub-module of  $R[x]^2$ , the former computes a Gröbner basis over  $R$ , while the latter computes bases over its residue field. These algorithms correct all errors up to half the minimum distance of the code, the former for the Hamming distance and the latter for the Lee distance.

List decoding includes a variety of procedures that can decode beyond half the minimum distance of a code. First introduced in Elias (1957), a polynomial time list decoding algorithm for RS codes was given in Sudan (1997), and since then many more papers have been published on the subject (Guruswami and Sudan 1999; O’Keeffe and Fitzpatrick 2002; Kötter and Vardy 2003; Ratnakar and Kötter 2005; Roth and Ruckenstein 2000). It turns out that the results of Guruswami and Sudan (1999) and O’Keeffe and Fitzpatrick (2002) extend in part to the ring case, and can be used to decode certain alternant codes over commutative rings (Armand 2005a, 2005b).

## 2 Gröbner Basis over Commutative Rings: the Lost Lore

### 2.1 Notation

This section is a straightforward extension and generalization of the results discussed in Mora (2009). Here we take the same approach and notation. We therefore assume the reader to be familiar with the results in Mora (2009).

$R$  denotes an arbitrary commutative ring with unity<sup>1</sup> and  $\mathcal{Q} := R[X] := R[x_1, \dots, x_n]$  the polynomial ring over the ring  $R$  whose set of terms we denote by

$$\mathcal{T} := \{x_1^{a_1} \cdots x_n^{a_n} : (a_1, \dots, a_n) \in \mathbb{N}^n\}.$$

For a free-module  $\mathcal{Q}^m$ ,  $m \in \mathbb{N}$  endowed with a valuation  $v : \mathcal{Q}^m \rightarrow \mathcal{T}$ , as usual we denote by the symbols  $\{\mathbf{e}_1, \dots, \mathbf{e}_m\}$  its canonical basis,  $\mathcal{T}^{(m)} = \{t\mathbf{e}_i : t \in \mathcal{T}, 1 \leq i \leq m\}$  its monomial  $R$ -basis and we write  $\prec$  to denote a well-ordering on  $\mathcal{T}^{(m)}$  compatible with a fixed term-ordering  $<$  on  $\mathcal{T}$ .

For each  $f = \sum_{\tau \in \mathcal{T}^{(m)}} \mathbf{c}(f, \tau) \tau \in \mathcal{Q}^m$ , its *leading term* is the term  $\mathbf{T}(f) := \max_{\prec}(\text{supp}(f))$ , its *leading coefficient* is  $\text{lc}(f) := \mathbf{c}(f, \mathbf{T}(f))$ , its *leading monomial* is  $\mathbf{M}(f) := \text{lc}(f)\mathbf{T}(f)$  and  $\mathcal{L}(f)$  denotes its *leading form* with respect to the valuation  $v$ .

For any set  $F \subset \mathcal{Q}^m$ , we define the following:

- $\mathbf{T}\{F\} := \{\mathbf{T}(f) : f \in F\}$ ,
- $\mathbf{M}\{F\} := \{\mathbf{M}(f) : f \in F\}$ ,
- $\mathcal{L}\{F\} := \{\mathcal{L}(f) : f \in F\}$ ;
- $\mathbf{T}(F) := \{\tau\mathbf{T}(f) : \tau \in \mathcal{T}, f \in F\}$ ,
- $\mathbf{M}(F) := \mathbb{I}(\mathbf{M}\{F\})$ ,
- $\mathcal{L}(F) := \mathbb{I}(\mathcal{L}\{F\})$ ,
- $\mathbf{N}(F) := \mathcal{T}^{(m)} \setminus \mathbf{T}(F)$ .

Let  $G := \{g_1, \dots, g_s\} \subset \mathcal{Q}^m$ , with  $\mathbf{M}(g_j) := c_j \tau_j \mathbf{e}_{l_j}$ , for each  $j$ . Consider the free module  $\mathcal{Q}^s$ , with canonical basis  $\{\mathbf{e}_1, \dots, \mathbf{e}_s\}$ . We impose the valuation  $v : \mathcal{Q}^s \rightarrow \mathcal{T}$  defined by  $v(\mathbf{e}_j) := \tau_j$  for each  $j$ . Define the map

$$\mathfrak{S} : \mathcal{Q}^s \rightarrow \mathcal{Q}, \quad \sum_{i=1}^s p_i \mathbf{e}_i \mapsto \sum_{i=1}^s p_i g_i.$$

We further define  $\mathfrak{H}(G) := \{\{l_1, l_2, \dots, l_r\} \subseteq \{1, \dots, s\} : \mathbf{e}_{l_1} = \dots = \mathbf{e}_{l_r}\}$  and for each  $H = \{l_1, l_2, \dots, l_r\} \in \mathfrak{H}(G)$  we set

$$\varepsilon_H := \mathbf{e}_{l_1} = \dots = \mathbf{e}_{l_r}, \quad \tau_H := \text{lcm}(\tau_i : i \in H) \quad \text{and} \quad \mathbf{T}(H) := \tau_H \varepsilon_H.$$

Observe that if  $f := \sum_j h_j \mathbf{e}_j \in \ker(\mathfrak{S})$  then denoting

$$\tau \varepsilon := \max_{\prec} \{\mathbf{T}_{\prec}(h_j) \mathbf{T}_{\prec}(g_j)\} \quad \text{and} \quad I := \{j, 1 \leq j \leq s : \mathbf{T}(h_j) \mathbf{T}(g_j) = \tau \varepsilon\}$$

its *leading form*  $\mathcal{L}(f) := \sum_j v_j \mathbf{e}_j \in \mathcal{Q}^s$  of degree  $\tau$  satisfies

- $0 \neq v_j \iff j \in I \quad \text{and} \quad v_j = \mathbf{M}(h_j) =: d_j \omega_j$ ,

---

<sup>1</sup>Most of what is written here can be nearly *verbatim* generalized *cum grano salis* to the non-commutative case. For the sake of simplicity, such easy generalization is not performed here and is left to the interested reader, who could consult (Pritchard 1996) for further details.

- $\sum_{j=1}^s v_j \mathbf{M}_<(g_j) = \sum_{j \in I} (d_j \omega_j) \cdot (c_j \tau_j \mathbf{e}_{l_j}) = (\sum_{j \in I} (d_j c_j) \cdot (\tau_j \omega_j)) \varepsilon = 0,$
- $\sum_{j \in I} d_j \text{lc}(g_j) = 0$  and  $\omega_j \mathbf{T}_<(g_j) = \tau \varepsilon$  for each  $j \in I$ .

**Definition 1** (Compare Mora 2009, Definitions 3, 9 and 10) Let  $\mathbf{N}$  be a finitely generated  $\mathcal{Q}$ -module,  $\Phi : \mathcal{Q}^m \mapsto \mathbf{N}$  be any surjective morphism and let  $\mathbf{M} = \ker \Phi$ . Let  $G = \{g_1, \dots, g_s\} \subset \mathbf{M}$ , with  $\mathbf{M}(g_j) := c_j \tau_j \mathbf{e}_{l_j} \forall j$ ; let  $f, h, f_1, f_2 \in \mathcal{Q}^m$ .

1.  $G$  is called a (*weak*) *Gröbner basis* of  $\mathbf{M}$  if  $\mathbf{M}(G) = \mathbf{M}(\mathbf{M})$ .
2.  $G$  is called a (*strong*) *Gröbner basis* of  $\mathbf{M}$  if for each  $f \in \mathbf{M}$  there is  $g \in \mathbf{M}$  such that  $\mathbf{M}(g) \mid \mathbf{M}(f)$ .
3. We say that  $f$  has a *Gröbner representation*  $\sum_{i=1}^{\mu} p_i g_i$  in terms of  $G$  if

$$f = \sum_{i=1}^{\mu} p_i g_i, \quad p_i \in \mathcal{Q}, g_i \in G, \quad \mathbf{T}(p_i) \mathbf{T}(g_i) \preceq \mathbf{T}(f), \quad \text{for each } i.$$

4. We say that  $f$  has the (*weak*) *Gröbner representation*  $\sum_{i=1}^{\mu} c_i t_i g_i$  in terms of  $G$  if

$$f = \sum_{i=1}^{\mu} c_i t_i g_i, \quad c_i \in \mathbb{F} \setminus \{0\}, t_i \in \mathcal{T}, g_i \in G,$$

with  $\mathbf{T}(f) = t_1 \mathbf{T}(g_1) \succeq \dots \succeq t_i \mathbf{T}(g_i) \succeq \dots$ .

5. We say that  $f$  has the (*strong*) *Gröbner representation*  $\sum_{i=1}^{\mu} c_i t_i g_i$  in terms of  $G$  if

$$f = \sum_{i=1}^{\mu} c_i t_i g_i, \quad c_i \in \mathbb{F} \setminus \{0\}, t_i \in \mathcal{T}, g_i \in G,$$

with  $\mathbf{T}(f) = t_1 \mathbf{T}(g_1) \succ \dots \succ t_i \mathbf{T}(g_i) \succ \dots$ .

6.  $h := \text{NF}_<(f, G)$  is called a *normal form* of  $f$  with respect to  $G$  if

- $f - h \in \mathbb{I}(G)$  has a (*weak*) Gröbner representation in terms of  $G$  and
- $h \neq 0 \implies \mathbf{T}(h) \notin \mathbf{T}(G)$ .

7. The *syzzygy module* of  $G$  is the module

$$\ker(\mathfrak{S}) := \left\{ (p_1, \dots, p_s) : \sum_{i=1}^s p_i g_i = 0 \right\} \subset \mathcal{Q}^s;$$

each of its elements is called a *szyzygy* of  $G$ .

8. any basis  $B \subset \mathbf{M}$  is called a *standard basis* of  $\mathbf{M}$  iff  $\mathcal{L}\{B\}$  generates the *leitmodul*  $\mathcal{L}(\mathbf{M})$  of  $\mathbf{M}$ .

Buchberger Normal Form Algorithm (over a ring)

```

 $(g, \sum_{i=1}^{\mu} c_i t_i g_i) := \text{NormalForm}(f, G)$ 
 $g := f, \mu := 0,$ 
While  $\mathbf{M}(g) \in \mathbf{M}(G)$  do
    Let  $t_j \in \mathcal{T}, c_j \in R, \gamma_j \in G$  such that
    
$$t_j \mathbf{T}(\gamma_j) = \mathbf{T}(g) \forall j, \quad \mathbf{M}(g) = \sum_{j=\mu+1}^{\nu} c_j t_j \mathbf{M}(\gamma_j)$$

    
$$g := g - \sum_{j=\mu+1}^{\nu} c_j t_j g_j, \quad \mu := \nu.$$


```

**Fig. 1** Normal form algorithm

In the notions related to Gröbner bases over a field<sup>2</sup> *strong* Gröbner representations were pinned up in Mora (2009) being the natural result of Buchberger reduction; over an arbitrary unital ring, Buchberger reduction returns *weak* Gröbner representations. For example, for the ideal  $\mathfrak{l} := \langle 2X, 3Y \rangle \subset \mathbb{Z}[X, Y]$ , the set  $A = \{2X, 3Y\}$  is a weak Gröbner basis of  $\mathfrak{l}$  and  $XY = X \cdot 3Y - Y \cdot 2X \in \mathfrak{l}$  has a weak Gröbner representation with respect to  $A$ , but not a strong one. In order to obtain a strong Gröbner bases of  $\mathfrak{l}$  we must add  $XY$  to  $A$ .<sup>3</sup>

The related *Buchberger Normal Form Algorithm* can be properly adapted as in Fig. 1.

Over a ring, the notion of *canonical form* never had practical interest.<sup>4</sup>

The following result by Möller characterizes a Gröbner basis of a module  $\mathbf{M}$ , (compare with Mora 2009, Theorem 15).

**Theorem 1** (Möller 1988) *Let  $\mathbf{M} \subset \mathbb{P}^m$  be a sub-module, and  $\{g_1, \dots, g_s\} =: G \subset \mathbf{M}$ , with  $\mathbf{M}(g_j) := c_j \tau_j \mathbf{e}_{l_j}$ , for each  $j$ ; denoting by  $\mathfrak{GM}$  any homogeneous basis of the syzygy module of  $\mathbf{M}\{G\}$ , the following conditions are equivalent:*

1. *G is a Gröbner basis of M;*
2.  *$f \in \mathbf{M} \iff$  it has a Gröbner representation in terms of G;*
3.  *$f \in \mathbf{M} \iff$  it has a weak Gröbner representation in terms of G;*
4. *for each  $f \in \mathbb{Q}^m \setminus \{0\}$  and any normal form  $h := \text{NF}(f, G)$  of  $f$  with respect to  $G$ ,  $f \in \mathbf{M} \iff h = 0$ ;*
5. *for each  $\phi \in \mathfrak{GM}$ , there is a syzygy  $f_\phi \in \ker(\mathfrak{S})$  of  $G$ , such that  $\mathcal{L}(f_\phi) = \phi$ ;*
6. *for each  $\phi \in \mathfrak{GM}$ ,  $\mathfrak{S}(\phi)$  has a Gröbner representation in terms of G.*

---

<sup>2</sup>The point is that over a field one can assume that each produced polynomial is monic.

The reason why the crucial notion of Gröbner representation is the *strong* one also in a chain ring is Artinianity.

<sup>3</sup>We remark that, *mutatis mutandis* the same example applies also to polynomials over the PIR  $\mathbb{Z}_{12}$ . The difference between a PIR and a *special* PIR is that the latter is a local ring. The example we have built in the (non-special) PIR  $\mathbb{Z}_{12}$  cannot be constructed for the special PIR  $\mathbb{Z}_{p^n}$ .

<sup>4</sup>Membership test has always been solved via normal and not canonical forms; in order to test  $f \equiv g \pmod{\mathbf{M}}$  no reasonable person tests whether  $\text{Can}(f, \mathbf{M}) = \text{Can}(g, \mathbf{M})$  instead of testing whether  $\text{NF}_<(f - g, \mathbf{M}) = 0$ .

```

Buchberger Algorithm (over a ring)

G := GröbnerBasis (F)
G := F := {g1, ..., gs} ,
Let B* be a homogeneous basis of the syzygy module of M{G}
B := B*;
While B ≠ ∅ do
  Choose φ ∈ B , B := B \ {φ} , h := S(φ)
  (h, ∑i=1μ citigi) := NormalForm (h, G)
  If h ≠ 0 then
    s := s + 1, gs := h, G := G ∪ {gs}
    Let C be a set such that B* ∪ C is a homogeneous basis of the syzygy module of M{G}
    B := B ∪ C, B* := B* ∪ C

```

**Fig. 2** Extended Buchberger algorithm

**Corollary 1** *With the same notation and under any of the equivalent conditions of Theorem 1, the set {f<sub>φ</sub> : φ ∈ Sℳ} is a standard basis of ker(S).*

Thus, given a finite basis  $F := \{g_1, \dots, g_s\} \subset M$ , an easy adaptation (Fig. 2) of the Buchberger Algorithm returns a Gröbner basis  $G$  of  $M$ .

As the reader may realize, in this version of Buchberger's Algorithm, Gröbner bases are produced by iteratively forcing condition (6); the difference with Gröbner theory over a *field*<sup>5</sup> is that the notions of (useful) S-polynomials and Gebauer-Möller sets, which were central in Gröbner theory over a *field*, must be interpreted as (*minimal*) *homogeneous basis of the syzygy module of M{G}* in order to play the same rôle in Gröbner theory over a generic *ring with unity*. The problem of course is to devise a procedure which allows to compute such bases.

## 2.2 Zacharias Rings

One of the oldest and most general settings in which Buchberger's algorithm can be applied is for a *Zacharias ring* (Zacharias 1978). The rationale is as follows: Gröbner bases are introduced in  $Q$  in order to test membership and to compute the syzygies of an ideal, thus one can assume that the same computations are performable in the coefficient ring  $R$  and clearly this is required as a precondition.

**Definition 2** A ring  $R$  with identity is called a Zacharias ring if it satisfies the following properties.

1.  $R$  is a Noetherian ring.
2. There is an algorithm such that for each  $c \in R$ , non-empty set  $C = \{c_1, \dots, c_t\} \subset R \setminus \{0\}$ , decides whether or not  $c \in I(C)$ , in which case it produces elements  $d_i \in R$  satisfying  $c = \sum_{i=1}^t c_i d_i$ .
3. There is an algorithm such that given  $C := \{c_1, \dots, c_t\} \subset R \setminus \{0\}$ , computes a finite set of generators for the syzygy  $R$ -module of  $C$ .

---

<sup>5</sup>Or, say, over a *principal ideal domain* (see Sect. 2.3).

**Proposition 1** (Zacharias 1978) Let  $G := \{g_1, \dots, g_s\} \subset M$ , with  $\mathbf{M}(g_j) := c_j \tau_j \mathbf{e}_{l_j}$ , for each  $j$ . Let  $T := \{\text{lcm}\{\tau_h : h \in H\}, H \in \mathfrak{H}(G)\}$  and for any  $m \in T$ ,  $i \in \{1, \dots, s\}$ , let us define

$$v(m)_i := \begin{cases} c_i & \text{if } T(g_i) \mid m \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad t_i(m) := \begin{cases} \frac{m}{T(g_i)} & \text{if } T(g_i) \mid m \\ 1 & \text{otherwise.} \end{cases}$$

Let  $C(m) \subset R^s$  be a finite basis of the syzygy module of  $\{v(m)_1, \dots, v(m)_s\}$  and set

$$S(m) := \{(c_1 t_1(m), \dots, c_s t_s(m)) : (c_1, \dots, c_s) \in C(m)\}.$$

Then  $S(G) := \bigcup_{m \in T} S(m)$  is a homogeneous basis of the syzygy module of  $\mathbf{M}[G]$ .

**Corollary 2** (Zacharias 1978) If  $R$  is a Zacharias ring, then it is possible to compute, via the algorithm of Fig. 2, a Gröbner basis of each given module  $\mathbb{I}(F) \subset \mathcal{Q}^m$ .

**Corollary 3** (Zacharias 1978) If  $R$  is a Zacharias ring, then  $\mathcal{Q}$  is a Zacharias ring.

*Proof* Condition (1) is trivial. Once a Gröbner basis  $G$  of a module  $\mathbb{I}(F) \subset \mathcal{Q}^m$  is computed via the algorithm of Fig. 2, Condition (2), i.e. membership testing, is granted by applying the algorithm of Fig. 1. Moreover, the computation of a Gröbner basis  $G$  returns a basis of the syzygy module of  $G$ . Since we have explicit linear representations of  $F$  in terms of  $G$  and conversely, elementary linear algebra allows to obtain also a basis of the syzygy module of  $F$ , giving Condition (3).  $\square$

### 2.3 Möller: Gröbner Basis over a Principal Ideal Ring

Concluding a mainstream of research (Kandri-Rody and Kapur 1988; Pan 1989), Möller extended Gröbner bases to PID's and PIR's by generalizing to them the construction and the main properties of Gebauer–Möller sets, as follows: let us assume that  $R$  is a *principal ideal ring* and for each  $H \subset \mathfrak{H}(G)$  let us also denote  $c_H := \text{lcm}(\text{lc}(h) : h \in H)$ ,  $M(H) := c_H \tau_H$  and

$$\mathbf{M}(H) = c_H \mathbf{T}(H) = c_H \tau_H \varepsilon_H = M(H) \varepsilon_H.$$

For each  $i, j, 1 \leq i < j \leq s$ ,  $\mathbf{e}_{l_i} = \mathbf{e}_{l_j}$  we set

$$\begin{aligned} b(i, j) &:= \frac{M(\{i, j\})}{M(i)} \mathbf{e}_i - \frac{M(\{i, j\})}{M(j)} \mathbf{e}_j \in \mathcal{Q}^s, \\ B(i, j) &:= \frac{M(\{i, j\})}{M(i)} g_i - \frac{M(\{i, j\})}{M(j)} g_j = \frac{\text{lcm}(c_i, c_j)}{c_i} \frac{\text{lcm}(\tau_i, \tau_j)}{\tau_i} g_i \\ &\quad - \frac{\text{lcm}(c_i, c_j)}{c_j} \frac{\text{lcm}(\tau_i, \tau_j)}{\tau_j} g_j, \end{aligned}$$

so that  $B(i, j) = \mathfrak{S}(b(i, j))$ . Also for each  $j$ , denote by  $a_j \in R$  the annihilator of  $\mathbb{I}(c_i)$ . Then

**Proposition 2** (Möller 1988) *The set  $\{b(i, j) : 1 \leq i < j \leq s, \mathbf{e}_{l_i} = \mathbf{e}_{l_j}\} \cup \{a_j \mathbf{e}_j\} \subset \mathcal{Q}^s$  is a homogeneous basis of the syzygy module of  $\mathbf{M}(G)$ .*

**Lemma 1** (Buchberger's First Criterion) *If  $\mathbf{M}$  is an ideal of  $\mathcal{Q}$ , there holds*

$$\mathbf{M}(i)\mathbf{M}(j) = \mathbf{M}(i, j) \implies \text{NF}(B(i, j), G) = 0.$$

**Definition 3** Let  $\mathfrak{B} := \{\{i, j\} : 1 \leq i < j \leq s, B(i, j) \text{ exists}\}$  and let

$$\mathfrak{B}_1 := \begin{cases} \{\{i, j\} : \mathbf{M}(i)\mathbf{M}(j) = \mathbf{M}(i, j)\} & \text{if } \mathbf{M} \text{ is an ideal,} \\ \{\emptyset\} & \text{otherwise.} \end{cases}$$

A subset  $\mathfrak{GM} \subset \mathfrak{B} \setminus \mathfrak{B}_1$  is called a *Gebauer–Möller set* for  $G$  if the set

$$\{b(i, j) : \{i, j\} \in \mathfrak{GM} \cup \mathfrak{B}_1\} \cup \{a_j \mathbf{e}_j, j \leq s\}$$

is a homogeneous basis of the syzygy module of  $\mathbf{M}(G)$ .

**Lemma 2** (Möller) *For each  $i, j, k : 1 \leq i, j, k \leq s, \mathbf{e}_{l_i} = \mathbf{e}_{l_j} = \mathbf{e}_{l_k}$ , there holds*

$$\frac{M(i, j, k)}{M(i, k)} B(i, k) - \frac{M(i, j, k)}{M(i, j)} B(i, j) + \frac{M(i, j, k)}{M(k, j)} B(k, j) = 0.$$

**Proposition 3** (Möller 1988) *Let  $\mathfrak{GM}_* \subset \{b(i, j), 1 \leq i < j < s\}$  be a Gebauer–Möller set for  $\{g_1, \dots, g_{s-1}\}$ , let*

$$\mathfrak{B}_2 := \{b(i, j) \in \mathfrak{GM}_* : \mathbf{M}(i, j, s) = \mathbf{M}(i, j), \mathbf{M}(i, s) \neq \mathbf{M}(i, j) \neq \mathbf{M}(j, s)\},$$

let  $\overline{M} := \{\mathbf{M}(j, s) : 1 \leq j < s\}$  and  $\overline{M}' \subset \overline{M}$  be the set of the elements  $\sigma \in \overline{M}$  such that either

- exists  $\sigma' \in \overline{M} : \sigma' \mid \sigma \neq \sigma'$  or
- (in the case that  $\mathbf{M}$  is an ideal) exists  $i_\sigma : 1 \leq i_\sigma < s, \mathbf{M}(i_\sigma)\mathbf{M}(s) = \mathbf{M}(i_\sigma, s) = \sigma$ ;

for each  $\sigma \in \overline{M} \setminus \overline{M}'$  choose  $i_\sigma, 1 \leq i_\sigma < s$ , such that  $\mathbf{M}(i_\sigma, s) = \sigma$  and define

$$\mathfrak{B}_3(G) := \{b(i_\sigma, s) : \sigma \in \overline{M} \setminus \overline{M}'\}.$$

Then  $(\mathfrak{GM}_* \setminus \mathfrak{B}_2) \cup \mathfrak{B}_3(G)$  is a Gebauer–Möller set for  $G$ .

The reader has thus being exposed to the esoteric revelation of Möller (1988), that the exoteric version of Buchberger's Algorithm proposed in Fig. 5 of Mora (2009) applies nearly *verbatim* also to polynomial rings over a principal ideal domain, provided that each  $\mathbf{T}(\cdot)$  is substituted with the corresponding  $\mathbf{M}(\cdot)$ , and indeed over a principal ideal ring if in addition annihilators of leading coefficients are properly disposed of.

## 2.4 Spear's Theorem

Local rings are now easily dealt with by a folklore result, probably due to Spear (1977), which was well-known to the computer algebra community already in the Eighties and which, as Möller's result, has been removed from the exoteric lore of Gröbner bases.

Let  $\mathfrak{l} \subset \mathcal{Q}$  be an ideal, let  $A := \mathcal{Q}/\mathfrak{l}$  and  $\Pi : \mathcal{Q} \mapsto A$  the canonical projection; let  $J \subset A^m$  be a submodule and let  $J' := \Pi^{-1}(J) \subset \mathcal{Q}^m$ .

**Theorem 2** (Spear) *With the present notation, let  $\{\mathbf{e}_1, \dots, \mathbf{e}_m\}$  be the canonical basis of both  $\mathcal{Q}^m$  and  $A^m$  we have*

1. *If  $B = \{g_1, \dots, g_s\}$  is a Gröbner basis of  $J'$ , then*

$$\{\Pi(g) : g \in B, \mathbf{T}(g) \notin \mathbf{T}(\mathfrak{l})\}$$

*is a Gröbner basis of  $J$ .*

2. *If  $C$  is a Gröbner basis of  $\mathfrak{l}$  and  $D \subset J'$  is a set such that*

- *for each  $g \in D$ ,  $\Pi(g) \neq 0$ , and  $\Pi(\mathbf{T}(g)) = \mathbf{T}(\Pi(g))$ ,*
- *$\{\Pi(g) : g \in D\}$  is a Gröbner basis of  $J$ ,*

*then  $\{f\mathbf{e}_j, f \in C, 1 \leq j \leq s\} \cup D$  is a Gröbner basis of  $J'$ .*

## 2.5 Szekeres Ideals

Mainly in connection with special PIR's, it is worthwhile to recall and extend an interesting pre-Gröbner concept introduced by Szekeres (Lauer 1976; Szekeres 1952) that has already proved fruitful in studying the structure of Gröbner bases over rings (Apel 2000; Assi 1991) and which will be useful for interpreting the algorithms we describe later. Let  $M$  be a  $\mathcal{Q}$ -submodule of  $\mathcal{Q}^m$ . For each  $\tau \in \mathcal{T}^{(m)}$  we define the ideal  $\mathfrak{l}_\tau := \{\text{lc}(f) : f \in M, \mathbf{T}(f) = \tau\} \cup \{0\} \subset R$  and for each ideal  $\mathfrak{a} \triangleleft R$  we define the semigroup ideal  $T_\mathfrak{a} := \{\tau \in \mathbf{T}(M) : \mathfrak{l}_\tau \supset \mathfrak{a}\} \subset \mathcal{T}^{(m)}$ .

Clearly we have, for terms  $\tau, \omega \in \mathcal{T}^{(m)}$  and ideals  $\mathfrak{a}, \mathfrak{b} \triangleleft R$ , the relations

$$\tau \mid \omega \implies \mathfrak{l}_\tau \subset \mathfrak{l}_\omega \quad \text{and} \quad \mathfrak{a} \supset \mathfrak{b} \implies T_\mathfrak{a} \subset T_\mathfrak{b}.$$

Now suppose that  $R$  is a PIR. For each  $\tau \in \mathcal{T}^{(m)}$ , let  $c_\tau \in R$  denote an arbitrary fixed generator of  $\mathfrak{l}_\tau$  and let  $f_\tau \in M$  be an arbitrary fixed element satisfying  $\mathbf{M}(f) = c_\tau \tau$ .<sup>6</sup>

**Definition 4** Let  $R$  be a PIR. For each ideal  $\mathfrak{a} \triangleleft R$  let  $\mathbf{G}_\mathfrak{a}$  denote the minimal basis of  $T_\mathfrak{a}$ . We define a *Szekeres-like basis* of  $M$  to be a set of the form

$$\mathbf{S}(M) := \{f_\tau : \tau \in \mathbf{G}_\mathfrak{a}, \mathfrak{a} \triangleleft R\}.$$

---

<sup>6</sup>With a slight abuse of notation we define  $c_\tau := f_\tau := 0$  iff  $\tau \in N(M)$ .

The Szekeres-like basis  $\mathbf{S}(M)$  is not a minimal strong Gröbner basis of  $M$  itself but a minimal strong Gröbner basis of  $M$  can be easily deduced from it by removing from it all elements  $f$  for which there is a  $g \in \mathbf{S}(M)$  such that  $M(g) | M(f)$ .

### 3 Finite Chain Rings

We now give a brief description of the notions and properties of commutative finite chain rings (cf. Gilmer 1972; McDonald 1974; Zariski and Samuel 1958). A finite chain ring  $R$  is a unital ring whose ideals can be linearly ordered to form a finite chain with respect to inclusion. Thus a finite chain ring is a local ring, and is a principal ideal ring.

Examples of finite chain rings include the integer modular rings  $\mathbb{Z}_{p^n}$ , the Galois rings  $GR(p^n, r)$  of  $p^{nr}$  elements and characteristic  $p^n$  and the quotient rings  $T[x]/\langle x^s + p, p^{n-1}x^t \rangle$  where  $p$  is a prime,  $T = GR(p^n, r)$ , and  $n, r, s, t$  are integers such that  $(p, s) = 1$ .

For the remainder, unless stated otherwise, the symbol  $R$  will denote a (commutative) finite chain ring,  $\mathfrak{p}$  its unique maximal ideal and  $\pi$  a generator of  $\mathfrak{p}$ . Then  $\mathfrak{p}$  is nilpotent in  $R$ , say with nilpotency  $n$  and the finite chain takes the form

$$\{0\} = \mathfrak{p}^n \triangleleft \mathfrak{p}^{n-1} \triangleleft \cdots \triangleleft \mathfrak{p}^2 \triangleleft \mathfrak{p} \triangleleft R.$$

The set  $R^* := R \setminus \mathfrak{p}$  will denote the group of units of  $R$ ,  $k_R$  its residue field and  $\mu$  the natural epimorphism from  $R$  onto  $k_R$ . We also use the symbol  $\mu$  to denote the obvious extension of this map to any  $R$ -module.

Given any  $\theta \in R$ , there exist  $u \in R^*$ , and a unique non-negative integer  $v(\theta)$  such that  $\theta = u\pi^{v(\theta)}$ . With respect to this notation  $\theta$  has nilpotency  $n - v(\theta)$ .

For computational purposes, we will assume that a Gröbner basis  $G = \{g_1, \dots, g_s\}$  over  $R$  is *minimal*, so that  $M(g_i)$  does not divide  $M(g_j)$  for  $i \neq j$  and that  $\text{lc}(g_i) = c_i = \pi^{\ell_i}$ ,  $1 \leq \ell_i = v(\text{lc}(g_i)) < n$  (for the finite field case, this reduces to the assumption that  $\text{lc}(g_i) = 1$  for each  $i$ ).

Möller's result gives for  $R$  *verbatim* the version of Buchberger's Algorithm described in Mora (2009, Fig. 5). In the case of a finite chain ring, in each loop, it computes the normal forms of

- the (useful!) S-pairs  $B(i, j) := \pi^{t_j - t_i} \frac{\text{lcm}(\tau_i, \tau_j)}{\tau_i} g_i - \frac{\text{lcm}(\tau_i, \tau_j)}{\tau_j} g_j$  with  $t_i \leq t_j$  and
- the annihilator-pairs  $\pi^{t_i} g_i$ .

We now restrict to arbitrary submodules of  $R[x]$  and  $R[x]^2$ . The Szekeres-like bases that we describe shortly in general are not minimal, each carrying redundant polynomials. However, it is trivial to obtain a minimal Gröbner basis from such a basis, and any Szekeres-like basis can be obtained from a Gröbner basis  $\mathcal{G}$ , for example by augmenting appropriate  $\pi^u$ -multiples of the elements of  $\mathcal{G}$ .

Gröbner bases of ideals in  $R[x]$  are easy to describe: each has the form  $\{g_i : i \in I\}$  where  $I \subset \{0, \dots, n-1\}$ ,  $M(g_i) = \pi^i x^{s_i}$  and  $s_i > s_j$  for  $i < j$ . On the other hand, any Szekeres-like basis has the form  $\{g_i, g_{i+1}, \dots, g_{n-1}\}$  with  $M(g_i) = \pi^i x^{s_i}$  and  $s_i \leq s_{i+1} \leq \dots \leq s_{n-1}$ .

**Lemma 3** Let  $B$  be an  $R[x]$ -submodule of  $R[x]^2$  generated by a set of monomials. Then there exist nonnegative integers  $\iota$  and  $\kappa$  such that  $B$  has a Szekeres-like basis which takes exactly one of the following forms

- I  $\{(\pi^i x^{s_i}, 0) : i \in \{\iota, \dots, n-1\}\}$
- II  $\{(0, \pi^j x^{t_j}) : j \in \{\kappa, \dots, n-1\}\}$
- III  $\{(\pi^i x^{s_i}, 0), (0, \pi^j x^{t_j}) : i \in \{\iota, \dots, n-1\}, j \in \{\kappa, \dots, n-1\}\}$

where  $s_i \leq s_j$  for all  $i, j \in \{\iota, \dots, n-1\}$  with  $j \leq i$ , and  $t_i \leq t_j$  for all  $i, j \in \{\kappa, \dots, n-1\}$  with  $j \leq i$ . Moreover, any Szekeres-like basis of  $B$  in the form I, II or III is unique.

The extension to arbitrary  $R[x]$ -submodules of  $R[x]^2$  is immediate:

**Theorem 3** Let  $A$  be an  $R[x]$ -submodule of  $R[x]^2$ . Then there exist nonnegative integers  $\iota$  and  $\kappa$  such that  $A$  has a Szekeres-like basis that takes exactly one of the following forms

- I  $\{(a_\iota, b_\iota), \dots, (a_{n-1}, b_{n-1})\}$
- II  $\{(c_\kappa, d_\kappa), \dots, (c_{n-1}, d_{n-1})\}$
- III  $\{(a_\iota, b_\iota), \dots, (a_{n-1}, b_{n-1}), (c_\kappa, d_\kappa), \dots, (c_{n-1}, d_{n-1})\}$

where

- (i) for all  $i \in \{\iota, \dots, n-1\}$  and  $j \in \{\kappa, \dots, n-1\}$ ,  $\mathbf{M}(a_i, b_i) = (\pi^i x^{s_i}, 0)$  and  $\mathbf{M}(c_j, d_j) = (0, \pi^j x^{t_j})$  for some nonnegative integers  $s_i$  and  $t_j$ ,
- (ii)  $s_i \leq s_j$  for all  $i, j \in \{\iota, \dots, n-1\}$  with  $j \leq i$ , and  $t_i \leq t_j$  for all  $i, j \in \{\kappa, \dots, n-1\}$  with  $j \leq i$ .

**Definition 5** Let  $A$  be an  $R[x]$ -submodule of  $R[x]^2$  of type III, and let  $\mathbf{M}(A)$  have Szekeres-like basis

$$\{(\pi^\iota x^{s_\iota}, 0), \dots, (\pi^{n-1} x^{s_{n-1}}, 0), (0, \pi^\kappa x^{t_\kappa}), \dots, (0, \pi^{n-1} x^{t_{n-1}})\}$$

for some integers  $\iota, \kappa \in \{0, \dots, n-1\}$  where  $s_\iota \geq s_1 \geq \dots \geq s_{n-1}$  and  $t_\kappa \geq t_1 \geq \dots \geq t_{n-1}$ . The *vector of minimal exponents* of  $A$ , denoted by  $\text{vme}(A)$ , is the vector of length  $2n - (\iota + \kappa)$  defined by

$$(s_\iota, \dots, s_{n-1}, t_\kappa, \dots, t_{n-1}).$$

## 4 Solving a Key Equation

We consider here how to extend to the case of a finite commutative chain ring the FGLM-like algorithms of Guerrini and Rimoldi (2009), which may be used to solve the polynomial congruence

$$aS \equiv b \pmod{x^r},$$

for some  $a, b \in R[x]$ , subject to certain degree constraints, given  $S$  in  $R[x]$  and a positive integer  $r$ . Such an equation is called a *key equation*. One approach towards its solution is to compute a Gröbner or Szekeres-like basis for the so-called solution module

$$M = \{(a, b) : aS \equiv b \pmod{x^r}\}.$$

This may also be viewed as a linear recurrence problem, in which case a modified Berlekamp–Massey algorithm may be applied (Interlando et al. 1997; Norton 1999; Norton and Sălăgean 2000).

**Definition 6** For each integer  $\ell$  consider the valuation  $v : R[x]^2 \rightarrow \mathcal{T}$  defined by  $v(\mathbf{e}_1) := x^\ell$ ,  $v(\mathbf{e}_2) := 1$  and define the term order  $<_\ell$  on  $R[x]^2$  as

$$x^i \mathbf{e}_{l_i} <_\ell x^j \mathbf{e}_{l_j} \iff v(x^i \mathbf{e}_{l_i}) < v(x^j \mathbf{e}_{l_j}) \quad \text{or} \quad v(x^i \mathbf{e}_{l_i}) = v(x^j \mathbf{e}_{l_j})$$

$$\text{and } l_j > l_i$$

so that, in particular

- (i)  $(x^i, 0) <_\ell (x^j, 0)$  and  $(0, x^i) <_\ell (0, x^j)$  for  $i < j$
- (ii)  $(0, x^j) <_\ell (x^i, 0)$  if and only if  $v((0, x^j)) = x^j \leq x^{i+\ell} = v((x^i, 0))$ , if and only if  $j \leq i + \ell$ .

It is easy to see that solution module  $M$  is generated by the set  $\{(1, S), (0, x^r)\}$ . Then of course the Zacharias–Möller algorithm adapting (Mora 2009, Fig. 5) could be used to compute a Gröbner basis from the generating set  $\{(1, S), (0, x^r)\}$  with respect to the term order  $<_\ell$ , but the complexity associated with this procedure is exponential in  $r$ .

Instead we propose the following algorithm, which we present below. This is an instance of an FGLM-like algorithm and is an adaptation of Fitzpatrick's algorithm (Fitzpatrick 1995, Sect. IV) where the coefficient ring is assumed to be a finite field, to the case of finite commutative chain ring. The complexity of the algorithm, like its Berlekamp–Massey equivalents, is quadratic in  $r$  (Byrne and Fitzpatrick 2002, Sect. VII).

We introduce some more notation. Let  $M^{(k)}$  denote the module of solutions to the key equation modulo  $x^k$ , i.e.

$$M^{(k)} = \{(a, b) : aS \equiv b \pmod{x^k}\}.$$

We get a sequence of modules

$$R[x]^2 = M^{(0)} \supset M^{(1)} \supset \cdots \supset M^{(r)} = M,$$

and since  $(0, x^k) \in M^{(k)} \setminus M^{(k+1)}$ , we observe that the chain above is strictly decreasing.

The algorithm proceeds by constructing a Szekeres-like basis of  $M^{(k+1)}$  from a Szekeres-like basis of  $M^{(k)}$  until after  $r$  iterations a basis of the solution module  $M$  is arrived at. To do this, we need the notion of a  $k$ th *discrepancy*.

Given a polynomial  $h \in R[x]$ , we use the symbol  $[h]_k$  to denote the coefficient attached to the term  $x^k$  in  $h$ . Given an element  $(f, g) \in R[x]^2$ , the  $k$ th discrepancy of  $(f, g)$  is given by  $[fS - g]_k$ . If  $(f, g)$  and  $(f', g')$  in  $M^{(k)}$  have  $k$ th discrepancies  $\alpha$  and  $\alpha'$ , respectively, such that  $\alpha + \theta\alpha' = 0$  for some  $\theta \in R$  then  $(f, g) + \theta(f', g')$  is contained in  $M^{(k+1)}$ , having  $k$ th discrepancy zero. In other words, we can construct an element of  $M^{(k+1)}$  given a pair of suitable elements of  $M^{(k)}$ .

Note that since  $(x^k, 0)$  and  $(0, x^k)$  are contained in  $M^{(k)}$  for each  $k$ ,  $M^{(k)}$  is a type III module with  $\iota = \kappa = 0$  and its vector of minimal exponents has the form  $\text{vme}(M^{(k)}) = (s_0, \dots, s_{n-1}, t_0, \dots, t_{n-1})$ .

**Definition 7** Let  $k \in \{0, \dots, r\}$  and let  $M^{(k)}$  have Szekeres-like basis  $\mathcal{B}_k$ . For each  $(f, g) \in \mathcal{B}_k$  we denote by  $\mathcal{Z}_k(f, g)$  the set of  $k$ th discrepancies of elements of  $\mathcal{B}_k$  with leading terms less than  $\mathbf{T}(f, g)$ , that is,

$$\mathcal{Z}_k(f, g) = \{[f'S - g']_k : (f', g') \in \mathcal{B}_k, \mathbf{T}(f', g') < \mathbf{T}(f, g)\}.$$

We also define

$$\mathcal{B}_k(f, g) := \{(f', g') \in \mathcal{B}_k : \mathbf{T}(f', g') < \mathbf{T}(f, g)\}.$$

The rules for updating each basis  $\mathcal{B}_k$  are based on the following result.

**Theorem 4** Let  $\iota, \kappa \in \{0, \dots, n-1\}$ , let  $\text{vme}(M^{(k)}) = (s_\iota, \dots, s_{n-1}, t_\kappa, \dots, t_{n-1})$  and let  $M^{(k)}$  have Szekeres-like basis  $\mathcal{B}_k = \{(a_\iota, b_\iota), \dots, (a_{n-1}, b_{n-1}), (c_\kappa, d_\kappa), \dots, (c_{n-1}, d_{n-1})\}$ , where, for each  $\iota \leq i \leq n-1, \kappa \leq j \leq n-1$  we have  $\mathbf{M}(a_i, b_i) = (\pi^i x^{s_i}, 0), \mathbf{M}(c_j, d_j) = (0, \pi^j x^{t_j})$ . Then

1.  $\text{vme}(M^{(k+1)}) = (s'_\iota, \dots, s'_{n-1}, t'_\kappa, \dots, t'_{n-1})$  for some nonnegative integers  $s'_i, t'_j$  satisfying  $s_i \leq s'_i \leq s_i + 1$  and  $t_j \leq t'_j \leq t_j + 1$ .
2. For each  $i, j$ , let  $\alpha_i = [a_i S - b_i]_k, \beta_j = [c_j S - d_j]_k$ . Let  $\ell \in \{\iota, \dots, n-1\}$  (resp.  $\{\kappa, \dots, n-1\}$ ). There exist  $\theta \in R$  and  $\zeta \in \mathcal{Z}_k(a_\ell, b_\ell)$  (resp.  $\mathcal{Z}_k(c_\ell, d_\ell)$ ) satisfying  $\alpha_\ell + \theta\zeta = 0$  (resp.  $\beta_\ell + \theta\zeta = 0$ ) if and only if  $s'_\ell = s_\ell$  (resp.  $t'_\ell = t_\ell$ ).

Given a Szekeres-like basis  $\mathcal{B}_k = \{(a_\iota, b_\iota), \dots, (a_{n-1}, b_{n-1}), (c_\kappa, d_\kappa), \dots, (c_{n-1}, d_{n-1})\}$  for  $M^{(k)}$ , we compute a Szekeres-like basis  $\mathcal{B}_{k+1} = \{(a'_\iota, b'_\iota), \dots, (a'_{n-1}, b'_{n-1}), (c'_\kappa, d'_\kappa), \dots, (c'_{n-1}, d'_{n-1})\}$  for  $M^{(k+1)}$  as follows. For simplicity we let  $(f_i, g_i)$  denote either  $(a_i, b_i)$  or  $(c_i, d_i)$ .

1. For each  $i$ , compute the  $k$ th discrepancies  $\zeta_i = \pi^{v(\zeta_i)} \varepsilon_i = [f_i S - g_i]_k$ .
2. For each  $\ell$ , we obtain exactly one element of  $\mathcal{B}_{k+1}$  from  $(f_\ell, g_\ell)$  as follows.
  - (a) If  $\zeta_\ell = 0$  then  $(f'_\ell, g'_\ell) := (f_\ell, g_\ell)$ .
  - (b) If  $\zeta_\ell \neq 0$  and there is some  $(f_j, g_j) \in \mathcal{B}_k(f, g)$  satisfying  $v(\zeta_j) \leq v(\zeta_\ell)$  then

$$(f'_\ell, g'_\ell) := (f_\ell, g_\ell) + \pi^{v(\zeta_\ell) - v(\zeta_j)} \varepsilon_j^{-1} \varepsilon_\ell (f_j, g_j).$$

- (c) Otherwise,  $(f'_\ell, g'_\ell) := (x f_\ell, x g_\ell)$ .

Let  $\text{vme}(M^{(k+1)}) = (s'_\ell, \dots, s'_{n-1}, t'_k, \dots, t'_{n-1})$ . If  $(f'_\ell, g'_\ell)$  is defined by 2 (a) or (b) and  $\mathbf{M}(f'_\ell, g'_\ell)$  is on the left (resp. right) then Theorem 4 implies that  $s'_\ell = s_\ell$ , (resp.  $t'_\ell = t_\ell$ ) so  $(f'_\ell, g'_\ell)$  is contained in a Szekeres-like basis for  $M^{(k+1)}$ . Otherwise  $(f'_\ell, g'_\ell)$  is defined by 2 (c) and  $s'_\ell = s_\ell + 1$ , (resp.  $t'_\ell = t_\ell + 1$ ), so that  $(f'_\ell, g'_\ell) = (x f_\ell, x g_\ell)$  is contained in a Szekeres-like basis for  $M^{(k+1)}$ . Thus the elements of the new set  $\mathcal{B}_{k+1}$  form a Szekeres-like basis for  $M^{(k+1)}$ . We summarize the above as follows.

**Theorem 5** Let  $k \in \{0, \dots, r - 1\}$ , let  $\text{vme}(M^{(k+1)}) = (s'_\ell, \dots, s'_{n-1}, t'_k, \dots, t'_{n-1})$ , and let  $M^{(k)}$  have Szekeres-like basis  $\mathcal{B}_k$  as in Theorem 4. Then the set

$$\mathcal{B}_{k+1} = \{(a'_i, b'_i), \dots, (a'_{n-1}, b'_{n-1}), (c'_\kappa, d'_\kappa), \dots, (c'_{n-1}, d'_{n-1})\}$$

with elements  $(a'_i, b'_i)$  and  $(c'_i, d'_i)$  constructed as above, is a Szekeres-like basis of  $M^{(k+1)}$  satisfying  $\mathbf{M}(a'_i, b'_i) = (\pi^i x^{s'_i}, 0)$  and  $\mathbf{M}(c'_j, d'_j) = (0, \pi^j x^{t'_j})$  for each  $i \in \{\ell, \dots, n-1\}$ ,  $j \in \{\kappa, \dots, n-1\}$ .

We remark that one advantage of this approach over direct generalizations of the BM method is that it is easy to see that each update is well-defined; even if there is no solution to the discrepancy equation that arises in 2 (b) we can still determine a valid update.

The required solution  $(a, b)$  for the key equation generally must satisfy certain degree constraints. If we can choose a term order  $<_\ell$  such that  $\mathbf{T}(a, b)$  is minimal in  $\mathbf{T}(M)$ , then  $(a, b)$  must be contained in a Gröbner (and hence Szekeres-like) basis of  $M$ .

For each  $k \in \{0, \dots, n\}$ , let  $M_k := M \cap \mathfrak{p}^k$  and let  $L_k := M \setminus M_k$  be the set-theoretic complement of  $M_k$  in  $M$ . Let  $\mathcal{G}_k$  be a reduced Gröbner basis of  $M_k$ . Clearly  $M/M_k = \{\text{Can}((a, b), \mathcal{G}_k) + M_k : (a, b) \in R[x]^2\}$ . If an element has the same canonical form as an element in some set then we say that it is contained *up to equivalence* in that set. Therefore, if  $(a, b) \in M$  with  $\mathbf{T}(a, b)$  minimal for some term order, then  $(a, b)$  is contained up to equivalence in a Gröbner basis of  $M$ . Moreover, if  $\mathbf{T}(a, b) = \mathbf{T}(a', b')$  is minimal in  $\mathbf{T}\{L_k\}$  then  $(a, b)$  and  $(a', b')$  both have the same canonical form with respect to  $\mathcal{G}_k$ . Of course any element  $(a, b)$  in a Szekeres-like basis of  $M$  with  $\mathbf{T}(a, b)$  minimal in  $\mathbf{T}\{L_k\}$  must also be contained in a Gröbner basis of  $M$ .

## 5 Alternant Codes

Since the important work of Nechaev (1991) and Hammons et al. (1994) the theory of codes over rings has been well-studied and there are now many papers describing the structure and properties of such codes. We give an incomplete list here: Byrne et al. (2007, 2008), Greferath et al. (2004), Greferath and Schmidt (2000), Greferath and O'Sullivan (2004), Kurakin et al. (1999), Norton and Sălăgean (2003), Udaya

and Bonnecaze (1999), Voloch and Walker (1999), Wood (1999), but see Greferath (2009).

There has been considerably less work done on decoding algorithms for families of codes over rings, and very little for distance functions other than the Hamming weight. This is a significant gap given the fact that the homogeneous weight has emerged as important in the theory of codes over rings.

Decoding algorithms that correct up to half the minimum Hamming distance for BCH and alternant codes over rings can be found in Byrne and Fitzpatrick (2000, 2001, 2002) and Interlando et al. (1997). For alternant codes as defined below, the Sudan-Guruswami list decoding algorithm can be applied. This has been looked at in detail in Armand (2005b), which extends work done in O’Keeffe and Fitzpatrick (2002).

We give a brief account of alternant codes over a finite commutative chain rings and outline algorithms for their decoding that use Gröbner bases.

Let  $T$  be a subring of  $R$ . Let  $\mathbf{H}$  be the matrix

$$\begin{bmatrix} \gamma_0 & \gamma_1 & \dots & \gamma_{N-1} \\ \gamma_0\alpha_0 & \gamma_1\alpha_1 & \dots & \gamma_{N-1}\alpha_{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_0\alpha_0^{r-1} & \gamma_1\alpha_1^{r-1} & \dots & \gamma_{N-1}\alpha_{N-1}^{r-1} \end{bmatrix}$$

where  $r \leq N \leq |k_R| - 1$ ,  $\gamma = (\gamma_0, \dots, \gamma_{N-1})$ ,  $\alpha = (\alpha_0, \dots, \alpha_{N-1}) \in (R^*)^N$ , and  $\alpha_i - \alpha_j$  is a unit for  $i \neq j$ . Let  $C$  be the  $T$ -submodule of  $R$  determined as the nullspace in  $T^N$  of the parity check matrix  $\mathbf{H}$ .  $C$  is called an alternant code. A standard determinant argument shows that the minimum Hamming distance of this code is greater than  $r$ , and hence  $C$  corrects up to  $t = \lfloor \frac{r}{2} \rfloor$  Hamming errors in any transmission.

For the case  $T = \mathbb{Z}_{p^n}$ ,  $\gamma = (1, 1, \dots, 1)$ , it has been shown that with certain restrictions on  $r$ , the minimum Lee distance of the code  $C$  is at least  $2r + 1$ . Hence up to  $r$  Lee errors may be corrected in any transmission (Byrne 2002, Theorem V.4).

In the next two sections we describe the use of Gröbner bases in correcting up to  $r$  Hamming or Lee errors. We emphasize that these algorithms are entirely deterministic, require no searching, and have complexity quadratic in the number of errors.

## 5.1 Unique Decoding $C$ for the Hamming Distance

Let  $\mathbf{v} = \mathbf{c} + \mathbf{e}$  be a received word, where  $\mathbf{c}$  is a codeword and the error vector  $\mathbf{e}$  has Hamming weight at most  $t$ . Let  $\mathbf{S} = \mathbf{H}\mathbf{v} = \mathbf{He}$  be the syndrome vector. Let  $\mathcal{J} \subseteq \{0, \dots, N-1\}$  be the set of indices of nonzero coefficients of  $\mathbf{e}$ . The first task of the decoder is to determine the set  $\mathcal{J}$  of error locations. We define the error polynomial  $e = \sum_{j \in \mathcal{J}} e_j x^j$  and syndrome polynomial  $S = \sum_{i=0}^{r-1} \sum_{j \in \mathcal{J}} e_j \gamma_j \alpha_j^i x^i$

in the usual way. The error locator polynomial is

$$\Sigma = \prod_{j \in \mathcal{J}} (1 - \alpha_j x)$$

and the error evaluator polynomial is

$$\Omega = \sum_{j \in \mathcal{J}} e_j \gamma_j \prod_{k \neq j, k \in \mathcal{J}} (1 - \alpha_k x).$$

These polynomials are related by the well known key equation,

$$\Sigma S \equiv \Omega \pmod{x^r}$$

and the decoding problem involves solving this congruence for  $\Sigma, \Omega$  satisfying  $\partial \Sigma \leq t, \partial \Omega \leq t - 1$  where  $t = \lfloor r/2 \rfloor$ .

**Theorem 6** Let  $t = \lfloor r/2 \rfloor$ , and let  $(a, b) \in M$  satisfy the following: for some integer  $k \in \{0, \dots, n - 1\}$

- (i)  $\partial b < \partial a = \partial(\mu a) \leq t$ ,
- (ii)  $\mathfrak{p}^k \subseteq \mathbb{I}(a, b) \triangleleft R[x]$ .

Then  $\mathbf{T}(a, b)$  is minimal in  $\mathbf{T}\{L_{n-k}\}$  with respect to the term order  $<_{-1}$ .

If  $(a, b)$  satisfies the conditions of Theorem 6 then the  $(<_{-1})$ -minimality of  $\mathbf{T}(a, b)$  in some  $\mathbf{T}\{L_{n-k}\}$  implies that  $(a, b)$  is contained up to equivalence in a Szekeres-like basis  $\mathbf{S}(M)$ .

We express the error vector  $\mathbf{e}$  as a sum of vectors with disjoint support. For each  $i \in \{0, \dots, n - 1\}$ ,  $j \in \{0, \dots, N - 1\}$ , write  $e_j = e_j^{(i)} \pi^i$  for some  $e_j^{(i)} \in R^* \cup \{0\}$  and let  $\mathbf{e}^{(i)}$  be the length  $N$  vector with  $j$ th component  $e_j^{(i)}$ . We now obtain the decomposition

$$\mathbf{e} = \mathbf{e}^{(0)} + \pi \mathbf{e}^{(1)} + \dots + \pi^{n-1} \mathbf{e}^{(n-1)}.$$

Associated with each error vector  $\mathbf{e}^{(i)}$  are the polynomials  $e^{(i)} = \sum_{j \in \mathcal{J}_i} e_j^{(i)} x^j$ ,

$$\Sigma^{(i)} = \prod_{j \in \mathcal{J}_i} (1 - \alpha_j x), \quad S^{(i)} = \sum_{k=0}^{r-1} \sum_{j \in \mathcal{J}_i} e_j^{(i)} \gamma_j \alpha_j^{k+1} x^k,$$

$$\Omega^{(i)} = \sum_{j \in \mathcal{J}_i} e_j^{(i)} \gamma_j \prod_{k \neq j, k \in \mathcal{J}_i} (1 - \alpha_k x).$$

The following result shows that  $\mathfrak{p}^{n-1} \subset \mathbb{I}(\Sigma, \Omega)$ , and hence  $\Sigma, \Omega$  satisfy the hypothesis of Theorem 6 with  $k = n - 1$ . Therefore,  $(\Sigma, \Omega)$  can be uniquely identified (up to equivalence) as the element of  $\mathbf{S}(M)$  with minimal leading term among those with leading coefficient 1.

**Theorem 7** Let  $\Sigma$  and  $\Omega$  be a pair of error locator and error evaluator polynomials for an error vector  $\mathbf{e}$ . With the same notation as the above,  $\mathbb{I}(\Sigma, \Omega)$  has a Szekeres-like basis of the form

$$\{\Psi^{(0)}, \pi \Sigma^{(2)} \dots \Sigma^{(n-1)}, \pi^2 \Sigma^{(3)} \dots \Sigma^{(n-1)}, \dots, \pi^{n-2} \Sigma^{(n-1)}, \pi^{n-1}\}.$$

**Corollary 4** The solution  $(\Sigma, \Omega)$  of the key equation is (up to equivalence) the element with minimal leading term among those with leading coefficient 1 in a Szekeres-like basis for the solution module  $M$ , under the term order  $<_{-1}$ .

Given an element  $(a, b) \in L_1$  with leading term minimal in  $\mathbf{T}\{L_1\}$ , we compute the roots of  $\Sigma$  as follows. Since  $\mu(a, b) = \mu(\Sigma, \Omega)$  for all  $(a, b)$  with leading term minimal in  $\mathbf{T}\{L_1\}$  then in particular

$$\mu a = \mu \Sigma = \prod_{j \in \mathcal{J}} (1 - \mu \alpha_j x).$$

The roots  $\alpha_j$  are then determined uniquely from the roots  $\mu(\alpha_j)$  and location vector  $\alpha = [\alpha_0, \dots, \alpha_{N-1}]$ , whose components comprise a set of distinct coset representatives for the cosets of  $M$  in  $R$ . Once the error locations have been identified, a modified Forney procedure (Forney 1965; Interlando and Palazzo 1995) may be applied to compute the error magnitudes.

## 5.2 Unique Decoding of $C$ for the Lee Distance

We describe a decoding algorithm for a subclass of alternant codes over  $T = \mathbb{Z}_{p^n}$  with respect to the Lee metric. This extends the results of Roth and Siegel (1994) (for codes over  $\mathbb{Z}_p$ ) taking an approach using Gröbner bases. A comparison of the performance of the codes over  $\mathbb{Z}_{p^n}$  and their counterparts is given in Byrne (2002, Sect. VI). The algorithm presented here again has complexity quadratic in the number of errors, and many computations are carried out over a finite field.

For any  $\theta \in \mathbb{Z}_{p^n}$ , we denote by  $|\theta|_L$  the Lee value of  $\theta$  which is the least integer magnitude of  $\theta$  modulo  $p^n$ . The Lee weight of a vector is then the sum of the Lee values of its components. Following Roth and Siegel (1994), given  $\mathbf{v} \in \mathbb{Z}_{p^n}^N$ , we define the vectors  $\mathbf{v}^+ = [v_1^+, \dots, v_N^+]$  and  $\mathbf{v}^- = [v_1^-, \dots, v_N^-]$  as follows

$$v_j^+ = \begin{cases} v_j & \text{if } v_j = |v_j|_L, \\ 0 & \text{otherwise,} \end{cases} \quad v_j^- = \begin{cases} p^n - v_j & \text{if } p^n - v_j = |v_j|_L, \\ 0 & \text{otherwise,} \end{cases}$$

which gives the decomposition  $\mathbf{v} = \mathbf{v}^+ - \mathbf{v}^-$ .

As usual, the error vector is  $\mathbf{e} = \mathbf{v} - \mathbf{c}$ . The positive and negative error vectors are given by  $\mathbf{e}^+$  and  $\mathbf{e}^-$ . The syndrome values are defined in the usual way as  $S_\ell = \sum_{j=0}^{N-1} e_j \alpha_j^\ell$  for  $0 \leq \ell < \infty$ . The positive and negative syndrome values are defined

by the sums  $S_\ell^+ = \sum_{j=0}^{N-1} e_j^+ \alpha_j^\ell$  and  $S_\ell^- = \sum_{j=0}^{N-1} e_j^- \alpha_j^\ell$ , respectively, for  $0 \leq \ell$  and the positive and negative error locator polynomials are given by  $\Sigma^+ = \prod_{j=0}^{N-1} (1 - \alpha_j x)^{e_j^+}$  and  $\Sigma^- = \prod_{j=0}^{N-1} (1 - \alpha_j x)^{e_j^-}$  where for ease of notation we identify the exponents  $e_j^\pm$  with their Lee values  $|e_j^\pm|$ . The *error locator ratio* is  $\rho = \Sigma^+ / \Sigma^- \in R[[x]]$ . Let  $\Phi$  be the unique polynomial of degree less than  $r$  such that  $\Phi \equiv \rho \pmod{x^r}$ . Then clearly

$$\Sigma^- \Phi \equiv \Sigma^+ \pmod{x^r} \quad (1)$$

giving a *key equation*, and

$$S_j + \sum_{i=1}^{j-1} \Phi_i S_{j-i} + j \Phi_j = 0 \quad (2)$$

for  $1 \leq j \leq r-1$ . The decoding problem amounts to solving (1) subject to certain conditions. Since the polynomials  $\mu \Sigma^+ = \prod_{j=0}^{N-1} (1 - \mu \alpha_j x)^{e_j^+}$  and  $\mu \Sigma^- = \prod_{j=0}^{N-1} (1 - \mu \alpha_j x)^{e_j^-}$  give the same error locations as the original positive and negative error locator polynomials  $\Sigma^+$  and  $\Sigma^-$ , we only need to solve (1), over the residue field  $\mathbb{Z}_p$ .

Computing the polynomial  $\mu \Phi$  is done by recursive computation of the coefficients determined by (2) modulo  $p^n, p^{n-1}, \dots, p$ , (Byrne 2002, Theorem V.3). We then consider the solution module  $M = \{(a, b) \in k_R[x]^2 : a\mu\Phi \equiv b \pmod{x^r}\}$ . Again, computing a basis of the solution module  $M$  (say, implementing any of the algorithms of Fitzpatrick 1995) returns the required error locator polynomials. Once these have been found, adopting a procedure such as a modified Chien search returns the error vector.

### 5.3 List Decoding of $C$ for the Hamming Distance

In Guruswami and Sudan (1999), improving results of Sudan (1997), the authors propose a list decoding algorithm that corrects up to  $N - \sqrt{N(N-\delta)}$  Hamming errors in an RS code of length  $N$  and designed distance  $\delta$ . In O’Keeffe and Fitzpatrick (2002), the authors present an FGLM-like algorithm that may be applied to a range of problems in systems theory and coding, including list decoding. The authors also observe that both Sudan (1997) and Guruswami and Sudan (1999) may be viewed as instances of the methods used in Kötter and Vardy (2003), and demonstrate that these methods translate to the Gröbner basis setting in a very natural way. In Armand (2005b) it is explicitly shown that Algorithm 6.3 of O’Keeffe and Fitzpatrick (2002) for the list decoding of generalized RS codes is valid over commutative rings. We give a brief outline of that work here.

We assume now that  $T = R$  is a finite unital commutative ring for the code  $C$  and weaken the constraints on the  $\alpha_i$  as follows: for each  $i \neq j$ , neither  $\alpha_i$  nor

```

Poly-Reconstruct([GS99, Arm05b])
INPUT:  $N, k = N - r, \ell, \ell^2 > kN$ ,  $(\alpha_0, y_0), \dots, (\alpha_{N-1}, y_{N-1})$ 
OUTPUT: a list  $L$  of  $f \in R[x]$  such that  $d_H((f(\alpha_0), \dots, f(\alpha_{N-1})), (y_0, \dots, y_{N-1})) \leq N - \ell$ 

INITIALISE:  $v = 1 + \left\lfloor \frac{kN + \sqrt{k^2 N^2 + 4(\ell^2 - kN)}}{2(\ell^2 - kN)} \right\rfloor$ 
1. Compute  $Q(x, y) \in R[x, y]$ ,  $Q(x, y) \neq 0$  such that
 $w(\mathbf{T}(Q(x, y))) < v\ell$ 
 $Q(x + \alpha_i, y + i) = \sum_{ab} q_{iab} x^a y^b$  satisfies  $q_{iab} = 0, a + b < v, i = 0, \dots, N - 1$ .
2. Compute the set  $P$  of all polynomials  $f(x) \in R[x]$  such that
 $Q(x, f(x)) = 0$ 
 $L := \{f(x) \in P \mid d_H((f(\alpha_0), \dots, f(\alpha_{N-1})), (y_0, \dots, y_{N-1})) \leq N - \ell\}$ 

```

**Fig. 3** Poly-Reconstruct algorithm

$\alpha_i - \alpha_j$  is a zero divisor in  $R$  (some authors describe such a set of elements of  $R$  as *subtractive*). The code  $C$  can be viewed as an algebraic geometry code, having words of the form

$$C = \{(\gamma'_0 f(\alpha_0), \dots, \gamma'_{N-1} f(\alpha_{N-1})) \mid f \in R[x], \partial f < N - r + 1\},$$

where the  $\gamma'_i$ 's satisfy the equations  $\sum_{i=0}^{N-1} \gamma_i \gamma'_i \alpha_i^j$ , for each  $j \in \{0, 1, \dots, N - 2\}$ . In fact, the  $\gamma'_i$ 's play the role of the  $\gamma_i$ 's in  $C^\perp$ .

Given a received word  $(y_0, \dots, y_{N-1})$ , a list decoder seeks to return a list of all polynomials  $f \in R[x]$  of degree at most  $N - r$  satisfying  $f(\alpha_i) = y_i$  for some  $\ell$  indices  $i$ , where  $N - \ell$  is the chosen list decoding radius. The Poly-Reconstruct algorithm of Guruswami and Sudan (1999, Sect. II B) does this in two stages. First it computes a polynomial  $Q(x, y)$  satisfying certain weighted degree constraints such that  $(\alpha_i, y_i)$  is a singularity of  $Q(x, y)$  of multiplicity at least  $v$ , where  $v$  is an integer determined by the parameters of  $C$  and  $\ell$  (see Fig. 3) in order that  $V := \#\{x^a y^b : a + b < v\}$  satisfies

$$NV < \#\{x^a y^b : w(x^a y^b) := a + kb \leq v\ell\}.$$

In other words, the constraints on  $Q(x, y)$  are set such that the corresponding homogeneous system is solvable over  $R$ , i.e. such that the McCoy rank of the system is less than the number of unknowns.

In the second part of the algorithm, Poly-Reconstruct finds all polynomials  $f$  of degree at most  $N - r$  such that  $y - f(x)$  is a factor of  $Q(x, y)$ . The soundness of this algorithm over  $R$  for the code  $C$  as defined follows since the differences  $\alpha_i - \alpha_j$  are non zero-divisors for distinct pairs  $i, j$  (compare Armand 2005b, Lemmas 2, 3, 4, 5 with Guruswami and Sudan 1999, Lemmas 4, 5, 6, 7). If  $Q(x, f(x)) = 0$  then  $y - f(x)$  divides  $Q(x, y)$  in  $R[x, y]$ , so appropriate polynomials  $f$  can be found by finding factors of  $Q(x, y)$ . The algorithm is outlined in Fig. 3.

We now describe an FGLM-like solution (O'Keeffe and Fitzpatrick 2002; Armand 2005b) of Part 1 of the algorithm. The expansion of  $Q(x, y) \in R[x, y]$

about  $\alpha_i, y_i$  gives

$$Q(x + \alpha_i, y + y_i) = \sum_{ab} q_{iab} x^a y^b$$

for some coefficients  $q_{iab}$ . For the points  $(\alpha_i, y_i)$  with multiplicities  $m_i$ , the polynomial sought is minimal with respect to a given term order and satisfies  $q_{iab} = 0$  for  $0 \leq i \leq N - 1$  and  $a + b < m_i$ .

For a Gröbner basis perspective, define on the terms of  $R[x, y]$  the weight function  $w(x^i y^j) := i + kj$  and consider the term-ordering  $<_{(1,k)}$  which is the refinement of  $w$  with the lex-ordering indexed by  $y < x$ , so

- (i)  $w(x^i y^j) = i + kj < s + kt = w(x^s y^t) \implies x^i y^j <_{(1,k)} x^s y^t;$
- (ii)  $w(x^i y^j) = i + kj = s + kt = w(x^s y^t) \implies (x^i y^j <_{(1,k)} x^s y^t \iff j > t).$

The required polynomial  $Q(x, y)$  can be found by computing a Gröbner basis for the ideal  $\hat{\mathcal{M}}$  of all solutions to the system of congruences

$$\hat{Q}(x + \alpha_i, y + y_i) \equiv 0 \pmod{\langle x^a y^b : a + b = m_i \rangle},$$

for  $i \in 0, \dots, N - 1$  and exponents  $m_i$ . More precisely, for any given term  $\delta$ , setting  $q := \min\{b \in \mathbb{N} : \delta <_{(1,k)} y^b\}$  we identify the set

$$R[x, y]_\delta = \{P(x, y) \in R[x, y] : \mathbf{T}(P) <_{(1,k)} \delta\}$$

with the  $R[x]$ -module  $R[x]^q$  by using the map

$$\phi : R[x, y]_\delta \longrightarrow R[x]^q : \phi\left(\sum_{b=0}^{q-1} h_b(X) Y^b\right) \mapsto (h_0, \dots, h_{q-1}) = \sum_{b=0}^{q-1} h_b \mathbf{e}_{b+1}.$$

We define a module  $\mathcal{M} \subset R[x]^q$  by

$$\{h \in R[x]^q : h = \phi(\hat{Q}(x, y)), \mathbf{T}(\hat{Q}) <_{(1,k)} \delta, \hat{Q}(x, y) \in \hat{\mathcal{M}}\}.$$

In the decoding application,  $Q(x, y)$  corresponds to the minimal element of  $\mathcal{M}$  with respect to the term order in  $R[x]^q$  induced by the order  $<_{(1,k)}$  in  $R[x, y]$ , namely

$$x^i \mathbf{e}_{j+1} <_{(1,k)} x^s \mathbf{e}_{t+1} \iff x^i y^j <_{(1,k)} x^s y^t.$$

The details of an algorithm to compute a Gröbner basis of  $\mathcal{M}$  from a standard basis can be read in O'Keeffe and Fitzpatrick (2002, Algorithm 6.3), and the general procedure has been extended for the ring case (Armand 2005b, Algorithm 2).

## References

- J. Apel, *Computational ideal theory in finitely generated extension rings*, Theoret. Comput. Sci. **244** (2000), nos. 1–2, 1–33.

- M. A. Armand, *Improved list decoding of generalized Reed–Solomon and alternant codes over Galois rings*, IEEE Trans. on Inf. Th. **51** (2005a), no. 2, 728–733.
- M. A. Armand, *List decoding of generalized Reed–Solomon codes over commutative rings*, IEEE Trans. on Inf. Th. **51** (2005b), no. 1, 411–419.
- A. Assi, *Effective constructions in commutative algebra*, Ph.D. thesis, Grenoble, 1991.
- T. Becker and V. Weispfenning, *Gröbner bases*, Graduate texts in mathematics, vol. **141**, Springer, Berlin, 1993. A computational approach to commutative algebra, In cooperation with Heinz Kredel.
- O. Billet and J. Ding, *Overview of cryptanalysis techniques in multivariate public key cryptography*, this volume, 2009, pp. 263–283.
- M. Brickenstein, *Gröbner bases with slim polynomials*, Reports in Comp. Alg. 35, Univ. Kaiserslautern, Kaiserslautern, 2005, <http://www.mathematik.uni-kl.de/>.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- E. Byrne, *Lifting decoding schemes over a Galois ring*, LNCS, vol. **2227**, Springer, Berlin, 2001, pp. 323–332.
- E. Byrne, *Decoding a class of Lee metric codes over a Galois ring*, IEEE Trans. on Inf. Th. **48** (2002), no. 4, 966–975.
- E. Byrne and P. Fitzpatrick, *Gröbner bases and alternant codes over Galois rings*, Proc. of ISIT 2000, 2000.
- E. Byrne and P. Fitzpatrick, *Gröbner bases over Galois rings with an application to decoding alternant codes*, J. Symb. Comput. **31** (2001), no. 5, 565–584.
- E. Byrne and P. Fitzpatrick, *Hamming metric decoding of alternant codes over Galois rings*, IEEE Trans. on Inf. Th. **48** (2002), no. 3, 683–694.
- E. Byrne, M. Greferath, and M. E. O’Sullivan, *The linear programming bound for codes over finite Frobenius rings*, Des. Codes Cryptogr. **42** (2007), no. 3, 289–301.
- E. Byrne, M. Greferath, and T. Honold, *Ring geometries, two-weight codes, and strongly regular graphs*, Des. Codes Cryptogr. **48** (2008), no. 1, 1–16.
- D. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms*, Springer, Berlin, 1992, An introduction to computational algebraic geometry and commutative algebra.
- P. Elias, *List decoding for noisy channels*, Tech. Rep. 335, MIT, Cambridge, 1957.
- J. C. Faugère, *A new efficient algorithm for computing Gröbner bases ( $F_4$ )*, J. Pure Appl. Algebra **139** (1999), nos. 1–3, 61–88.
- J. C. Faugère, *A new efficient algorithm for computing Gröbner bases without reduction to zero ( $F_5$ )*, Proc. of ISSAC 2002, ACM, New York, 2002, pp. 75–83.
- J. C. Faugère and A. Joux, *Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases*, LNCS, vol. **2729**, Springer, Berlin, 2003, pp. 44–60.
- J. C. Faugère, P. Gianni, D. Lazard, and T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, J. Symb. Comput. **16** (1993), no. 4, 329–344.
- P. Fitzpatrick, *On the key equation*, IEEE Trans. on Inf. Th. **41** (1995), no. 5, 1290–1302.
- P. Fitzpatrick, *On the scalar rational interpolation problem*, Math. Control Sign. Sys. **9** (1996), no. 4, 352–369.
- P. Fitzpatrick, *Solving a multivariable congruence by change of term order*, J. Symb. Comput. **11** (1997), 505–510.
- P. Fitzpatrick and S. M. Jennings, *Comparison of two algorithms for decoding alternant codes*, AAECC **9** (1998), no. 3, 211–220.

- G. D. Forney, *On decoding BCH codes*, IEEE Trans. on Inf. Th. **11** (1965), 549–557.
- R. Gebauer and H. M. Möller, *On an installation of Buchberger's algorithm*, J. Symb. Comput. **6** (1988), nos. 2–3, 275–286.
- R. Gilmer, *Multiplicative ideal theory*, Pure and applied mathematics, vol. **12**, Dekker, New York, 1972.
- A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso, “One Sugar cube, please” or selection strategies in the Buchberger algorithm, Proceedings of ISSAC 1991, ACM, New York, 1991, pp. 49–54.
- M. Greferath, *An introduction to ring-linear coding theory*, this volume, 2009, pp. 219–238.
- M. Greferath and M. E. O’Sullivan, *On bounds for codes over Frobenius rings under homogeneous weights*, Discrete Math. **289** (2004), nos. 1–3, 11–24.
- M. Greferath and S. E. Schmidt, *Finite-ring combinatorics and MacWilliams’ equivalence theorem*, J. Combin. Theory Ser. A **92** (2000), no. 1, 17–28.
- M. Greferath, A. Nechaev, and R. Wissbauer, *Finite quasi-Frobenius modules and linear codes*, J. Algebra Appl. **3** (2004), no. 3, 247–272.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick’s approach to recent developments*, this volume, 2009, pp. 197–218.
- V. Guruswami and M. Sudan, *Improved decoding of Reed–Solomon and algebraic geometric codes*, IEEE Trans. on Inf. Th. **45** (1999), no. 6, 1757–1767.
- A. R. Hammons Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, *The  $\mathbf{Z}_4$ -linearity of Kerdock, Preparata, Goethals, and related codes*, IEEE Trans. on Inf. Th. **40** (1994), no. 2, 301–319.
- J. C. Interlando and R. J. Palazzo, *Multisequence generation and decoding of cyclic codes over  $\mathbf{Z}_q$* , Proc. of ISIT 1995, 1995, pp. 1–6.
- J. C. Interlando, R. Palazzo Jr., and M. Elia, *On the decoding of Reed–Solomon and BCH codes over integer residue rings*, IEEE Trans. on Inf. Th. **43** (1997), no. 3, 1013–1021.
- A. Kandri-Rody and D. Kapur, *Computing a Gröbner basis of a polynomial ideal over a Euclidean domain*, J. Symb. Comput. **6** (1988), no. 1, 37–57.
- R. Köetter and A. Vardy, *Algebraic soft-decision decoding of Reed–Solomon codes*, Trans. on Inf. Th. **49** (2003), no. 11, 2809–2825.
- V. Kurakin, A. Kuzmin, V. Markov, A. Mikhalev, and A. Nechaev, *Linear codes and polylinear recurrences over finite rings and modules (a survey)*, LNCS, vol. **1719**, Springer, Berlin, 1999, pp. 365–391.
- M. Lauer, *Canonical representative for residue classes of a polynomial ideal*, Proc. of ACM SSAC1976, 1976, pp. 339–345.
- B. R. McDonald, *Finite rings with identity*, Dekker, New York, 1974.
- H. M. Möller, *On the construction of Gröbner bases using syzygies*, J. Symb. Comput. **6** (1988), nos. 2–3, 345–359.
- T. Mora, *Solving polynomial equation systems. II, Macaulay’s paradigm and Gröbner technology*, Encyclopedia of mathematics and its applications, vol. **99**, Cambridge University Press, Cambridge, 2005.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, this volume, 2009, pp. 69–91.
- T. Mora, E. Orsini, and M. Sala, *General error locator polynomials for binary cyclic codes with  $t \leq 2$  and  $n < 63$* , BCRI preprint, [www.bcri.ucc.ie](http://www.bcri.ucc.ie) 43, UCC, Cork, Ireland, 2006.
- A. A. Nechaev, *Kerdock codes in cyclic form*, Discrete Math. Appl. **1** (1991), no. 4, 365–384.
- G. Norton, *On minimal realization over a finite chain ring*, Des. Codes Cryptogr. **16** (1999), no. 2, 161–178.
- G. H. Norton and A. Sălăgean, *On the key equation over a commutative ring*, Des. Codes Cryptogr. **20** (2000), no. 2, 125–141.
- G. H. Norton and A. Sălăgean, *Cyclic codes and minimal strong Gröbner bases over a principal ideal ring*, Finite Fields Appl. **9** (2003), no. 2, 237–249.
- H. O’Keeffe and P. Fitzpatrick, *Gröbner bases solutions of constrained interpolation problems*, Linear Algebra and Its Applications **351–352** (2002), 533–551.

- E. Orsini and M. Sala, *Correcting errors and erasures via the syndrome variety*, J. Pure Appl. Algebra **200** (2005), 191–226.
- E. Orsini and M. Sala, *General error locator polynomials for binary cyclic codes with  $t \leq 2$  and  $n < 63$* , IEEE Trans. on Inf. Th. **53** (2007), 1095–1107.
- L. Pan, *On the D-bases of polynomial ideals over principal ideal domains*, J. Symb. Comput. **7** (1989), no. 1, 55–69.
- F. L. Pritchard, *The ideal membership problem in non-commutative polynomial rings*, J. Symb. Comput. **22** (1996), no. 1, 27–48.
- N. Ratnakar and R. Köetter, *Exponential error bounds for algebraic soft-decision decoding of Reed–Solomon codes*, IEEE Trans. on Inf. Th. **51** (2005), no. 11, 3899–3917.
- R. M. Roth and G. Ruckenstein, *Efficient decoding of Reed–Solomon codes beyond half the minimum distance*, IEEE Trans. on Inf. Th. **46** (2000), no. 1, 246–257.
- R. M. Roth and P. H. Siegel, *Lee-metric BCH codes and their application to constrained and partial-response channels*, IEEE Trans. on Inf. Th. **40** (1994), 1083–1096.
- D. A. Spear, *A constructive approach to commutative ring theory*, Proc. of the 1977 MACSYMA Users’ Conference, NASA CP-2012, 1977, pp. 369–376.
- M. Sudan, *Decoding of Reed–Solomon codes beyond the error correction bound*, J. of Complexity **13** (1997), 180–193.
- G. Szekeres, *A canonical basis for the ideals of a polynomial domain*, Amer. Math. Monthly **59** (1952), 379–386.
- C. Traverso and L. Donato, *Experimenting the Gröbner basis algorithm with AlPI system*, Proc. of ISSAC 1989, ACM, New York, 1989, pp. 192–198.
- P. Udaya and A. Bonnecaze, *Decoding of cyclic codes over  $F_2 + uF_2$* , IEEE Trans. on Inf. Th. **45** (1999), no. 6, 2148–2157.
- J. F. Voloch and J. L. Walker, *Codes over rings from curves of higher genus*, IEEE Trans. on Inf. Th. **45** (1999), no. 6, 1768–1776.
- J. A. Wood, *Duality for modules over finite rings and applications to coding theory*, Amer. J. Math. **121** (1999), no. 3, 555–575.
- G. Zacharias, *Generalized Gröbner bases in commutative polynomial rings*, Ph.D. thesis, MIT, 1978.
- O. Zariski and P. Samuel, *Commutative algebra*, vol. I, Van Nostrand, Princeton, 1958.

# Overview of Cryptanalysis Techniques in Multivariate Public Key Cryptography

Olivier Billet and Jintai Ding

**Abstract** This paper summarizes most of the main developments in the cryptanalysis of multivariate cryptosystems and discuss some problems that remain open. A strong emphasis is put on the symbolic computation tools that have been used to achieve these advances.

## 1 Introduction

The most widely deployed public key cryptosystem nowadays is without any doubt the RSA cryptosystem. Its security is somewhat related to the fact that no reasonably fast algorithm for the factorization of large integers is known up to now. Due to fast developments in the field of integer factorization, a secure public key cryptosystem relying on the assumption that factoring integers is a hard problem must use integers  $N = pq$  where  $p$  and  $q$  are prime numbers of size at least 1024 bits and preferably 2048 bits. This implies heavy computations during the encryption process, which makes it inefficient and costly. Moreover, a new threat has recently appeared that would break the RSA cryptosystem: quantum computers. Under the assumption that quantum computers can be built, Shor (1997) discovered an algorithm that could factor an integer in polynomial time in terms of its size in bits, thus rendering the RSA cryptosystem useless. Shor's algorithm can also break essentially all number theoretic based public key cryptosystem as well as the elliptic curve cryptosystems or the Diffie–Hellman key exchange. There have been great efforts dedicated to the construction of quantum computers and although nobody has built such computers able to attack the RSA or the discrete logarithm based cryptosystems, definitely there is a need for other efficient and secure cryptosystems.

There are currently a few families of cryptosystems that could potentially resist future quantum computers: these are the cryptosystems based on error-correcting codes (McEliece 1978; Niederreiter 1986), the public key cryptosystems based on

---

O. Billet

Orange Labs, 38–40 rue du Général Leclerc, 92794 Issy-les-Moulineaux, France  
e-mail: [olivier.billet@orange-ftgroup.com](mailto:olivier.billet@orange-ftgroup.com)

J. Ding

Department of Mathematical Sciences, Department of Computer Sciences,  
University of Cincinnati, Cincinnati, OH 45220, USA  
e-mail: [ding@math.uc.edu](mailto:ding@math.uc.edu)

lattices (Regev 2006; Nguyen and Stern 2001), and the multivariate public key cryptosystems.

The class of multivariate cryptosystems is a special class of schemes whose security is related to the hardness of solving sets of multivariate equations. The obvious way of solving them is to compute a Gröbner basis (Buchberger 1965, 1970, 1985, 1998, 2006). Solving sets of multivariate equations is a well known hard problem that is not only hard on the average but already for sets of equations that are practical to evaluate, like for instance a hundred of randomly chosen quadratic equations in a hundred of unknowns defined over the binary field (Bardet 2004; Fraenkel and Yesha 1980). For obvious efficiency reasons, the multivariate polynomials that constitute the system are generally chosen to be quadratic polynomials defined over a small finite field—that is ranging from  $\mathbb{F}_2$  to  $\mathbb{F}_{2^8}$ —though there exist rare exceptions (Billet and Gilbert 2003; Wang et al. 2006).

In the particular case of symmetric cryptographic primitives, it is often possible to randomly draw the multivariate polynomials with carefully selected parameters in order to obtain a security reduction to a generic instance of the underlying NP-hard problem: this is for instance the case for the stream-cipher QUAD proposed in Berbain et al. (2006), Berbain and Gilbert (2007), but also for the hash function MQ-HASH proposed in Billet et al. (2007). However in the case of asymmetric multivariate schemes, the designer has to embed a trapdoor in order to enable the owner of the secret key to solve the system of equation derived from the public key and the cipher text or the message to be signed. The side effect of embedding such a trapdoor in the public set of polynomials is that there is usually no reduction to a generic instance of the underlying hard problem anymore, since the corresponding systems are not randomly chosen. The security of the scheme has to be assessed by other means, usually by conducting experiments with the best system solvers or by mounting a specially crafted algebraic attack that exploits the underlying algebraic structure.

The current proposals for multivariate asymmetric cryptosystems might be classified into three main categories, some of which combine features from several categories: Matsumoto-Imai like schemes, Oil and Vinegar like schemes, stepwise triangular schemes, and an additional fourth category called *Polly Cracker* schemes. In this survey however, we focus on the first three categories; the reader can find more information on *Polly Cracker* schemes in Fellows and Koblitz (1994), Caboara et al. (2008) and especially in Levy-dit-Vehel et al. (2009). All of the schemes from the first three categories rely on the hardness of system solving, but some of them additionally rely on other hard problems such as finding rational mappings between polynomial maps or finding a linear combination of small rank of a given set of matrices.

## 2 Inversion Attacks

Although there exist several multivariate authentication schemes, we hereafter focus on multivariate asymmetric encryption schemes and multivariate signature schemes.

We tried to unify the notations as much as possible with the problem of system solving in mind: We denote the base field by  $\mathbb{K}$ , and use  $x$  and  $y$  to respectively denote the input and the output of a public key function. The input of the public key being an element of the vector space  $\mathbb{K}^n$ , we sometimes make use of the standard underlying coordinate system and write  $x = (x_1, \dots, x_n)$ . Finally, we denote a multivariate public key by a polynomial mapping from the vector space  $\mathbb{K}^n$  to the vector space  $\mathbb{K}^m$ :

$$\begin{aligned} f : \mathbb{K}^n &\longrightarrow \mathbb{K}^m, \\ x = (x_1, \dots, x_n) &\longmapsto y = (p_1(x), \dots, p_m(x)), \end{aligned} \tag{1}$$

where  $p_1, \dots, p_m$  are multivariate polynomials defined over  $\mathbb{K}[x_1, \dots, x_n]$ . In the case of encryption schemes,  $x$  and  $y$  respectively denote the plain text and the cipher text. In the case of signature schemes,  $x$  and  $y$  respectively denote the signature and the hashed value to be signed.

This part describes several attacks against the underlying system solving problem of several multivariate cryptosystems, that is, it reports successful methods to invert the public key of some asymmetric cryptosystems. We first review the linearization attack of Patarin (1995) against Matsumoto–Imai scheme A, and then describe different attacks against a generalisation of it named Hidden Field Equations (HFE), that was proposed in Patarin (1996).

## 2.1 Matsumoto–Imai Scheme A and Its Variations

Starting from 1983, Matsumoto and Imai proposed a series of public key cryptosystems relying on the hardness of system solving. In Imai and Matsumoto (1985), they proposed a scheme “based on obscure representation of polynomials” often called  $C^*$  and hereafter called Matsumoto–Imai scheme A. This scheme uses exponentiation over an extension  $\mathbb{E}$  of degree  $n$  of a base finite field  $\mathbb{K}$  of size  $q$ . (We denote by  $\varphi$  the canonical embedding of  $\mathbb{K}^n$  into  $\mathbb{E}$  and  $\mathbf{x} = \varphi(x)$ .) The exponent is chosen of the form  $1 + q^\theta$  and prime to  $q^n - 1$  so as to allow efficient inversion. This exponentiation is then concealed by two change of variables  $S$  and  $T$  of  $\mathbb{K}^n$ . The public key is therefore given by the  $n$ -tuple  $(p_1, \dots, p_n)$  of polynomials in  $n$  unknowns  $x_1, \dots, x_n$  defined over  $\mathbb{K}$  via:

$$\begin{aligned} \mathbb{K}^n &\longrightarrow \mathbb{K}^n, \\ x = (x_1, \dots, x_n) &\longmapsto (p_1(x), \dots, p_n(x)) = T \circ \varphi^{-1}((\varphi \circ S(x))^{1+q^\theta}). \end{aligned} \tag{2}$$

One key fact allowing an efficient representation of the public key as the  $n$ -tuple of polynomials  $(p_1, \dots, p_n)$  is that the mapping  $\mathbf{x} \mapsto \mathbf{x}^q$  (which is often also called the Frobenius endomorphism) is a  $\mathbb{K}$ -linear mapping and thus elevating to the power of  $1 + q^\theta$  is  $\mathbb{K}$ -quadratic. Another mandatory property is the ability for the owner

of the secret key to efficiently compute a solution to the system:

$$\begin{cases} p_1(x_1, \dots, x_n) = y_1, \\ \vdots \\ p_n(x_1, \dots, x_n) = y_n, \end{cases} \quad (3)$$

for every  $n$ -tuple  $y = (y_1, \dots, y_n)$ , which should ideally correspond to the ability of performing decryption or signature. In order to solve (3), the secret key owner uses his knowledge of the secret linear mappings  $S$  and  $T$  and of an exponent  $e$  such that  $e(1 + q^\theta) \equiv 1 \pmod{q^n - 1}$  to invert each component of the public map in turn, which amounts to the following computation:  $x = S^{-1} \circ \varphi^{-1}((\varphi \circ T^{-1}(y))^e)$ . The name “obscure representation” comes from the assumption that the input and output coordinate systems are unknown to anyone but the secret key owner. Hence, the security of the cryptosystem not only relies on the hardness of solving (3), but also on the hardness of recovering any pair of mappings  $S_0$  and  $T_0$  such that:  $\forall x \in \mathbb{K}^n$ ,  $T_0 \circ \varphi^{-1}((\varphi \circ S_0(x))^{1+q^\theta}) = T \circ \varphi^{-1}((\varphi \circ S(x))^{1+q^\theta})$ . A more general version of this problem of crucial importance to the security of multivariate public schemes is discussed later on in this paper.

This construction can obviously be extended to accommodate several other internal transformations instead of the original exponentiation. However, there must be an efficient way to invert this internal transformation, and the public key should have an efficient representation. With these constraints in mind, Patarin (1996) proposed to use an internal transformation of type:

$$f : \mathbb{E} \longrightarrow \mathbb{E}, \quad \mathbf{x} \longmapsto \sum_{\substack{1 \leq i \leq j \leq n \\ q^i + q^j \leq D}} a_{i,j} \mathbf{x}^{q^i + q^j} + \sum_{\substack{1 \leq k \leq n \\ q^k \leq D}} b_k \mathbf{x}^{q^k} + c. \quad (4)$$

This internal transformation  $f$  of  $\mathbb{E}$  has the special property that its overall degree is bounded by some reasonable constant  $D$ : this trick enables the owner of the secret key to solve the equation  $f(\mathbf{x}) = \mathbf{y}$  in the unknown  $\mathbf{x}$  for any value  $\mathbf{y}$  of  $\mathbb{E}$ , since there exist algorithms polynomial in  $D$  and  $n$  for this task (von zur Gathen and Shoup 1992; Knuth 1997). The resulting cryptosystem is called Hidden Field Equations (HFE).

Another generalization of the Matsumoto–Imai scheme A is the use of a projection instead of a bijection for the change of coordinates  $T$ , and the public key then becomes a mapping from  $\mathbb{K}^n$  to  $\mathbb{K}^m$  with  $m < n$ . This can be seen as a modification of the original scheme where some of the polynomials in the public key have been removed. Patarin et al. (1998a, 2000) applied this idea to the Matsumoto–Imai scheme A to create the SFLASH signature scheme, and proposed a very similar variation around the HFE cryptosystem (Patarin 1996). We however note that this construction is mainly of interest in the setting of signature schemes since the public key mapping is not a bijection anymore.

Finally, we note that the secret changes of coordinate system can be taken as linear mappings or affine mappings. However, as shown by Geiselmann et al. (2001),

the constant parts of the secret affine mappings can often be deduced by an attacker (i.e. with the knowledge of the public key alone) and sometimes even leaks some information about the secret mappings themselves.

## 2.2 Direct Inversion Attacks

The essence of public key encryption (resp. signature) schemes is to give public access to a mechanism allowing the computation of a cipher text  $y$  from a plain text  $x$  (resp. verifying a signature  $x$  from a hashed value  $y$ ). In the special case of multivariate schemes, we have seen in the previous section that this mechanism is a polynomial mapping having a low degree in the input variables because of efficiency reasons. This mapping  $p$  constitutes the public key and an attacker can directly search for a value  $x$  verifying  $p(x) = y$  in order to decrypt  $y$  or to forge a signature  $x$ . Such attacks consist in solving a system of polynomial equations of low degree (quadratic in the case of Matsumoto–Imai and HFE), and there have been several algorithms designed to solve this task. The most famous are Buchberger's algorithm (Buchberger 1965, 1970, 1985, 1998, 2006; Mora 2009), Faugère's algorithms F4 and F5 (Faugère 1999, 2002), and basic algorithms such as the linearization tool XL suggested in Courtois et al. (2000) which is a particular case of F4 (Ars et al. 2004). The rationale behind the design of multivariate asymmetric cryptosystems is that the complexity of solving systems of randomly generated quadratic multivariate equations defined over a finite field is exponential in the number of unknowns on the average. At the same time, the trapdoor introduced in the public key of asymmetric multivariate cryptosystems makes the resulting system of equations specific and sometimes distinguishable from randomly generated ones.

The set of equations derived from the public key of Matsumoto–Imai scheme A instances can be solved by computing Gröbner bases: Dobbertin reported to have successfully solved such systems with Gebauer and Möller's version of Buchberger's algorithm while working at the BSI.<sup>1</sup> However, the first public cryptanalysis of Matsumoto–Imai scheme A was published by Patarin (1995): it is very instructive in that it explains why solving the system of equation through the computation of a Gröbner basis is possible. The key remark is that there exist bilinear equations relating the input and the output of the system. Indeed, recall that the internal transformation maps any element  $\mathbf{x}$  of the extension field  $\mathbb{E}$  to  $\mathbf{y} = \mathbf{x}^{1+q^\theta}$ , so that  $\mathbf{y} \mathbf{x}^{q^2\theta} = \mathbf{x} \mathbf{y}^{q^\theta}$ . This last bilinear equation still holds between the input and output variables  $x$  and  $y$  since they are  $\mathbb{K}$ -linear transformations of  $\varphi^{-1}(\mathbf{x})$  and  $\varphi^{-1}(\mathbf{y})$  respectively, so that the following holds for some set of coefficients  $a_{i,j}$ ,  $b_i$ , and  $c_j$ :

$$\sum_{1 \leq i, j \leq n} a_{i,j} y_i x_j + \sum_{1 \leq i \leq n} b_i y_i + \sum_{1 \leq j \leq n} c_j x_j + d = 0. \quad (5)$$

---

<sup>1</sup>Bundesamt für Sicherheit in der Informationstechnik which is the German Federal Office for Information Security.

Recall that  $y_i = p_i(x)$ . A common way to represent the set  $I_\delta$  of polynomials of degree  $\delta$  that belongs to the ideal generated by  $(p_1, \dots, p_n)$  is to construct a matrix whose columns are indexed by the monomials of  $I_\delta$  and whose lines are obtained by multiplying each  $p_i$  with every possible monomial  $u$  such that  $\deg(up_i) = \delta$ . This matrix is called a Macaulay matrix of degree  $\delta$ , see Macaulay (1916). In the case of Matsumoto–Imai scheme A, we see that the Macaulay matrix of degree 3 already contains the polynomials from (5)—remember that the  $y_i$  are polynomials of degree 2 in the  $x_i$ —and this explains why a direct Gröbner basis computation is efficient against this cryptosystem. The attack described by Patarin reads as follows Patarin (1995), Koblitz (1999): although the bilinear equations (5) are *a priori* unknown to the attacker, they can be easily interpolated by generating matching plain text/cipher text pairs from the public key. After that, finding an  $x$  corresponding to a given  $y$  is easy: just replace  $y$  in the interpolated equations (5) and solve the resulting linear system in  $x$ .

A number of multivariate cryptosystems are actually susceptible to attacks relying on low degree relations between input variables and output variables. This is for instance the case with the weak proposal (Wang et al. 2006) where a cryptanalysis directly stems from the above remarks (Ding et al. 2007a). Variants of another proposal called Tame Transformation Method (TTM) and published in Moh (1999) were also shown to be susceptible in Ding and Schmidt (2003, 2004).

A much less obvious behavior is exhibited by the HFE cryptosystem described above. Here again, the attacker can take advantage of the specific structure of the internal transformation to invert the public key with Gröbner basis methods efficiently. A simple counting argument briefly sketched in Faugère and Joux (2003) shows that the biggest Macaulay matrix constructed during a Gröbner basis computation with F4 has a much lower degree than that of a randomly drawn system of the same size. To see why, first remember that the internal transformation of HFE defined over  $\mathbb{E}$  of characteristic 2 has a degree bounded by  $D$ . Let us denote the public key of HFE by  $g$ . Then consider a constant  $H$  such that  $D \leq H < 2^n$ , and the number of pairs of integers  $(d_i, k)$  for which  $d_i$  is a sum of at most  $w - 2$  powers of 2 such that  $\varphi(x)^{d_i}(\varphi \circ g(x))^{2^k}$  has its degree bounded by  $H$ . It can be shown that there exists a value of  $H$  such that the number of monomials appearing in the set of equations generated this way is lower than the number of equations. Since  $\mathbf{x} \mapsto \mathbf{x}^{2^k}$  is a  $\mathbb{K}$ -linear mapping, the number  $w$  exactly corresponds to the degree of the biggest Macaulay matrix constructed during the Gröbner basis computation. This degree is smaller than the one encountered in the Gröbner basis computation for a randomly chosen system of equivalent size. This theoretical explanation is supported by various experiments. Indeed, with his own optimized implementation of F5, Faugère solved the HFE challenge posted on Courtois' web page (Patarin 1998). This challenge is an HFE public key with 80 equations in 80 unknowns defined over  $\mathbb{F}_2$  corresponding to an internal transformation of total degree 96. It was first solved by Faugère in about 52 hours on an HP workstation with an alpha EV68 processor running at 1000 MHz and  $2^{33}$  bytes of memory, and later on by Steel with Magma in about 22 hours on a 750 MHz Sunfire v880 using about  $2^{34}$  bytes of memory. As suggested by the above explanation, the degree of the biggest Macaulay matrix

encountered was especially low and always bounded by  $w$ . And indeed, the data from Faugère and Joux (2003) obtained from several runs on various HFE parameters confirmed the fact that this value  $w$  is way too small for the cryptosystem to be secure:

$$\begin{aligned} 5 \leq D \leq 12 &\rightarrow w = 4, & 128 \leq D \leq 1280 &\rightarrow w = 6, \\ 16 \leq D \leq 96 &\rightarrow w = 5, & 1536 \leq D \leq 4096 &\rightarrow w = 7. \end{aligned}$$

An interesting fact is that these values are independent of the number  $n$  of unknowns, at least for  $n < 160$ , which corresponds to public keys of practical sizes.

Along with the first challenge that was originally broken by Faugère, a second challenge was proposed that is still not broken. It consists in an HFE public key with 36 variables defined over the finite field  $\mathbb{F}_{2^4}$  of which four quadratic polynomials have been removed.

One might wonder if it is not possible to escape Gröbner basis attacks by tweaking the internal transformation so that its degree is not bounded anymore. Obviously, since the internal transformation has to be invertible by the legitimate user, this means that something must be relaxed somewhere. There has been some proposals along those lines in Ding et al. (2007b), Wang et al. (2006), all of which have been broken (Fouque et al. 2008a; Ding et al. 2007a). While these proposals were very specific, one might consider a broadest class that encompass such schemes and that we call Intermediate Field Systems: it comprises the schemes that have as internal transformation a set of multivariate polynomials in a small number of variables and defined over an intermediate extension field  $\mathbb{L}$ . Such an internal transformation might be inverted through the computation of Gröbner bases. In Billet et al. (2008), this class of schemes has been analyzed from the point of view of Gröbner basis attacks and it has been shown that the security achieved is asymptotically the same as that of the HFE cryptosystem.

## 2.3 MinRank

We just reviewed attacks against Matsumoto–Imai like cryptosystems aiming at directly solving the system arising from the public key. These “direct inversion” attacks do not try to recover a hidden specific structure implied by the presence of a trapdoor, though they rely on the existence of low degree relations between the value of the polynomials and their input variables. We describe here another family of attacks that first recover the hidden structure so that the attacker is in a position similar to that of the secret key’s owner. More precisely, we focus on multivariate asymmetric cryptosystems whose public key consist of quadratic polynomials having rank peculiarities, like Fell and Diffie (1985), Shamir (1993), Moh (1999). The general structure of such cryptosystems is based on the family of triangular (or “de

Jonqui  re") mappings  $x \mapsto y = J(x)$  defined as

$$\begin{cases} y_1 = x_1, \\ y_2 = x_2 + p_2(x_1), \\ y_3 = x_3 + p_3(x_1, x_2), \\ \vdots \\ y_n = x_n + p_n(x_1, x_2, \dots, x_{n-1}), \end{cases} \quad (6)$$

where the  $p_i$  are polynomial mappings, and for efficiency reasons usually restricted to quadratic polynomial mappings. It can be easily checked that inverting such an application is easy since it amounts to incrementally solve linear equations in a single variable. The first cryptanalysis against such cryptosystems is given by Copper-smith et al. (1997) and uses the rank in order to break the bi-rational permutations scheme proposed by Shamir (1993).

Before describing the underlying rank problem, we recall basic properties of multivariate quadratic polynomials. The first fact is that every quadratic form  $p$  has a canonical form that can be computed in polynomial time, that is there exists a change of coordinates  $S : (x_1, \dots, x_n) \mapsto (z_1, \dots, z_m)$  which can be efficiently found so that  $m \leq n$  is minimal and there exists another quadratic form  $\tilde{p}$  such that for all  $x$ ,  $p(x) = \tilde{p}(S(x))$ . This minimal  $m$  is called the rank of  $p$ . The other fact is that a unique symmetric matrix of size  $n$  can be associated to any quadratic form in  $n$  unknowns the usual way: entry  $(i, j)$  of the matrix is half the coefficient of monomial  $x_i x_j$  in the quadratic form and the diagonal coefficients are the ones of the monomials  $x_i^2$ . There are some difficulties in the case of a field of characteristic two that can be resolved by defining both entry  $(i, j)$  and entry  $(j, i)$  as the coefficient of monomial  $x_i x_j$  in the quadratic form when  $i \neq j$  and by defining entry  $(i, i)$  to be zero. Then, the rank of the symmetric matrix is equal to the rank of the quadratic form.

Thus, in the process of cancelling the effect of the linear mixing of the polynomials in the triangular form (aimed at hiding this specific structure), or alternatively in the process of recovering an equivalent version of the secret change of coordinates, the following problem naturally arises:

**Definition 1** (Minimun Rank) Given a set  $\{\mathbf{A}_1, \dots, \mathbf{A}_m\}$  of  $n \times n$  matrices defined over a finite field  $\mathbb{K}$  and an integer  $r < n$ , find a non-trivial linear combination over  $\mathbb{K}$  of rank less than or equal to  $r$ .

The complexity of the general MinRank problem over various fields has been studied by Buss et al. (1999), where it has been shown to be NP-complete when  $r$  varies with  $n$ . However, for a fixed  $r$ , there are polynomial algorithms to solve this problem. Several of them are described in Goubin (2003). One of it was devised and used in Goubin and Courtois (2000) by Goubin and Courtois to break the TTM scheme proposed in Moh (1999), and was later on extended in Billet and Gilbert (2006) by Billet and Gilbert to take advantage of particular settings. The exhaustive search was also extended in a similar way in Yang and Chen (2005). Most of these

algorithms merely use linear system solving combined with some form of exhaustive search.

Another point of view has been given by the authors of Coppersmith et al. (1993): the solutions of a MinRank instance with a set of  $m$  matrices of size  $n \times n$  are also the solutions of the system encoding the fact that every sub-matrix of size  $(r + 1) \times (r + 1)$  of the sought linear combination has determinant zero. The overall complexity<sup>2</sup> of solving such a system of equations is  $O\left(\frac{1}{(r+1)!}m^{\omega(r+1)}\right)$  provided there are enough equations to apply the linearization technique. Hence, this strategy works well if the rank is small and enough linearly independent equations can be derived.

An attack against HFE suggested by Kipnis and Shamir (1999) uses matrices over the extension field  $\mathbb{E}$  of degree  $n$  over  $\mathbb{K}$  of size  $q$ , and can be reduced to solving a huge system of equations. We briefly describe now this attack aiming at recovering HFE's private key. First of all, notice that the equation relating the public key and the private key can be rewritten as:  $t^{-1} \circ g = f \circ s$ , where  $s$  and  $t$  are the secret one-to-one linear mappings defined over  $\mathbb{E}$ ,  $f$  is a  $\mathbb{K}$ -quadratic mapping defined over  $\mathbb{E}$ , and  $g$  is the public mapping resulting from their composition. Since any linear mapping can be written in the form of  $x \mapsto \sum_{1 \leq i \leq n} \alpha_i x^{q^i}$ , the homogeneous component of degree two of the public mapping can be described as:

$$g^{(2)}(x) = \sum_{1 \leq i, j \leq n} \gamma_{i,j} x^{q^i + q^j}.$$

Hence, a symmetric matrix  $G$  can be associated to  $g$  such that  ${}^t X G X = g^{(2)}(x)$  where  $X = (x^q, \dots, x^{q^n})$ . (Again, some care has to be taken in the case of characteristic 2.) If  $s(x) = \sum_{1 \leq i \leq n} s_i x^{q^i}$  and  $t^{-1}(x) = \sum_{1 \leq i \leq n} t_i x^{q^i}$ , then the authors of Kipnis and Shamir (1999) show that  $\tilde{G} = {}^t W F W$  where  $F$  is the symmetric matrix associated to  $f$  as described above,  $W$  is defined by  $W_{i,j} = s_i^{q^j}$ ,  $G_{i,j}^{\circ k} = G_{i+k, j+k}$  with indices taken modulo  $n$ , and  $\tilde{G} = \sum_{1 \leq k \leq n} t_k G^{\circ k}$ . The authors of Kipnis and Shamir (1999) then tried to solve a huge system of equations derived from this property, the complexity of which remained unclear. However, the equation  $\tilde{G} = {}^t W F W$  can be re-interpreted from a rank point of view when remembering that  $F$  has rank  $r = \log_q D$ —because the degree of  $f$  has been bounded by  $D$  so as to allow efficient inversion of  $f$ . This remark was formulated by Courtois in Courtois (2001) who showed that the problem of recovering the right  $t_k$  basically amounts to solve a MinRank problem with  $r$  about  $\log_q D$  given the set of matrices  $G^{\circ k}$  that are directly derived from the public key and suggested to use the sub-matrices strategy to solve it, the complexity of which would be:

$$\frac{1}{(\alpha \log_q n)!} \exp[O(\omega \alpha (\log_q n)^2)], \quad (7)$$

---

<sup>2</sup>Where the constant  $\omega$  depends on the method for solving linear systems; for instance  $\omega$  is about 2.807 when using Strassen's algorithm.

if enough linearly independent equations can be derived, since  $D = O(n^\alpha)$  for some  $\alpha \geq 1$ . An interesting point is that the resulting complexity estimate is slightly better than the one given in Granboulan et al. (2006) and gives an even stronger result: HFE's secret key can be recovered in quasi-polynomial time. However, the authors of Jiang et al. (2007) expressed some doubts about the ability to solve the MinRank problem by these means: they indeed proved that the algebraic system constructed as explained above has a lot of solutions, which shows that the complexity estimate (7) is too optimistic.

Obviously, being able to solve systems of equations arising from MinRank problems more efficiently than via linearization attacks would advance the state of the art in the cryptanalysis of many multivariate cryptosystems, such as schemes from the TTM family (Moh 1999; Yang and Chen 2005), Rainbow (Ding and Schmidt 2005a), or even HFEv (Ding and Schmidt 2005b). A new approach has just been proposed for special MinRank instances (Faugère and Perret 2008a).

## 2.4 Unbalanced Oil and Vinegar

The Oil and Vinegar signature scheme has been designed by Patarin and was first exposed in Patarin (1997). This design with a radically different trapdoor might have been inspired to Patarin by the linearization attack against Matsumoto–Imai like cryptosystems. In Oil and Vinegar schemes indeed, the secret transformation is made of  $o$  multivariate quadratic polynomials whose homogeneous part of degree two have the following specific form:

$$\sum_{\substack{1 \leq i \leq o \\ 1 \leq j \leq v}} a_{i,j} x_i y_j + \sum_{1 \leq i, j \leq v} b_{i,j} y_i y_j. \quad (8)$$

That is, two sets of variables  $O = \{x_i\}_{1 \leq i \leq o}$  and  $V = \{y_j\}_{1 \leq j \leq v}$  are used, but only monomials from  $\{zy\}_{(z,y) \in (O \cup V) \times V}$  are allowed to appear in the polynomials. It is easy to find a pre-image for a tuple of  $o$  such polynomials: after random values have been assigned to the variables from  $V$ , only a linear system in the variables from  $O$  remains. Finding a pre-image is then reduced to solving a linear system in these  $o$  variables. (Assuming these systems are uniformly distributed in the set of randomly drawn systems and there are  $o$  polynomials defined over a finite field of size  $q$ , the probability that such a system is invertible is given by  $(1 - \frac{1}{q}) \cdots (1 - \frac{1}{q^o})$ .) Hence, after a few trials with others random choices for the variables from  $V$ , a pre-image of the original system will be found.) This is why variables from  $O$  and  $V$  are respectively called oil and vinegar variables: assigning values to vinegar variables makes oil variables appear. As usual, this specific structure of the secret polynomials is hidden by a change of coordinates.

The balanced version of this Oil and Vinegar scheme, that is with  $o = v$ , was broken by Kipnis and Shamir (1998). The security of the unbalanced case as exposed by Kipnis et al. (1999) is still not well understood, although it is definitely

not secure when the number of vinegar variables is much bigger than the number of oil variables. A system of  $m$  randomly chosen multivariate quadratic equations in  $n$  unknowns can be easily solved when  $n \geq m^2$ , see Kipnis et al. (1999). Kipnis and Shamir (1998) also show that the attack against the balanced case, which is heavily relying on the fact that  $o = v$ , can actually be used in the case where  $v$  is only slightly bigger than  $o$  with the help of exhaustive search—the overall complexity then becomes  $O(o^4 q^{v-o-1})$ . The experiments from Braeken et al. (2005) show that if direct Gröbner basis attacks can be efficient against the balanced Oil and Vinegar scheme, it is of exponential complexity in the unbalanced case. Faugère and Perret (2008b) also showed that it is possible to attack some set of parameters by computing Gröbner basis of several modified versions of the original system. It is however possible to select the parameters of the system so as to escape this signature forgery attack.

Several other asymmetric multivariate schemes are closely related to the Oil and Vinegar construction like for instance the signatures schemes Rainbow (Ding and Schmidt 2005a) and TTS (Yang et al. 2004). It is not difficult to see that the balanced and unbalanced Oil and Vinegar constructions are broken as soon as an attacker is able to recover an isomorphic version of the secret oil vector space. Up to now, no such structural attack is known against the unbalanced schemes.

## 2.5 Defense Mechanisms

In the previous paragraph, we reviewed several attacks using system solving techniques against asymmetric multivariate cryptosystems. Several extensions have consequently been proposed to slow down these attacks by making inefficient the system solving algorithms. We now briefly describe the most widespread of these and discuss their effects. These experiments might be classified into two families: the removal of some information in the published mappings and the addition of randomly chosen quadratic polynomials.

### 2.5.1 Removing Equations

The idea of discarding some of the polynomials from the public key was originally introduced by Shamir (1993). Patarin later suggested to use it to strengthen HFE in the context of signature and called the resulting scheme HFE<sup>--</sup>. Patarin et al. (2000) designed a signature scheme by applying this idea to the original Matsumoto–Imai scheme A that they submitted to the NESSIE project. It seems that the effect of the removal of polynomials from the public key is quite efficient against the system solving threat: the second challenge on HFE is still unbroken and the NESSIE proposal SFLASH withstood all system solving attacks. Yet this is not enough for these schemes to be secure and an attack taking advantage of the underlying monomial structure of SFLASH has recently been found by Dubois et al. (2007a). This attack uses the associated bilinear form to regenerate the missing polynomials, and

thus allows for the application of the attack originally found by Patarin against Matsumoto–Imai scheme A. The applicability of analogous techniques to HFE<sup>--</sup> remains an open question. Furthermore, a rigorous analysis of the impact of removing equations from the public key of such schemes on the system solving techniques is still an open problem.

### 2.5.2 Perturbations

Another strategy devised to thwart system solving techniques is to perturb the public key by mixing additional randomly chosen multivariate quadratic polynomials to the public key. This strategy is quite natural since the problem of solving randomly chosen systems of multivariate quadratic equations is a hard problem. Let us denote by  $g = (g_1, \dots, g_n)$  the  $n$ -tuple of polynomials corresponding to the original public key and  $\tilde{g} = (g_1 + q_1, \dots, g_n + q_n)$  the public key after the introduction of  $m$  randomly chosen polynomials  $\rho_1, \dots, \rho_m$ . The introduction of the random polynomials should obviously not disallow the legitimate user to invert the resulting public key. To this end, it is limited in one of the two following ways: either  $m = n$  and there exists some linear mapping  $\lambda : \mathbb{K}^n \rightarrow \mathbb{K}^r$  of rank  $r$  such that

$$q_i(x_1, \dots, x_n) = \rho_i \circ \lambda(x_1, \dots, x_n), \quad 1 \leq i \leq n,$$

or  $m = r$  and there is a linear mapping  $\lambda : \mathbb{K}^r \rightarrow \mathbb{K}^n$  of rank  $r$  such that

$$(q_1(x_1, \dots, x_n), \dots, q_n(x_1, \dots, x_n)) = \lambda(\rho_1(x_1, \dots, x_n), \dots, \rho_r(x_1, \dots, x_n)).$$

In the first type of perturbation, called internal perturbation, the random polynomials  $\rho_i$  only depend on a small number  $r$  of variables. Then given some cipher text  $c$  and knowing the polynomials  $\rho_i$ , it is enough for the legitimate user to compute the value  $z = \rho(w)$  for all the possible inputs  $w$  and try to invert the original public key  $g$  on the corresponding value  $c + z$ . In the second type of perturbation, often denoted by ‘+’, the random polynomials depend on all the variables  $x_1, \dots, x_n$  but there are only  $r$  of them and so their value can be guessed as well by the legitimate user.

The second strategy has been proposed by Patarin et al. (1998a) while the first one was later on suggested by Ding and Gower (2005). Once again, the effect of these perturbations against system solving techniques is not well understood and waits for a rigorous analysis. However, it is interesting to see that the proposal of Ding and Gower (2005) was again defeated by Fouque et al. (2005) by analyzing a distinguisher based on the kernel of the differential of the public key and extended their attack to the perturbed HFE in Dubois et al. (2007b).

## 3 Structural Attacks

In the previous part, we have reviewed several direct inversion attacks against various multivariate asymmetric cryptosystems. We now describe algebraic attacks

against the trapdoor's structure of some of these cryptosystems. The two basic mechanisms we focus on are the problem of finding isomorphisms between two sets of polynomials, and the problem of polynomials decomposition. The first problem is related to the problem of recovering the key of UOV cryptosystems and Matsumoto–Imai like cryptosystems such as HFE and SFLASH. This problem is also related to the study of substitution and permutation networks in symmetric cryptography (Biryukov et al. 2003). The second problem is a natural problem arising in multivariate cryptography. It has been used to design an interesting public encryption scheme (Patarin and Goubin 1997) mixing techniques from symmetric cryptography and multivariate polynomials to turn it into an asymmetric scheme.

### 3.1 Isomorphism of Polynomials

There is a natural equivalence class on the set of tuples of multivariate polynomials in  $n$  variables. For two  $m$ -tuples  $f = (f_1, \dots, f_m)$  and  $g = (g_1, \dots, g_m)$  of multivariate polynomials in  $n$  variables we say that  $f$  and  $g$  are equivalent if and only if there exists an invertible change of coordinates  $S$  such that  $f(x) = g \circ S(x)$ . This equivalence relation in the special case  $m = 1$  and with  $f_1$  and  $g_1$  multivariate quadratic polynomials exactly corresponds to the classification of multivariate quadratic forms, which has been completed by Dickson (1971); the problem of isomorphism between polynomials of degree  $d$  is studied in Thierauf (2000). The equivalence can be further generalized as follows: two  $m$ -tuples  $f$  and  $g$  are IP-equivalent if and only if there exist two invertible changes of coordinates  $S$  and  $T$  such that  $T \circ f(x) = g \circ S(x)$ . This second equivalence relation has been formally introduced by Patarin (1996) in cryptography and further studied by Patarin et al. (1998b). (However, Matsumoto and Imai already made the implicit assumption that this problem is hard when they designed their scheme A.) Thus, the computational problem associated to deciding the IP-equivalence can be stated as follows:

**Definition 2** (IP Problem) Given  $f$  and  $g$ , two  $m$ -tuples of multivariate polynomials in  $n$  variables, find two invertible linear mappings  $S \in \text{GL}_n(\mathbb{K})$  and  $T \in \text{GL}_m(\mathbb{K})$  such that:

$$g(x_1, \dots, x_n) = T \circ f \circ S(x_1, \dots, x_n). \quad (9)$$

One might wonder why not keep the map  $f$  secret and only publish  $g$ . The reason is that in multivariate asymmetric cryptosystems, the existence of a trapdoor considerably reduces the number of possible mappings  $f$ . For instance, only a few monomial can be used as the internal transformation in Matsumoto–Imai scheme A. It is therefore safer to assume that map  $f$  is also publicly known.

It has been proved in Patarin et al. (1998b) that deciding IP-equivalence is not NP-complete. It was also shown in the same paper that deciding another equivalence—which has been called MP-equivalence for it does not require the linear mappings  $S$  and  $T$  to be invertible—is NP-hard. Finally, the authors of Patarin

et al. (1998b) reduced the problem of deciding graphs isomorphism to the problem of deciding for two  $m$ -tuples of quadratic multivariate polynomials  $f$  and  $g$  the existence of a linear mapping  $S$  (not necessarily invertible) such that  $g(x) = f \circ S(x)$ . The problem of deciding graphs isomorphism is a well known problem in complexity theory and it is used to define a whole complexity class which is thought to be disjoint both from P and NP-complete although this has not yet been proven. However, while most practical instances of the graph isomorphism problem are easy to solve, most practical instances of IP seem to be difficult to solve.

Thus, the IP-equivalence seems to be a good candidate to be used as a hard problem in cryptology. We already mentioned that the security of Matsumoto–Imai like cryptosystems rely on this problem, but other types of cryptosystems can be built based on the hardness of the IP problem like for instance the authentication scheme proposed by Patarin (1996), or the traitor tracing scheme proposed by Billet and Gilbert (2003). The IP problem is also of interest in symmetric cryptography where it was studied as a means to derive equivalent descriptions of block ciphers, and as a way of describing big S-boxes by substitution and permutation networks with much smaller S-boxes in order to ease their analysis (Biryukov et al. 2003).

There have been several algorithms designed to solve the IP problem, most of which are described in Patarin et al. (1998b), Biryukov et al. (2003), Perret (2005), Faugère and Perret (2006a). The best algorithm from Patarin et al. (1998b) to solve instances of the IP problem is based on a “to and fro” algorithm and has a complexity of  $n^{O(1)}q^{\frac{n}{2}}$  both in time and memory; However, this algorithm only work for (almost) one-to-one mappings and the above mentioned complexity relates to the case of quadratic polynomials. In the case of non bijective mappings, another algorithm proposed in Patarin et al. (1998b) has polynomial complexity in memory and  $n^{O(1)}q^n$  in time. The algorithm designed by Biryukov et al. (2003) share some features with the “to and fro” strategy and basically has the same time complexity. Biryukov et al. (2003) also contains a generalization to the affine setting. Perret also presented in Perret (2005) an algorithm for the simple equivalence of polynomials with has a time complexity lower bounded by  $n^6q^n$ . We focus here on an algorithm presented in Faugère and Perret (2006a) since it amounts to solve a system of equations; unfortunately, its complexity is not well understood. First of all, let us summarize the basic solving problem one is faced with the IP problem: assuming an internal transformation that consists of an  $m$ -tuple of multivariate polynomials of degree  $d$  in  $n$  variables and using additional variables to describe the unknown changes of coordinates, (9) gives a set of equations in the variables representing the change of coordinates. A quick counting of these equations shows that the system is over-defined with  $m\binom{n+d}{d}$  equations in  $m^2 + n^2$  variables when  $m$  is about  $n$  as is the case with several multivariate asymmetric cryptosystems. However, as the overall degree  $d$  increases, the number of equations and terms the attacker has to deal with increases at fast pace. This basic way to put the IP problem into equations can actually be much improved in the case where the internal transformation  $f$  contains monomials of low degree—constant, linear, or quadratic—so as to be independent of the overall degree  $d$ . Such a strategy has been proposed in Faugère and Perret (2006a) in the case of non homogeneous systems and is described hereafter. First of

all, notice that (9) arising from the IP problem can be rewritten as:

$$T^{-1} \circ g(x_1, \dots, x_n) = f \circ S(x_1, \dots, x_n), \quad (10)$$

so that using variables for the unknown entries of the matrices corresponding to  $S$  and  $T^{-1}$  gives a lower total degree in the resulting equations. Let us denote these variables by  $s_{1,1}, \dots, s_{n,n}$  and  $t_{1,1}, \dots, t_{m,m}$  respectively. Then taking advantage of the fact that the internal transformation is not homogeneous, the above equation also holds for the homogeneous parts alone:  $\forall k \leq d$ ,  $T^{-1} \circ g^{(k)}(x) = f^{(k)} \circ S(x)$ , where  $g^{(k)}$  and  $f^{(k)}$  denotes the homogeneous part of degree  $k$  of  $g$  and  $f$ . Thus, when the internal transformation has both a constant component and a degree one component, a lot more linear constraints in the variables  $s_{i,j}$  and  $t_{i,j}$  can be derived. But this set of  $m(n+1)$  linear equations in the  $n^2 + m^2$  variables is not big enough to be over-defined. One has to adjunct another set of equations derived from a component of higher degree, usually a component of homogeneous degree two: these additional equations then suffice to render the system over-defined, in most cases of interest. (This is for instance the case with an internal transformation consisting of an  $n$ -tuple of quadratic multivariate polynomials in  $n$  variables which is quite representative in asymmetric multivariate cryptography.) This is the reason why the IP problem with internal transformations composed of low degree monomials is insensitive to the value of the overall degree  $d$ . However, this is *not true at all* for IP problems with homogeneous internal transformations of degree  $d$ , which explains the discrepancies between experiments with Matsumoto–Imai scheme A of degree four and experiments with randomly generated polynomials (with components of every degree) of overall degree four in the results of Faugère and Perret (2006a). It is not straightforward to derive the complexity of the strategy just described. However, experimental results from Faugère and Perret (2006a) show that the complexity of the IP problem for cryptographic purposes has sometimes been over-estimated (Patarin 1996; Billet and Gilbert 2003; Patarin et al. 1998b).

A powerful attack against the IP problem in the special case of the Matsumoto and Imai scheme A has also been proposed in Fouque et al. (2008b) and allows to recover the secret key of the Matsumoto and Imai scheme A, not only to invert it. This attack builds on a previous attack against SFLASH (Dubois et al. 2007a) and only uses efficient linear algebra. Finally, there has been no success up to now in attacking the IP problem underlying the HFE cryptosystem.

### 3.2 Two Rounds

We have seen in the previous sections that embedding a trapdoor in a tuple of quadratic multivariate polynomials is not an easy task. A natural way to try circumventing the difficulty is to rely on the composition of two multivariate mappings  $f$  and  $g$ . The first proposal based on such a strategy can be found in Patarin and Goubin (1997). In order to ease the exposition, we only describe a restricted version of it. It makes use of three mappings  $f = (f_1, f_2, \dots, f_n)$ ,  $U$ , and  $g = (g_1, g_2, \dots, g_n)$

where the  $f_i$  and  $g_i$  are  $k$ -tuples of multivariate quadratic polynomials in  $k$  variables and  $U$  is a change of coordinates over  $\mathbb{K}^{kn}$ . Thus, the published mapping is the composition  $T \circ g \circ U \circ f \circ S$  where  $S$  and  $T$  are additional changes of coordinates over  $\mathbb{K}^{kn}$ . This proposal, called two rounds by their designers, can be thought of as an asymmetric version of the substitution and permutation network construction classical in symmetric cryptography where the  $f_i$  and  $g_i$  play the role of S-boxes. (Note that these mappings  $f_i$  and  $g_i$  are not required to be one-to-one.) Obviously, those S-boxes can be easily inverted when considered alone. Thus, the security of the proposed scheme heavily relies on the hardness of the problem of decomposition of the public mapping:

**Definition 3** (Decomposition Problem) Given a set of  $n + 1$  multivariate polynomials  $f, h_1, \dots, h_n$ , in  $n$  variables defined over some finite field  $\mathbb{K}$ , find (provided it exists) a polynomial  $g$  of degree  $r$  such that:

$$f(x_1, \dots, x_n) = g(h_1(x_1, \dots, x_n), \dots, h_n(x_1, \dots, x_n)).$$

For multivariate polynomials of arbitrary degree, this problem is often assumed to be difficult to solve Dickerson (1989), von zur Gathen et al. (2003), as expected by the authors of two rounds. The decisional version of the decomposition problem is also sometimes referred to as the ring membership problem since it amounts to deciding the membership of  $f$  to the ring  $\mathbb{K}[h_1, \dots, h_n]$  restricted to polynomials of degree  $r$ . However for efficiency reasons, the degree  $r$  of  $g$  is assumed to be two in the two round scheme, and in this case, the corresponding decomposition problem becomes easy. Ye, Lam, and Dai indeed proposed in Ye et al. (1999, 2001) an efficient strategy to solve it based on the following simple remark: when the degree of  $g$  is two, the partial derivatives of  $f$  are nothing but elements  $l_i h_j$  where  $l_i$  is a linear form and thus span an ideal  $\Delta_f$  contained in  $\langle x_1 h_1, \dots, x_n h_1, x_1 h_2, \dots, x_n h_n \rangle$ . Hence, computing  $(\Delta_f : \langle x_1, \dots, x_n \rangle)$  (that is, the set of polynomials  $p$  such that  $Lp$  lies in  $\Delta_f$  for every linear form  $L$ ) reveals  $\langle h_1, \dots, h_n \rangle$ . This fact was verified by experiments by the authors of Ye et al. (1999). To complete the attack, a basis of this last ideal gives an  $n$ -tuple  $\tilde{h} = (\tilde{h}_1, \dots, \tilde{h}_n)$  where the  $\tilde{h}_i$  are linear combinations of the original polynomials  $h_i$ , which is enough to recover—by interpolation for instance—the remaining mapping  $\tilde{g}$  such that  $\tilde{g} \circ \tilde{h} = f$ .

The authors of Patarin and Goubin (1997) tweaked their original construction so as to thwart this new threat and proposed to remove several public equations of two rounds, so that  $f = g \circ h$  and  $h$  are mappings in  $n$  variables, but  $f$  and  $g$  are  $m$ -tuples of multivariate polynomials in  $n$  variables with  $m < n$ . But Faugère and Perret (2006b) refined the ideas of Ye et al. (1999) and showed that the scheme can still be cryptanalysed. Let us briefly describe their strategy: the basic idea is to compute  $(\Delta_f : x_n^\delta)$  for some well chosen  $\delta > 0$ . Indeed, the relations:

$$\frac{\partial f_i}{\partial x_j} = \sum_{1 \leq k, l \leq n} g_{k,l}^{(i)} \left( \frac{\partial h_k}{\partial x_j} h_l + \frac{\partial h_l}{\partial x_j} h_k \right) \quad (11)$$

show that any linear combination of polynomials of the form  $z^{[\delta-1]} \frac{\partial f_i}{\partial x_j}$ , where  $z^{[\delta-1]}$  stands for any monomial of degree  $\delta - 1$  in the variables  $x_i$ , is also a linear combination of polynomials of the form  $z^{[\delta]} h_i$ . If  $V$  denotes the vector space spanned by the polynomials of the form  $z^{[\delta]} h_i$  and  $\tilde{V}$  denotes the vector space spanned by the polynomials of the form  $z^{[\delta-1]} \frac{\partial f_i}{\partial x_j}$ , then  $x_n^\delta h_i$  belongs to  $\tilde{V}$  for all  $i$  as soon as the dimension of  $\tilde{V}$  as a vector space over  $V$  is at least  $n \binom{n+\delta-1}{\delta}$ . Thus, the computation of a Gröbner basis of  $(\Delta_f : x_n^\delta)$  provides the  $n$ -tuple  $(\tilde{h}_1, \dots, \tilde{h}_n)$  we were seeking. Faugère and Perret (2006b) also give an upper bound for the degree  $\delta$  which helps evaluate the complexity of the Gröbner basis computation: the attack succeeds as soon as  $\delta \geq \frac{m}{n}$ . Thus, the results of Ye et al. (1999) come as the special case  $m = n$ .

## 4 Discussion

This overview of the state of the art in the cryptanalysis of multivariate asymmetric cryptosystems shows that system solving techniques brought a lot to the understanding of multivariate cryptosystems. It helped uncover structural properties of those schemes and pushed the limits of our knowledge with respect to some difficult problems such as the functional decomposition problem or the problem of finding isomorphisms between tuple of polynomials. The extensive experiments with the computation of Gröbner basis of randomly generated systems of polynomials together with the mathematical insights brought by the complexity analyzes from Bardet (2004) yield useful tools for dimensioning symmetric multivariate cryptosystems such as Berbain et al. (2006, 2007).

While several multivariate asymmetric schemes have been shown to be susceptible to some extent to Gröbner basis techniques, a lot of these attacks still lack rigorous complexity analysis. Several of them remain slow and progresses in the understanding of system solving techniques as applied to multivariate asymmetric cryptosystems would be of interest to the cryptographers' community. It also has to be emphasized that the cryptanalytic work performed against asymmetric multivariate cryptosystems has already benefited other areas of cryptography such as the cryptanalysis of stream ciphers which as witness a new range of attacks called algebraic attacks (Faugère and Ars 2003; Courtois and Meier 2003).

Apart from obtaining a better understanding of existing attacks, there are several other challenges for the cryptanalysts. Concerning the multivariate schemes, the unbalanced Oil & Vinegar scheme remains unbroken. Furthermore, the effect of removing equations from the public key was shown to be inefficient in the case of the SFLASH cryptosystem and the natural following step is to settle the case of HFE<sup>––</sup>. On the side of the underlying hard problems, the functional decomposition problem has been shown to be useless to design cryptosystems but the problem of finding isomorphisms between tuples of polynomials needs a lot more study. In particular, cryptographers need a better understanding of the mechanisms behind the attack from Faugère and Perret (2006a) and a natural question is the possibility of mounting a key recovery attack against the HFE cryptosystem, at least with a rigorous complexity analysis.

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

This work has been supported in part by the French government through the ANR MAC project.

## References

- G. Ars, J. C. Faugère, H. Imai, M. Kawazoe, and M. Sugita, *Comparison between XL and Gröbner basis algorithms*, Proc. of Asiacrypt 2004 (P. J. Lee, ed.), LNCS, vol. **3329**, Springer, Berlin, 2004, pp. 338–353.
- M. Bardet, *An investigation on overdetermined algebraic systems and applications to error-correcting codes and to cryptography*, Ph.D. thesis, University of Paris 6, Paris, France, 2004.
- C. Berbain and H. Gilbert, *On the security of IV dependent stream ciphers*, FSE 2007 (A. Biryukov, ed.), LNCS, vol. **4593**, Springer, Berlin, 2007, pp. 254–273.
- C. Berbain, H. Gilbert, and J. Patarin, *QUAD: A practical stream cipher with provable security*, EUROCRYPT 2006 (S. Vaudenay, ed.), LNCS, vol. **4004**, Springer, Berlin, 2006, pp. 109–128.
- O. Billet and H. Gilbert, *A traceable block cipher*, Asiacrypt 2003 (C. S. Laih, ed.), LNCS, vol. **2894**, Springer, Berlin, 2003, pp. 331–346.
- O. Billet and H. Gilbert, *Cryptanalysis of Rainbow*, SCN 2006 (R. De Prisco and M. Yung, eds.), LNCS, vol. **4116**, Springer, Berlin, 2006, pp. 336–347.
- O. Billet, M. J. B. Robshaw, and T. Peyrin, *On building hash functions from multivariate quadratic equations*, ACISP 2007 (J. Pieprzyk, H. Ghodosi and E. Dawson, eds.), LNCS, vol. **4586**, Springer, Berlin, 2007, pp. 82–95.
- O. Billet, J. Patarin, and Y. Seurin, *Analysis of Intermediate Field Systems*, SCC 2008 (D. Wang and J.-C. Faugère, eds.), 2008.
- A. Biryukov, B. Preneel, A. Braeken, and C. de Cannière, *A toolbox for cryptanalysis: linear and affine equivalence algorithms*, Eurocrypt 2003 (E. Biham, ed.), LNCS, vol. **2656**, Springer, Berlin, 2003, pp. 33–50.
- A. Braeken, B. Preneel, and C. Wolf, *A study of the security of unbalanced Oil & Vinegar signature schemes*, CT-RSA 2005 (A. Menezes, ed.), LNCS, vol. **3376**, 2005, p. 29.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- J. F. Buss, G. S. Frandsen, and J. O. Shallit, *The computational complexity of some problems of linear algebra*, J. Comput. Syst. Sci. **58** (1999), no. 3, 572–596.
- M. Caboara, F. Caruso, and C. Traverso, *Gröbner bases for public key cryptography*, Proc. of ISSAC 2008 (L. Gonzalez-Vega, ed.), ACM, New York, 2008.
- D. Coppersmith, J. Stern, and S. Vaudenay, *Attacks on the birational permutation signature schemes*, CRYPTO93 (D. R. Stinson, ed.), LNCS, vol. **773**, Springer, Berlin, 1993, pp. 435–443.
- D. Coppersmith, J. Stern, and S. Vaudenay, *The security of the birational permutation signature schemes*, Journal of Cryptology **10** (1997), no. 3, 207–221.

- N. T. Courtois, *The security of Hidden Field Equations (HFE)*, Proc. of CT-RSA 2001 (D. Naccache, ed.), LNCS, vol. **2020**, Springer, Berlin, 2001, pp. 266–281.
- N. Courtois and W. Meier, *Algebraic attacks on stream ciphers with linear feedback*, EUROCRYPT 2003 (E. Biham, ed.), LNCS, vol. **2656**, Springer, Berlin, 2003, pp. 345–359.
- N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient algorithms for solving overdefined systems of multivariate polynomial equations*, Proc. of EUROCRYPT 2000, LNCS, vol. **1807**, Springer, Berlin, 2000, pp. 392–407.
- M. T. Dickerson, *The functional decomposition of polynomials*, Ph.D. thesis, Cornell University, Ithaca, NY, USA, 1989.
- L. E. Dickson, *History of the theory of numbers*, vol. **3**, Chelsea, New York, 1971.
- J. Ding and J. E. Gower, *Inoculating multivariate schemes against differential attacks*, Cryptology ePrint Archive, Report 2005/255, 2005.
- J. Ding and D. Schmidt, *A defect of the implementation schemes of the TTM cryptosystem*, Cryptology ePrint Archive, Report 2003/085, 2003.
- J. Ding and D. Schmidt, *The new implementation schemes of the TTM cryptosystem are not secure*, Progr. Comput. Sci. Appl. Logic **23** (2004), 113–127.
- J. Ding and D. Schmidt, *Rainbow, a new multivariable polynomial signature scheme*, ACNS 2005 (J. Ioannidis, A. D. Keromytis and M. Yung, eds.), LNCS, vol. **3531**, Springer, Berlin, 2005a, pp. 164–175.
- J. Ding and D. Schmidt, *Cryptanalysis of HVEv and internal perturbation of HFE*, PKC 2005 (S. Vaudenay, ed.), LNCS, vol. **3386**, Springer, Berlin, 2005b, p. 288.
- J. Ding, L. Hu, X. Nie, J. Li, and J. Wagner, *High order linearization equation (HOLE) attack on multivariate public key cryptosystems*, PKC 2007 (T. Okamoto and X. Wang, eds.), LNCS, Springer, Berlin, 2007a.
- J. Ding, C. Wolf, and B.-Y. Yang,  *$\ell$ -invertible cycles for multivariate quadratic (MQ) public key cryptography*, PKC 2007 (T. Okamoto and X. Wang, eds.), LNCS, vol. **4450**, Springer, Berlin, 2007b, pp. 266–281.
- V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern, *Practical cryptanalysis of SFLASH*, CRYPTO 2007 (A. Menezes, ed.), LNCS, vol. **4622**, Springer, Berlin, 2007a, pp. 1–12.
- V. Dubois, L. Granboulan, and J. Stern, *Cryptanalysis of HFE with internal perturbation*, PKC 2007 (T. Okamoto and X. Wang, eds.), LNCS, vol. **3494**, Springer, Berlin, 2007b.
- J. C. Faugére, *A new efficient algorithm for computing Gröbner bases ( $F_4$ )*, J. Pure Appl. Algebra **139** (1999), nos. 1–3, 61–88.
- J. C. Faugére, *A new efficient algorithm for computing Gröbner bases without reduction to zero ( $F_5$ )*, Proc. of ISSAC 2002, ACM, New York, 2002, pp. 75–83.
- J. Faugére and G. Ars, *An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases*, INRIA Research Report 4739, 2003.
- J. C. Faugére and A. Joux, *Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases*, LNCS, vol. **2729** Springer, Berlin, 2003, pp. 44–60.
- J. C. Faugére and L. Perret, *Polynomial equivalence problems: algorithmic and theoretical aspects*, EUROCRYPT 2006, LNCS, vol. **4004**, Springer, Berlin, 2006a, pp. 30–47.
- J. C. Faugére and L. Perret, *Cryptanalysis of 2R<sup>+</sup> schemes*, CRYPTO 2006 (C. Dwork, ed.), LNCS, vol. **4117**, Springer, Berlin, 2006b, pp. 357–372.
- J.-C. Faugére and L. Perret, *Cryptanalysis of MinRank*, CRYPTO 2008 (D. Wagner, ed.), LNCS, vol. **5157**, Springer, Berlin, 2008a, pp. 280–296.
- J.-C. Faugére and L. Perret, *On the security of UOV*, SCC 2008 (D. Wang and J. C. Faugére, eds.), 2008b.
- H. J. Fell and W. Diffie, *Analysis of a public key approach based on polynomial substitution*, CRYPTO 85 (H. C. Williams, ed.), LNCS, vol. **218**, Springer, Berlin, 1985, pp. 340–349.
- M. Fellows and N. Koblitz, *Combinatorial cryptosystems galore!*, Finite Fields: Theory, Applications, and Algorithms (G. L. Mullen and P. J.-S. Shiue, eds.), Contemporary Mathematics, vol. **168**, AMS, Providence, 1994, pp. 51–61.
- P.-A. Fouque, L. Granboulan, and J. Stern, *Differential cryptanalysis for multivariate schemes*, EUROCRYPT 2005 (R. Cramer, ed.), LNCS, vol. **3494**, Springer, Berlin, 2005, pp. 341–353.

- P. A. Fouque, G. Macario-Rat, L. Perret, and J. Stern, *Total break of the -IC signature scheme*, PKC 2008, LNCS, vol. **4939**, Springer, Berlin, 2008a, pp. 1–17.
- P.-A. Fouque, G. Macario-Rat, and J. Stern, *Key recovery on hidden monomial multivariate schemes*, EUROCRYPT 2008 (N. P. Smart, ed.), LNCS, vol. **4965**, Springer, Berlin, 2008b, pp. 19–30.
- A. S. Fraenkel and Y. Yesha, *Complexity of solving algebraic equations*, Inf. Process. Lett. **10** (1980), nos. 4–5, 178–179.
- W. Geiselmann, R. Steinwandt, and T. Beth, *Attacking the affine parts of SFLASH*, Cryptography and coding—IMA 2001, Springer, Berlin, 2001, pp. 355–359.
- L. Goubin, *Théorie et Pratique de la Cryptologie sur Carte à Microprocesseur*, Mémoire d’habilitation à diriger des recherches, 2003.
- L. Goubin and N. T. Courtois, *Cryptanalysis of the TTM cryptosystem*, ASIACRYPT 2000 (T. Okamoto, ed.), LNCS, vol. **1976**, Springer, Berlin, 2000, pp. 44–57.
- L. Granboulan, A. Joux, and J. Stern, *Inverting HFE is quasipolynomial*, CRYPTO2006 (C. Dwork, ed.), LNCS, vol. **4117**, Springer, Berlin, 2006, pp. 345–356.
- H. Imai and T. Matsumoto, *Algebraic methods for constructing asymmetric cryptosystems*, Proc. of AAECC 3, LNCS, vol. **229**, Springer, Berlin, 1985, pp. 108–119.
- X. Jiang, J. Ding, and L. Hu, *Kipnis-Shamir’s attack on HFE revisited*, Inscrypt 2007 (D. Feng and Y. Zhang, eds.), LNCS, Springer, Berlin, 2007.
- A. Kipnis and A. Shamir, *Cryptanalysis of the oil & vinegar signature scheme*, CRYPTO ’98, LNCS, vol. **1462**, Springer, Berlin, 1998, pp. 257–266.
- A. Kipnis and A. Shamir, *Cryptanalysis of the HFE public key cryptosystem by relinearization*, CRYPTO 99 (M. J. Wiener, ed.), LNCS, vol. **1666**, Springer, Berlin, 1999, pp. 19–30.
- A. Kipnis, J. Patarin, and L. Goubin, *Unbalanced oil & vinegar signature schemes*, EUROCRYPT ’99 (J. Stern, ed.), LNCS, vol. **1592**, Springer, Berlin, 1999, pp. 206–222.
- D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Third ed., Addison-Wesley, Reading, 1997.
- N. Koblitz, *Algebraic aspects of cryptography*, Algorithms and computation in mathematics, vol. **3**, Springer, Berlin, 1999.
- F. Levy-dit-Vehel, M. G. Marinari, L. Perret, and C. Traverso, *A survey on Polly Cracker systems*, this volume, 2009, pp. 285–305.
- F. S. Macaulay, *The algebraic theory of modular systems*, Cambridge University Press, Cambridge, 1916.
- R. J. McEliece, *A public key cryptosystem based on algebraic coding theory*, JPL DSN **42–44** (1978), 114–116.
- T. T. Moh, *A fast public key system with signature and master key functions*, Proc. of CrypTEC99, Hong Kong City Press, 1999.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- P. Q. Nguyen and J. Stern, *The two faces of lattices in cryptology*, CaLC 2001 (J. H. Silverman, ed.), LNCS, vol. **2146**, Springer, Berlin, 2001, pp. 146–180.
- H. Niederreiter, *Knapsack-type cryptosystems and algebraic coding theory*, Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform. **15** (1986), no. 2, 159–166.
- J. Patarin, *Cryptoanalysis of the Matsumoto and Imai public key scheme of Eurocrypt ’88*, CRYPTO 95 (D. Coppersmith, ed.), LNCS, vol. **963**, Springer, Berlin, 1995, pp. 248–261.
- J. Patarin, *Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms*, EUROCRYPT ’96 (U. M. Maurer, ed.), LNCS, vol. **1070**, Springer, Berlin, 1996, pp. 33–48.
- J. Patarin, *The oil & vinegar signature scheme*, Proc. of Dagstuhl Workshop on Cryptography, 1997.
- J. Patarin, *Challenge HFE*, <http://www.minrank.org/hfe#challenge>, 1998.
- J. Patarin and L. Goubin, *Asymmetric cryptography with S-boxes*, ICICS 97, LNCS, vol. **1334**, Springer, Berlin, 1997, pp. 369–380.
- J. Patarin, L. Goubin, and N. T. Courtois,  *$C^{*-+}$  and HM: variations around two schemes of T. Matsumoto and H. Imai*, ASIACRYPT ’98 (K. Ohta and D. Pei, eds.), LNCS, vol. **1514**, Springer, Berlin, 1998a, pp. 35–49.

- J. Patarin, L. Goubin, and N. T. Courtois, *Improved algorithms for isomorphisms of polynomials*, EUROCRYPT 98 (K. Nyberg, ed.), LNCS, vol. **1403**, Springer, Berlin, 1998b, pp. 184–200.
- J. Patarin, L. Goubin, and N. T. Courtois, *SFLASH, a Fast Asymmetric Signature Scheme for Low Cost Smart-Cards*, <https://www.cosic.esat.kuleuven.be/nessie/workshop/submissions/sflash.zip>, 2000.
- L. Perret, *A fast cryptanalysis of the isomorphism of polynomials with one secret problem*, EUROCRYPT 2005 (R. Cramer, ed.), LNCS, vol. **3494**, Springer, Berlin, 2005, pp. 354–370.
- O. Regev, *Lattice-based cryptography*, Proc. of CRYPTO2006 (C. Dwork, ed.), LNCS, vol. **4117**, Springer, Berlin, 2006, pp. 131–141.
- A. Shamir, *Efficient signature schemes based on birational permutations*, CRYPTO93 (D. R. Stinson, ed.), LNCS, vol. **773**, Springer, Berlin, 1993, pp. 1–12.
- P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. **26** (1997), 1484–1509.
- T. Thierauf, *The computational complexity of equivalence and isomorphism problems*, LNCS, vol. **1852**, Springer, Berlin, 2000, pp. 1–135.
- J. von zur Gathen and V. Shoup, *Computing Frobenius Maps and Factoring Polynomials*, Computational Complexity **2** (1992), 187–224.
- J. von zur Gathen, J. Gutierrez, and R. Rubio, *Multivariate polynomial decomposition*, Appl. Algebra Eng. Commun. Comput. **14** (2003), no. 1, 11–31.
- L.-C. Wang, B.-Y. Yang, Y.-H. Hu, and F. Lai, A “Medium-Field” multivariate public-key encryption scheme, CT-RSA 2006 (D. Pointcheval, ed.), LNCS, vol. **3860**, Springer, Berlin, 2006, pp. 132–149.
- B.-Y. Yang and J.-M. Chen, *Building secure tame-like multivariate public-key cryptosystems: The new TTS*, ACISP 2005 (C. Boyd and J. M. G. Nieto, eds.), LNCS, vol. **3574**, Springer, Berlin, 2005, pp. 518–531.
- B.-Y. Yang, J.-M. Chen, and Y.-H. Chen, *TTS: High-speed signatures on a low-cost smart card*, CHES 2004 (M. Joye and J. J. Quisquater, eds.), LNCS, vol. **3156**, Springer, Berlin, 2004, pp. 371–385.
- D.-F. Ye, K.-Y. Lam, and Z.-D. Dai, *Cryptanalysis of “2 R” schemes*, CRYPTO 99, LNCS, vol. **1666**, Springer, Berlin, 1999, pp. 315–325.
- D.-F. Ye, Z.-D. Dai, and K.-Y. Lam, *Decomposing attacks on asymmetric cryptography based on mapping compositions*, J. of Cryptology **14** (2001), no. 2, 137–150.

# A Survey on Polly Cracker Systems

**Françoise Levy-dit-Vehel,  
Maria Grazia Marinari, Ludovic Perret and  
Carlo Traverso**

**Abstract** In 1993 Boo Barkee and others have written a paper “*Why you cannot even hope to use Gröbner Bases in Public Key Cryptography: an open letter to a scientist who failed and a challenge to those who have not yet failed.*” Since 1994, further attempts have been made, that gave rise to several cryptosystems now known as Polly Cracker systems. None of these proposals have been successful, and while Gröbner Bases are now an established tool for cryptanalysis, the challenge of Boo Barkee still stands w.r.t. the design point of view. We outline a description of how all these attempts have failed.

**Keywords** Polly Cracker systems · Combinatorial-algebraic cryptosystems

## 1 Introduction

Multivariate algebra plays a central role in today’s cryptography. The most popular public key cryptosystems based on multivariate polynomials are more or less related to the Matsumoto–Imai (1985) scheme, dating back to late eighties (see Billet and Ding 2009).

The Polly Cracker-like family, arising in the early nineties, proposed an alternative use of multivariate algebra in cryptography. Surprisingly enough, this concept became quite popular in the cryptographic community. In this paper, we survey the

---

F. Levy-dit-Vehel  
ENSTA, 32 boulevard Victor, 75739 Paris cedex 15, France  
e-mail: [levy@ensta.fr](mailto:levy@ensta.fr)

M.G. Marinari  
DIMA, Università di Genova, via Dodecaneso 35, 16146 Genova, Italy  
e-mail: [marinari@dima.unige.it](mailto:marinari@dima.unige.it)

L. Perret  
SALSA Project, INRIA, Centre Paris-Rocquencourt UPMC, Univ. Paris 06, CNRS,  
UMR 7606, LIP6, 104, avenue du Président Kennedy, 75016 Paris, France  
e-mail: [ludovic.perret@lip6.fr](mailto:ludovic.perret@lip6.fr)

C. Traverso  
Dipartimento di Matematica “Leonida Tonelli”, Università di Pisa, Via F. Buonarroti 2,  
56127 Pisa, Italy  
e-mail: [traverso@dm.unipi.it](mailto:traverso@dm.unipi.it)

constructions and results having appeared so far on Polly Cracker-like schemes. Our goal is to reevaluate the provocative assertion made more than ten years ago by Barkee et al. in their seminal paper (Mora 2003), namely that one “*cannot even hope to use Gröbner Bases in Public-key Cryptography*”. Note that, because of recent uses of Gröbner bases (Buchberger 1965, 1970, 1985, 1998, 2006) in cryptanalysis, it should rather be replaced by “Why one cannot hope to use Polly Cracker-like systems in cryptography”. We will try here to explain why this assertion is fundamentally valid today. In other words, why one cannot use such schemes for the design of secure and efficient cryptosystems.

In order to introduce Polly Cracker systems properly, let us first fix some notation (see Mora 2009a). We will denote by  $\mathbb{F}$  a field, frequently (but not always) finite of characteristic 2, and often  $\mathbb{F}_2$ .  $\mathcal{P}$  will be a polynomial ring,  $\mathcal{T}$  the monoid of its terms (power products), and we will assume that there is a term-ordering defined in  $\mathcal{P}$ , so that we can define the leading term  $\mathbf{T}(f)$  of a polynomial  $f$  and Gröbner bases;  $\mathbb{I}(G)$  denotes the ideal generated by a set  $G$  of polynomials;  $\mathbf{T}(\mathbb{I})$  and  $\mathbf{N}(\mathbb{I})$  will be respectively the sets of leading and normal terms of an ideal  $\mathbb{I}$ .

By a Polly Cracker-like system, we mean a system in which the secret key is a Gröbner basis of a multivariate ideal  $\mathbb{J} \subset \mathcal{P}$ , together with a subset  $S$  of the set of normal terms  $\mathbf{N}(\mathbb{J})$ . Let  $T$  be a set of terms mapping bijectively to  $S$  through the normal form map. The public key is composed of a set of polynomials of an ideal  $\mathbb{I} \subseteq \mathbb{J}$  and the set  $T$ ; the plaintext space is the vector space  $\text{Span}_{\mathbb{F}}(T)$  generated by  $T$ , and encryption of  $M \in \text{Span}_{\mathbb{F}}(T)$  is done by adding to  $M$  an element of  $\mathbb{I}$  chosen with a randomized procedure. Decryption of a ciphertext  $C$  consists in computing the canonical form of  $C$  w.r.t.  $\mathbb{I}$  using the Gröbner basis, and mapping back to  $\text{Span}(T)$ . The term-ordering is not public, but the security of the cryptosystem does not rely on the privacy of the term-ordering.

This family of systems includes the initial scheme of Barkee et al. (Sect. 2.1), the CA-style<sup>1</sup> systems of Fellows and Koblitz (Sect. 3), and several others, as well as the recently proposed non-commutative versions of Barkee et al.’s cryptosystem (Sect. 6.1). Remark that in several cases, including the systems of Fellows and Koblitz, for which the “Polly Cracker” name was first used, the private ideal is a maximal ideal, and the knowledge of a root is equivalent to the knowledge of a Gröbner basis of a maximal ideal.

To support our claim, we shall show that all the schemes proposed so far are vulnerable to attacks which are either of a cryptographic nature—oracle attacks, more precisely chosen ciphertext attacks—or of a structural one, i.e. exploiting the structure of the public-key to recover the plaintext or an equivalent secret key. It is to be noted that the oracle attacks also permit to recover a secret key, and it is not always clear how to design a padding scheme in order to prevent those.

The paper first presents the scheme of Barkee et al. (Sect. 2), which is the most general setting, as well as some generic attacks on it (Gröbner basis computation, linear algebra attack, chosen ciphertext attack). We then focus (Sect. 3)

---

<sup>1</sup>CA stands for Combinatorial-Algebraic.

on a family of schemes that are pretty close in design, namely the CA-style cryptosystems of Fellows and Koblitz (1994). Schemes in this class essentially differ from one another by their underlying hard problem: they mainly include graph theory problems, solving sparse multivariate systems over a finite field, as well as 3-SAT; each of those schemes was designed to patch an attack made on previous proposals, a chronology we adopt in our presentation. In that section, we also quote a somehow different scheme, namely Polly-two, which elegantly counters all previously known attacks on Polly Cracker-like cryptosystems. Unfortunately, we will see that this scheme is very sensitive to an enhanced structural attack.

Our last section is concerned with the extension of Polly Cracker schemes to algebras different from commutative polynomial algebras, yet allowing a Gröbner Basis theory; these include non-commutative free algebras. While sharing the weaknesses of the commutative schemes, they introduce new ones.

## 2 The Seminal Paper

In 1993, B. Barkee et al. wrote a paper (Barkee et al. 1994) whose aim was to dispel the urban legend that “Gröbner bases are hard to compute”. Another goal was to orient research on applications of Gröbner bases to cryptosystems towards the use of sparse multivariate schemes. To do so, they proposed the most obvious *dense* Gröbner-based cryptosystem.

### 2.1 Barkee’s Cryptosystem

Their pseudo-system consisted in first writing down an easy-to-produce Gröbner basis  $F = \{f_1, \dots, f_s\}$  generating an ideal  $\mathbb{I} := \mathbb{I}(F) \subset \mathcal{P}$ . This can be efficiently performed via Macaulay’s trick (Mora 2003, 2005). The public key is then a set  $G := \{g_1, \dots, g_\ell\} \subset \mathbb{I}(F)$  of *dense* polynomials of degree at most  $d$  in  $\mathcal{P}$  and a set

$$T := \{\tau_1, \dots, \tau_s\} \subset \mathbf{N}(\mathbb{I}(F)) = \mathcal{T} \setminus \mathbf{T}(\mathbb{I}(F))$$

of *normal terms* belonging to the Gröbner *escalier* of  $\mathbb{I}(F)$  either the whole of it, or, for added security, a subset of it (Barkee et al. 1994).

In order to encrypt a message  $M := \sum_{i=1}^s c_i \tau_i \in \text{Span}_{\mathbb{F}}(T)$ , the sender produces random *dense* polynomials  $p_j \in \mathcal{P}$ ,  $1 \leq j \leq \ell$ ,  $\deg(p_j) = r$  and encrypts  $M$  as

$$C := M + \sum_{j=1}^{\ell} p_j g_j.$$

The legitimate receiver—possessing the Gröbner basis of  $\mathbb{I}(F)$ —applies Buchberger’s reduction to obtain

$$\text{Can}(C, \mathbb{I}(F)) = M = \sum_{i=1}^s c_i \tau_i.$$

It is easy to realize that denoting, for each  $\delta \in \mathbb{N}$

$$\mathcal{T}(\delta) := \{\tau \in \mathcal{T} : \deg(\tau) \leq \delta\} \quad \text{and} \quad T(\delta) := \#\mathcal{T}(\delta) = \binom{\delta+n}{n},$$

both encoding and decoding costs between  $\mathcal{O}(T(d+r))$  (the time needed to scan a dense message) and  $\mathcal{O}(T^2(d+r))$  (the cost of Buchberger’s reduction algorithm in the generic case). The point of the paper was that an enemy would have been able to read the message without even attempting to perform the hard<sup>2</sup> Gröbner basis computation but with a more elementary linear-algebra based approach. Namely they proposed the following two attacks.

## 2.2 The Fantomas Attack

The *Fantomas attack* is based on the following result of the TERA community (Dickenstein et al. 1991). Let  $G := \{g_1, \dots, g_\ell\}$ ,  $\deg(g_i) \leq d$  and  $C$  be a polynomial, of degree  $\deg(C)$  smaller than or equal to  $d+r$ , for which  $C - \text{Can}(C, \mathbb{I}(F)) = \sum_{j=1}^\ell p_j g_j$  satisfies  $\deg(p_j) \leq r$ . It is then possible to compute  $\text{Can}(C, \mathbb{I}(F))$  by a modified version of Buchberger’s algorithm (each reduction of S-polynomials of degree higher than  $d+r$  being not performed).

The attacker does not know the exact value  $r$  since there could be highest-degree cancellation so that  $r > \deg(C) - d$  but this is not a problem: computations involving S-polynomials of degree higher than  $D := \deg(C)$  are *postponed* instead of not being performed; if the first round fails to return an element in  $\text{Span}_{\mathbb{P}}(T)$ , the algorithm sets  $D := D + 1$  and performs row reductions of S-polynomials of degree bounded by  $D$ . Repeating this procedure after  $r+d-\deg(C)$  rounds, the attacker finds both  $r$  and  $M$ . The Fantomas attack costs  $\mathcal{O}(T^4(d+r))$ , using a Buchberger’s algorithm truncated at degree  $d+r$ .

## 2.3 The Moriarty Attack

The method is based on the following observation. By the very construction of the ciphertext  $C$  it holds that

---

<sup>2</sup> $\mathcal{O}(T^4(\delta))$  where  $\delta := \max\{\deg(\tau) : \tau \in \mathbf{G}_<(\mathbb{I})\} = \mathcal{O}(d^{n/2^n})$ .

$$C = \sum_{j=1}^{\ell} p_j g_j + \sum_{\tau \in T} c_{\tau} \tau$$

We can solve this equation by regarding the coefficients of  $g_j$ s as unknowns and get linear equations by identifying the coefficients of the terms of  $C$  with the coefficients of the terms of  $\sum_{j=1}^{\ell} p_j g_j$ . More precisely, starting with  $D = \deg C$  and increasing it at every iteration, consider the  $p_j$  as polynomials of degree  $D - \deg(g_j)$ ,  $p_j = \sum_{\sigma \in T(D - \deg(g_j))} b_{j\sigma} \sigma$  and solve the equation with respect to the unknowns  $c_{\tau}, b_{j\sigma}$ , that appear linearly. If the system is not solvable, increase  $D$  and repeat. The system will be solvable when we reach the level  $r$  that was used in the encoding procedure. Being a *dense* linear algebra problem, the Moriarty Attack costs  $\mathcal{O}(T^3(d+r))$  with Gaussian algebra,  $\mathcal{O}(T^{2.4\dots}(d+r))$  with fast linear algebra.

## 2.4 Bulygin's Attack

We describe in this part a chosen ciphertext attack against Barkee's system. In this context, an attacker has access to a decryption oracle  $\mathcal{O}$  permitting to obtain the plaintext  $m$  corresponding to a ciphertext  $C$ , i.e.  $\mathcal{O}(C) = m$ . This attack is due to Bulygin (2005), and was originally described against a non-commutative version of Barkee's scheme (Sect. 6.1). In this attack, we suppose that the whole set  $\mathbf{T}(\mathbb{I}(F))$  is known. Note that this assumption is verified for the particular (non-commutative) scheme studied by Bulygin, but cannot be longer true in a more general context, i.e. Barkee's scheme. However, Alonso and Marinari (2008) recently proved that this assumption is somehow irrelevant, rendering then Bulygin's attack much more powerful. The basic idea of this attack is to remark that for each  $f_i \in F$ ,  $\text{Can}(\mathbf{T}(f_i), \mathbb{I}(F)) = f_i - \mathbf{T}(f_i)$ . He then builds fake ciphertexts

$$\tilde{C}_i := \sum_{j=1}^{\ell} p_j g_j + \mathbf{T}(f_i).$$

The decrypted version of this message is  $\text{Can}(\tilde{C}_i, \mathbb{I}(F)) = f_i - \mathbf{T}(f_i)$ , allowing then to obtain the polynomials  $f_i = \text{Can}(\tilde{C}_i, \mathbb{I}(F)) + \mathbf{T}(f_i)$  of the secret key.

### 2.4.1 Rai–Bulygin: Protecting Barkee's Scheme Against Bulygin's Attack

Bulygin and Rai (2006) remarked that it is not difficult to detect the fake ciphertexts  $\tilde{C}_i$ . They suggest to publish a subset  $T \subset \mathbf{N}(\mathbb{I}(F))$  such that:

$$(\mathbf{N}(\mathbb{I}(F)) \setminus T) \cap \text{supp}(f_i) \neq \emptyset, \quad \forall i, 1 \leq i \leq s.$$

The decryption procedure will be then modified such that an error message is returned as soon as the decrypted message  $M$  does not satisfy  $\text{supp}(M) \subset T$ .

This defense is very partial, since either  $T$  is considerably reduced with respect to  $N(\mathbb{I}(F))$  or there is a non-negligible probability that a crafted message is accepted, thus revealing some private information.

## 3 CA-Style Cryptosystems

### 3.1 Generic Design

At about the same time, and independently (Mora 1994) from the work of Barkee et al., Fellows and Koblitz (1994) proposed a framework for the design of public-key cryptosystems, the ideas of which are very similar to Barkee’s cryptosystem, but which differ on two essential aspects: first, the polynomials generating the public ideal are derived from combinatorial or algebraic NP-complete problems; such systems were thus naturally named CA-systems—for “combinatorial-algebraic” systems. Second, these schemes are bound to be *sparse*: the main reason is to render the linear algebra attack (2.3) exponential time (Koblitz 1998), another reason is to allow for a reasonable-size public key. Additionally, the secret key is not a Gröbner basis of the public ideal, but more simply a root of it, i.e. a Gröbner basis of a maximal ideal containing the public ideal. The main illustration of such systems was the Polly Cracker cryptosystem.

In this system, the public-key is a set  $G = \{g_1, \dots, g_\ell\} \subset \mathcal{P}$ , and the secret-key is a zero  $\alpha$  of the ideal  $\mathbb{I} = \mathbb{I}(G)$ . To encrypt a message  $m \in \mathbb{F}$ , one chooses random polynomials  $p_i \in \mathcal{P}$ ,  $1 \leq i \leq n$ , computes  $C = \sum_{i=1}^{\ell} p_i g_i + m$  and sends  $C$ . Knowing  $\alpha$  then allows the legitimate receiver to decrypt the ciphertext just by evaluating  $C(\alpha)$ .

The set  $G$  is an encoding of an instance of an NP-complete (combinatorial or algebraic) problem in such a way that knowing  $G$  is equivalent to knowing the considered instance, and that finding a secret-key from  $G$  is equivalent to finding a solution for this particular instance. M. Fellows and N. Koblitz suggest several NP-complete problems for use in this context, mainly based on graph theory, though not investigating the way of generating “hard” (random) solved instances of these problems.

#### 3.1.1 Effective Proposals and Basic Attacks

The first instantiation of CA-style cryptosystems use combinatorial problems on graphs. To implement such a system, one needs to choose a graph  $\Gamma$  having a combinatorial property, and define a basis  $G \subset \mathcal{P}$  such that the roots of the ideal  $\mathbb{I} := \mathbb{I}(G)$  are exactly all solutions to the problem.

### 3.2 Graph 3-Coloring

To set up the system, one needs to choose a graph  $\Gamma = (V, E)$ ,  $V = \{1, \dots, n\}$ ,  $E \subset \{\{i, j\}, 1 \leq i < j \leq n\}$ , for which a proper 3-coloring is known. A 3-coloring is a map  $\Phi : V \rightarrow \{1, 2, 3\}$  such that  $\{i, j\} \in E \implies \Phi(i) \neq \Phi(j)$ . A 3-coloring can be given by assigning  $\{0, 1\}$  values to a set of  $3n$  variables  $X_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq 3$  such that  $X_{i,j} = 1 \Leftrightarrow \Phi(i) = j$ . Set  $G := G_0 \cup G_1 \cup G_2 \cup G_3 \subset \mathbb{F}_2[X_{ic}, 1 \leq i \leq n, 1 \leq c \leq 3]$ , where

- $G_0 = \{X_{i1}X_{i2}, X_{i1}X_{i3}, X_{i2}X_{i3} : 1 \leq i \leq n\}$ —each vertex cannot be colored in two different ways;
- $G_1 = \{g_i := X_{i1} + X_{i2} + X_{i3} - 1 : 1 \leq i \leq n\}$ —each vertex has at least one color; thus, in connection with  $G_0$ , a single color;
- $G_2 = \{X_{i1}X_{j1}, X_{i2}X_{j2}, X_{i3}X_{j3}, \{i, j\} \in E\}$ —two connected vertices have different colors;
- $G_3 = \{X_{ic}^2 - X_{ic}, 1 \leq i \leq n, 1 \leq c \leq 3\}$ —the root components are either 0 or 1.  
Remark that this last set of equations can be derived from  $G_0$  and  $G_1$ .

The public key is  $G$ , while the secret key is a 3-coloring of  $\Gamma$ . In order to encrypt a message  $m \in \mathbb{F}_2$ , one randomly chooses a polynomial  $p$  in the ideal generated by the polynomials of  $G$ , and sets the ciphertext to be  $C = p + m$ . The recipient, who knows a 3-coloring of  $\Gamma$  and hence an  $\mathbb{F}_2$ -point  $\alpha$  of  $V(G)$ , simply computes  $C(\alpha) = p(\alpha) + m = m$ .

### 3.3 Graph Perfect Code

This time, one chooses a graph  $\Gamma = (V, E)$  (notation as above) which owns a *perfect code*. A perfect code is a subset  $V'$  of  $V$  such that for each  $u \in V$  the neighborhood  $N[u]$  of  $u$ ,  $N[u] := \{u\} \cup \{v \in V : \{u, v\} \in E\}$ , contains exactly one element in  $V'$ . A subset of  $V$  can be identified by the characteristic function, that in turn is a  $\{0, 1\}$  assignment to a set of variables  $X_u$ ,  $u \in V$ . Set  $G := G_0 \cup G_1 \cup G_2 \subset \mathbb{F}_2[X_u, 1 \leq u \leq n]$  where

- $G_0 = \{X_vX_w, v \neq w, v, w \in N[u], u \in V\}$ —for each  $u \in V$ ,  $\#(N[u] \cap V') \leq 1$ ;
- $G_1 = \{g_u := 1 - \sum_{v \in N[u]} X_v : 1 \leq u \leq n\}$ ;—in connection with  $G_0$  and  $G_2$  there is a single element in  $N[u] \cap V'$ ;
- $G_2 = \{X_{ic}^2 - X_{ic}, 1 \leq i \leq n, 1 \leq c \leq 3\}$ —the root components are either 0 or 1.  
Remark that these equations can be deduced from the previous ones.

The public key is  $G$ , the secret key is  $V'$ ; encryption and decryption are the same as for 3-coloring.

As shown in Endsuleit et al. (2002), the graph perfect code-based scheme with random chosen graphs is vulnerable to a Gröbner basis attack, where one computes a Gröbner basis  $G'$  of  $\mathbb{I}(G)$  and recovers  $m$  as  $\text{Can}(C, G')$ .

In both graph coloring and perfect code, elaborate randomized procedures are described, that might allow to reduce the sensitivity to the following linear algebra attack.

### 3.4 Intelligent Linear Algebra Attack

Soon after Fellows and Koblitz proposed their CA-systems, H.W. Lenstra (Koblitz 1998) noticed that the basic linear algebra attack (attack 2.3) could be improved by decreasing the number of unknowns in the following way: let

$$T(C) = \left\{ t \in \mathcal{T} : \exists t_g \in \bigcup_{j=1}^{\ell} \text{supp}(g_j), \exists t_C \in \text{supp}(C) \text{ such that } t_C = tt_g \right\}.$$

Roughly speaking,  $T(C)$  denotes the set of terms that Bob can potentially use to construct the given ciphertext  $C$ . If:

$$\bigcup_{j=1}^{\ell} \text{supp}(p_j) \subseteq T(C), \quad (1)$$

i.e. for each  $j$ ,  $1 \leq j \leq \ell$ , every term of  $p_j$  divides at least one term of the ciphertext  $C$ , then we can set, for each  $j$ ,

$$T_j := \{ \tau \in \mathcal{T} : \deg(g_j) : \exists \omega \in \text{supp}(g_j), v \in \text{supp}(C) : v = \tau \omega \},$$

where  $D := \deg(C)$ . It is then possible to reconstruct the polynomials  $p_j$  used to produce the ciphertext by solving the (smaller) system of linear equations which are the coefficients of each term in  $\mathcal{T}$  in the following polynomial equation

$$\sum_{\tau \in \mathcal{T}(D)} a_{\tau} \tau - \sum_{j=1}^{\ell} \left( \sum_{\tau \in T_j} b_{j\tau} \tau \right) g_j - m = 0$$

(where  $\sum_{\tau \in \mathcal{T}(D)} a_{\tau} \tau = C$ , the known received message), with unknowns

$$\{b_{j\tau} : \tau \in T_j, 1 \leq j \leq \ell\}.$$

The graph 3-coloring scheme was not designed to resist this attack; on the other hand, the graph perfect code-based scheme was Koblitz (1998): indeed, due to the form of its public key, it is always possible to construct a ciphertext for which condition (1) is not achieved—i.e. there exists at least one term  $t \in \bigcup_{j=1}^{\ell} \text{supp}(p_j)$  which does not divide any of the terms of the ciphertext.

### 3.5 EnRoot

Grant et al. (2000) proposed an encryption scheme based on the difficulty of finding a solution to a given system of sparse polynomial equations over a large finite field.

The system parameters are a large finite field  $\mathbb{F}_q$ , and (small) positive integers  $k, s_i, t_i, 1 \leq i \leq k$ . The legitimate recipient randomly chooses:

- $k$  polynomials  $h_i \in \mathbb{F}_q[X_1, \dots, X_n]$ ,  $1 \leq i \leq k$ , each  $h_i$  being of degree at most  $q - 1$ , with  $\#\text{supp}(h_i) < t_i$ ,
- non-zero elements  $a_1, \dots, a_n \in \mathbb{F}_q$ .

The public key consists of the polynomials  $f_i(X) = h_i(X) - h_i(a_1, \dots, a_n)$ ,  $1 \leq i \leq k$ , while the secret key is  $a = (a_1, \dots, a_n)$ .

To encrypt  $m \in \mathbb{F}_q$ , Bob randomly chooses  $k$  polynomials  $g_i \in \mathbb{F}_q[X_1, \dots, X_n]$ , each  $g_i$  being of degree at most  $q - 1$ , with  $\#\text{supp}(g_i) < s_i$ , such that  $g_i(0) \neq 0$ . He computes  $\Phi = \text{Can}(\sum_{i=1}^k g_i f_i, \mathbf{l})$ , where  $\mathbf{l} = (X_1^q - X_1, \dots, X_n^q - X_n)$ , and sends  $C = \Phi + m$ . The recipient, knowing  $a$ , decrypts  $C$  by evaluating it at  $a$ .

Soon after its publication, it has been shown in Bao et al. (2001) that the system is vulnerable to a 0-evaluation attack, the success of which is enhanced by the sparsity of the polynomials involved. In the paper of Banks et al. (2001) a variant of EnRoot is discussed, on which the same attacks are successful.

### 3.6 0-Evaluation Attack

The idea of the 0-evaluation attack is to reconstruct the polynomials  $p_i$  used by the sender to produce the ciphertext. In the case of a regular Polly Cracker system, we observe that, as the plaintext  $m$  is a constant, we have:

$$m = C(0) - \sum_{i=1}^{\ell} p_i(0)g_i(0). \quad (2)$$

Thus, to reveal  $m$ , it is sufficient to retrieve the constant terms of the polynomials  $p_i$  for those  $i$  for which  $g_i(0) \neq 0$ . As there are  $\binom{n+d}{d}$  terms of total degree less than  $d$  in  $n$  variables, one can assume<sup>3</sup> that the public polynomials  $g_i$  are not dense. Thus, one can suppose that each  $g_i$  contains a term  $t_i$  that does not occur in  $\text{supp}(g_j)$ , for  $j \neq i$  (such a term is called “characteristic term”). It is also probable that more than one characteristic term exists in every  $g_i$ .

A characteristic term  $\tau$  will probably appear in  $\sum p_j g_j$  and it is also likely that the only contribution to the sum will be the product of the monomial with term  $\tau$  in  $g_i$  times the constant term  $p_i(0)$  of  $p_i$ . So from any characteristic term of  $g_i$  we will obtain a guess for  $p_i(0)$ . If there are several characteristic terms in  $g_i$  it is likely that most of them will propose the same guess for  $p_i(0)$  while a few others will propose different values<sup>4</sup>.

For the graph 3-coloring instance of Polly Cracker, the polynomials of the public-key having non-zero constant terms are the  $g_i(X) = X_{i1} + X_{i2} + X_{i3} + 1$ ,  $1 \leq i \leq n$

<sup>3</sup>Otherwise this would result in a huge public key.

<sup>4</sup>Note that the above condition is likely to be satisfied if the polynomials  $p_i$  are sparse, a realistic assumption for efficient encryption. This is particularly the case for the EnRoot encryption scheme, where the polynomials are required to be very sparse and chosen at random.

(the other ones have total degree two and no non-constant term). Thus, it is sufficient to recover the  $p_i(0)$  for those  $i$  only. The terms  $X_{ij}$ ,  $1 \leq j \leq 3$ , can thus serve as characteristic terms of  $g_i$ . This scheme is therefore prone to this attack.

Note that the perfect code-based scheme is not, since, for any two vertices  $u$  and  $v$ ,  $d(u, v) \leq 2$ , we have  $\text{supp}(g_u) \cap \text{supp}(g_v) \neq \emptyset$ . This attack can sometimes be defeated by modifying the choice of the  $p_j$ 's, making the assumptions on which it relies false.

### 3.7 3-SAT

Having in mind to design a scheme resistant to the 0-evaluation attack, as well as to the intelligent linear algebra one, whilst respecting the building upon hard problems-spirit of CA-style systems, Levy-dit-Vehel and Perret (2004) proposed a scheme which relies on the satisfiability problem.

In this system, a finite field  $\mathbb{F}_q$  with  $q \geq 3$ , and positive integers  $m$  and  $n$  are chosen, as well as a random vector  $y$  of  $\{T, F\}^n$ ,  $T, F \in \mathbb{F}_q$ . The public key<sup>5</sup> is a formula, i.e. a conjunction of  $m$  clauses in  $n$  variables  $\mathcal{C} = \bigwedge_{i=1}^m C_i$ , admitting  $y$  as model (i.e. truth assignment making all  $C_i$  true);  $\mathcal{C}$  belongs to the class of so-called doubly-balanced 3-SAT formulae. Instances from this class are much more difficult to solve in general than random 3-SAT instances, as they are designed to have structural regularities, thus confusing variable selection heuristics that are used by most solvers (see Hirsch 2009 for a precise description of such formulae, and a generating method for those.) The parameters  $q$ ,  $n$  and  $m$  are also public. The private key is  $y$ .

The encryption phase follows the idea of a regular Polly Cracker scheme. But the practical realization is quite different from Fellows and Koblitz (1994). Denote by  $\{g_1, \dots, g_m\}$ , the polynomials constructed from the clauses  $\{C_1, \dots, C_m\}$ —clause  $C = X_j \vee \bar{X}_k \vee X_\ell$  yielding polynomial  $g_C(X) = (X_j - T)(X_k - F)(X_\ell - T)$ —and by  $I$ , the ideal generated by these polynomials. It is then clear that a satisfying truth assignment of  $X$  for  $C$  corresponds to a zero of the polynomial  $g_C(X)$ .

Let  $k \in \{1, \dots, m\}$ ,  $\{i_1, \dots, i_k\} \subset \{1, \dots, m\}$  and  $\{C_{i_j}\}_{1 \leq j \leq k}$  be a set of clauses. Denote by  $\{\text{Var}(C_{i_j})\}$ , the variables occurring in clause  $C_{i_j}$ .  $\{\text{Var}(C_{i_j})\}_{1 \leq j \leq k}$  is a disjoint set if for all  $a, b \in \{i_1, \dots, i_k\}$ ,  $a \neq b$ ,  $\text{Var}(C_a) \cap \text{Var}(C_b) = \emptyset$ . Note that, if  $\{\text{Var}(C_{i_j})\}_{1 \leq j \leq k}$  is a disjoint set, then  $\{g_{i_j}\}_{1 \leq j \leq k}$  is a reduced Gröbner basis of  $\langle g_{i_j} \rangle_{1 \leq j \leq k}$  for the degree lexicographical (*deglex*) order. To perform encryption, one needs and element from  $I$ ; it is constructed with the following algorithm:

---

<sup>5</sup>It would have been equivalent—from an information theoretic viewpoint—to publish the  $m$  polynomials corresponding to these  $m$  clauses, but the “clause-representation” allows for a more compact form.

**Algorithm 1**

**Input:**  $f \in \mathbb{F}_q[X]$ ,  $\ell \geq 2$ ,  $\{\lambda_1, \dots, \lambda_\ell\}$ ,  $\lambda_i \in \mathbb{F}_q$  with  $\sum_{i=1}^\ell \lambda_i \equiv 0[q]$  and  $\mathfrak{D} = \{\mathfrak{d}_1, \dots, \mathfrak{d}_\ell\}$  a set of indices subsets such that  $\forall 1 \leq i \leq \ell$ ,

$\{\text{Var}(C_{ij})\}_{j \in \mathfrak{d}_i}$  is a disjoint set.

**Output:** an element of the ideal  $\mathfrak{l}$ .

**For**  $i$  from 1 to  $\ell$  **do**

1. compute  $N_i(f) = \text{Can}(f, \mathfrak{J}_i)$ , where  $\mathfrak{J}_i = \langle g_j : j \in \mathfrak{d}_i \rangle$

**End For**

**Return**  $e_{\mathfrak{l}} = \sum_{i=1}^\ell \lambda_i N_i(f)$ .

To encrypt  $M \in \mathbb{F}_q$ , choose  $\beta = (\beta_1, \dots, \beta_n) \in \text{supp}(e_{\mathfrak{l}})$ ,  $\beta \neq 0$ , compute the ciphertext defined by  $C = e_{\mathfrak{l}} + MX_1^{\beta_1} \cdots X_n^{\beta_n}$ , and send  $(C, \beta)$ .

*Decryption*

Upon receiving  $(C, \beta)$ , the legitimate recipient evaluates  $\frac{C(y)}{y^\beta} = \frac{e_{\mathfrak{l}}(y) + My^\beta}{y^\beta} = M$  and recovers<sup>6</sup> the plaintext.

The polynomial  $f$  used to generate  $e_{\mathfrak{l}}(X)$  for encryption is of the form  $f = aX^\alpha + g(X) \in \mathbb{F}_q[X]$  with  $(a, \alpha) \in \mathbb{F}_q^* \times \mathbb{N}^n$ . It is shown in Levy-dit-Vehel and Perret (2004) that if  $X^\alpha$  is a term of total degree  $d$  and is a multiple of  $\prod_{i=1}^n X_i$ , and the terms of  $g(X)$  are of total degree strictly smaller than  $d - 3$ , then the scheme is resistant to the intelligent linear algebra attack, though not notwithstanding a *differential attack* (see next section).

## 4 Further Attacks

### 4.1 Basic CCA (Steinwandt and Geiselmann 2002)

Assume that instead of the ciphertext polynomial  $C = \sum_{i=1}^\ell p_i g_i + m$ ,  $m \in \mathbb{F}_q$ , an attacker sends  $\tilde{C}_i = \sum_{i=1}^\ell p_i g_i + X_i$ . Such a fake ciphertext cannot in principle be distinguished from a correct one (i.e both are considered valid). The plaintext corresponding to  $\tilde{C}_i$  is  $y_i$ , the  $i$ -th coordinate of the private key  $y \in \mathbb{F}_q^n$ . Thus, by  $n$  queries  $\tilde{C}_1, \dots, \tilde{C}_n$  to a decryption oracle, the attacker learns the entire private key  $y$ .

Note that, in a restricted context, this CCA is similar to Bulygin's attack. Suppose that  $y$  is the only zero of the ideal  $\mathbb{I}(F)$ . In this case, we know that its reduced Gröbner basis has the following form  $\{X_1 - y_1, \dots, X_n - y_n\}$ . In this setting:

$$\text{Can}(X_i, \mathbb{I}(F)) = y_i, \quad \text{for all } i, 1 \leq i \leq n.$$

Thus, the two attacks coincide in this particular context.

---

<sup>6</sup> $T$  and  $F$  being two non-zero field elements, it follows that  $y^\beta \neq 0$  for any choice of  $\beta$ .

## 4.2 Differential Attack

Hofheinz and Steinwandt propose in Hofheinz and Steinwandt (2002) a method to enhance the feasibility of the intelligent linear algebra attack previously described. In particular, their attack permits to recover “hidden monomials” in the Koblitz’s graph perfect code instance of Polly-Cracker (Koblitz 1998) (Chap. 5). We detail here the ideas of this attack. For  $f = \sum_{\mu \in \mathbb{N}^n} a_\mu X^\mu$ , we set:

$$\Delta(f) = \left\{ \frac{a_\mu}{a_v} X^{\mu-v} : X^v < X^\mu, a_\mu \cdot a_v \neq 0 \right\}.$$

Suppose that for some  $i$ ,  $1 \leq i \leq m$ , there exists a “characteristic difference”  $\delta_i$ , i.e.  $\delta_i = \frac{a_{\mu_i}}{a_{v_i}} X^{\mu_i - v_i}$ , with  $a_{\mu_i} X^{\mu_i}, a_{v_i} X^{v_i}$  monomials in  $g_i$  and such that:

$$\delta_i \in \Delta(g_i) \setminus \left( \bigcup_{j \neq i} \Delta(g_j) \right).$$

Suppose in addition that there exists a monomial  $a_{\eta_i} X^{\eta_i}$  in  $p_i$  such that  $X^{\eta_i} X^{\mu_i}$  and  $X^{\eta_i} X^{v_i}$  do not occur among the monomials of  $C - a_{\eta_i} X^{\eta_i} g_i$ . If for this characteristic difference, an adversary can find monomials  $m_1, m_2$  in the ciphertext with  $X^{\mu_i} | m_1$  and  $m_1/m_2$  being equal to  $\delta_i$ , then we can identify a potential monomial  $m_p$  of  $p_i$  as:

$$m_p = \frac{m_1}{a_{\mu_i} X^{\mu_i}} = \frac{m_2}{a_{v_i} X^{v_i}}.$$

The adversary cannot be sure about the correctness of his guess (i.e. if  $m_p$  is really a monomial of  $p_i$ ). But he can check it by computing the number of monomials in the simplified ciphertext  $C' = C - m_p g_i$ . Indeed, if the number of monomials in  $C'$  is smaller than in  $C$ , it is then very likely that  $m_p$  is a monomial of  $p_i$ . Notice that  $C$  and  $C'$  encrypt the same plaintext.

An adversary repeats this simplification process of the ciphertext for each characteristic difference in the set  $\Delta(g_i) \setminus (\bigcup_{j \neq i} \Delta(g_j))$  and for all  $i$ ,  $1 \leq i \leq \ell$ . If at some point of this simplification process  $C'$  is a constant, then the encrypted plaintext has been recovered successfully. Otherwise, he can try to perform an intelligent linear algebra attack on the simplified ciphertext. Subtracting a polynomial from the ciphertext can reveal hidden monomials. Indeed, the fact that a monomial  $m_{p_j}$  in  $p_j$  is hidden in the ciphertext  $C$  implies that for all  $i$ ,  $1 \leq i \leq \ell$  there exist two monomials  $m_{p_i}$  in  $p_i$  and  $m_{g_i}$  in  $g_i$  such that:

$$m_{p_j} g_j + \sum_{i=1}^{\ell} m_{p_i} m_{g_i} = 0.$$

Therefore, if one can find a monomial  $m_{p_i} \in \{m_{p_1}, \dots, m_{p_\ell}\}$ , then we know that the simplified polynomial  $C' = C - m_{p_i} g_i$  contains a monomial of the form  $m_{p_j} m_{g_j}$ ,  $m_{g_j}$  being a monomial of  $g_j$ . Therefore, the monomial which was hidden in the

ciphertext  $C$  is no longer hidden in the simplified ciphertext  $C'$ . We also would like to emphasize that it is not clear that the sets  $\{\Delta(g_i) \setminus (\bigcup_{j \neq i} \Delta(g_j))\}_{1 \leq i \leq \ell}$  always contain enough characteristic differences to recover all the hidden monomials.

Following these remarks, Levy-dit-Vehel and Perret (2004) propose an improvement of the differential attack. In particular, they no longer consider characteristic differences. Given a ciphertext  $C$ , they first compute (for a monomial  $m_i$  occurring in a decomposition of the form  $m_C = m_i m_{g_i}$ ) the polynomial  $C' = C - m_i g_i$ . This polynomial can validate the choice of the guess (we don't know if  $m_i$  is really a monomial of  $g_i$ ). Indeed, if  $\#\text{supp}(C') = \#\text{supp}(C) - \#\text{supp}(m_i g_i)$ , then this can be taken as evidence that  $m_i$  is a monomial of  $g_i$ . If this equality on the number of monomials does not hold, the polynomial  $C'$  can also be useful to reveal hidden monomials: if there exists a monomial  $m'_j$  in  $C'$  which is not a monomial of  $C$ , and which occurs in a decomposition of the form  $m_{C'} = m'_j m_{G_j}$ , for some monomial  $m_{g_j}$  of  $g_j$  (indeed, we then have  $m_{C'} = m'_j m_{g_j} = m_i m_{g_i}$ ) then, in addition to the fact that  $m_i$  is probably a monomial of  $g_i$ , it is also very likely that  $m'_j$  was a monomial of  $g_j$  that was hidden in the ciphertext  $C$ . In all other cases,  $m_i$  is not a monomial of  $g_i$ , and we then set  $C' = C$ .

At the second step, we select a monomial  $m_k \neq m_i$  in a decomposition of the form  $m_{C'} = m_k m_{g_k}$ , with  $m_{c'}$  a monomial of  $C'$  and for some monomial  $m_{g_k}$  of  $g_k$ . We compute  $C'' = C' - m_k g_k$  and we verify as previously whether  $m_k$  is a correct guess. We iterate this process while the simplified ciphertext is not a constant. Notice that even if there are hidden monomials in the ciphertext, it is very likely that these monomials can be guessed by considering simplified ciphertexts. As presented here, the attack of Hofheinz and Steinwandt (2002) and the improvement described above appear to be quite generic, and thus apply to the SAT based system too.

### 4.3 The 2-Nomial Attack

Endsuleit et al. (2002) suggests an “astonishingly simple” improvement of the trivial attack against the graph based CA systems consisting in computing the Gröbner basis of  $G$ , which takes strong advantage of the shape of the basis  $G$ . If  $n$  is a natural number, an  $n$ -nomial is a polynomial having at most  $n$  monomials. Hence a 2-nomial is either a monomial or a sum of 2 monomials. The public basis  $G$  of the graph-based CA-style cryptosystems discussed earlier are composed of degree 1 polynomials ( $G_1$ ) and 2-nomials ( $G_0$ ,  $G_2$  and  $G_3$ ). It is well known that Buchberger’s algorithm applied to a 2-nomial set returns a 2-nomial Gröbner basis and a very effective specialized version of Buchberger’s algorithm for 2-nomials exists. The 2-nomial attack consists in iteratively performing linear algebra on the degree 1 part of  $G$  with the aim of finding degree 1 2-nomial elements to add to  $G$ , and Buchberger’s algorithm on the 2-nomial part of  $G$  with the aim of finding new degree 1 elements.

#### 4.4 Further Linear Algebra Attacks

The differential attack and the 2-nomial attack can be combined and generalized in the following way. In the encoding procedure of a message, the set of monomials of all the polynomials used is quite limited, since the encoding procedure has to be performed in very short time. Hence all the computation takes place in a vector space spanned by a small set of monomials. If we can guess a small vector space containing this space, we can perform dense linear algebra in this space to reconstruct the message. Explicit relations between the generators simplify the linear algebra in this vector space.

Every operation performed consists in elementary steps, each one being the addition of a monomial multiple of an element of the public key, or possibly a simple combination of such polynomials. We call such polynomials, in the public key or easily derived from the public key, *elementary polynomials*. These elementary polynomials are usually of different categories:

- Monomials, that are usually clear from the problem description.
- Binomials involving just one variable, expressing the fact that a coordinate can take only some values in the ground field (e.g. the  $x_i^2 - x_i$  equations express the fact that coordinates have  $(0, 1)$  values)
- Other binomials (polynomials with two monomials).
- Polynomials with at least three monomials.

The first two types of polynomials allow the immediate reduction of the vector space under consideration: monomials can be removed from every polynomial under consideration, and binomials express an equivalence of a monomial with another, thus one of the two can be replaced with the other one, hence removed from every support.

We assume for a moment that the ideal generated by the 2-nomials is easy to study; ideally, we might be able to compute the Gröbner basis generated by the first three types of polynomials.

Because of the sparsity, every operation involving polynomials with at least three monomials can cancel usually just one monomial of the polynomial under construction, and sometimes can cancel more than one, but usually leaves more than one monomial: one or more monomials of the current polynomial are erased, but at least two are added. If this is true, one can find the support of the computation as follows:

- Start with the support of the cryptogram;
- Add to it the support of the monomial multiples of the elementary polynomials whose support is partly contained in the current support.
- Attempt a linear algebra decryption. If it fails, repeat with the enlarged support.

This might fail only if the binomial part of the ideal cannot be explicitly described. In particular, it will fail for hard toric ideals. The fact that toric ideals can model hard combinatorial problems, like integer programming (Conti and Traverso 1992; Bigatti et al. 1999) might give some faint hope of using ideals generated by binomials for a variant of Polly Cracker (Caboara et al. 2008).

## 5 Polly-Two

Recently (Ly 2006) a cryptosystem has been proposed, that fits our definition of Polly Cracker, but that has a different specification, that apparently makes some attacks more difficult.

The setting is the following: let  $\mathbb{F}$  be a finite field,  $m > n$  non-negative integers,  $\phi, \psi$  maps  $\mathbb{F}[y_1, \dots, y_m] \xrightarrow{\phi} \mathbb{F}[x_1, \dots, x_n] \xrightarrow{\psi} \mathbb{F}$ , and  $L \subseteq \mathbb{F}[Y]$  an ideal contained in  $\ker(\psi \circ \phi)$  given through a set of generators  $f_1, \dots, f_n$ . The public ideal is  $I = \ker(\phi) + L$  (specified by giving  $\phi$  and  $L$ ) and the private ideal  $J = \ker(\psi \circ \phi)$  is the private key. A technical condition is that  $\psi(\phi(y_i)) \neq 0$ .

Remark that  $\psi$  is nothing else than a point  $P \in \mathbb{F}^n$ ,  $\phi$  is a vector  $G = (g_1, \dots, g_m) \in \mathbb{F}[X]^m$ , the condition on  $L$  is that  $P$  is a root of  $I$ . The condition  $\psi(\phi(y_i)) \neq 0$  means that no  $g_i(P)$  vanishes, and this is necessary for decoding the cryptograms.

The  $f_i$  are chosen as polynomials with few monomials of high degree, and  $\phi$  is defined through  $g_i = \phi(y_i)$  of low degree, such that a Gröbner basis of  $\ker(\phi)$  can be easily computed. This combination should ensure that a Gröbner basis of  $\phi(L)$ , or anyway a root of  $L + \ker(\phi)$  is hard to compute.

A message is an element  $c \in \mathbb{F}$ , and a cryptogram is a pair  $(p = p' + my^\alpha, \alpha)$ ,  $p' = p_1 + p_2$ ,  $p_1 \in L$ ,  $p_2 \in \ker(\phi)$ . The choice of  $p_1$  and  $p_2$  is made in such a way that the support of  $p_1$  does not meet the support of  $p_1 + p_2$ , and  $y^\alpha$  is chosen in the support of  $p_1 + p_2$ . This should make intelligent linear algebra impossible.

The standard attacks are difficult, since the public key is not a list of polynomials, but a recipe to produce them, and many different variants can be used. There are however other attacks that can be used.

The first attack (Steinwandt 2006) works as follows. The encoding method computes first  $p_1$ , then every monomial of  $p_1$  is erased by a monomial of one element of  $\ker(\phi)$ . The other monomials usually do not cancel, and one can usually partition the support in “clouds” of monomials, recognizable by their GCD. Each cloud  $c$  is an element of  $\ker(\phi)$  with one monomial removed (and possibly another one modified); hence  $\phi(c)$  behaves like a monomial (the opposite of the missing monomial) and it can be easily identified.

The second attack is more complex, but independent of any concealing strategy that could be designed to mix the “clouds” making them unidentifiable. One cannot compute in  $\phi(I)$  if one uses a representation of polynomials as sparse multivariate polynomials. This would mean replacing, in the polynomials  $f_i$  defining  $L$ , the  $y_i$  with  $f(y_i)$ , i.e. variables with polynomials in monomials of very large degree, hence  $\phi(f_i)$  would be a polynomial with thousands of monomials. But one can instead use a different representation. Evaluating  $\phi(p(y_1, \dots, y_m)) = p(\phi(y_1), \dots, \phi(y_m))$  at a point of  $Q \in \mathbb{F}^n$  is easy, it is just the evaluation of the monomials of  $p$  in the  $\phi(y_i)$ , i.e. compute the value  $v_i \in \mathbb{F}$  at  $Q$  of each  $\phi(y_i)$  and replace  $y_i$  in  $p$  with  $v_i$ ; this is the evaluation of a very sparse polynomial, and can be made by simply evaluating the monomials as power products and summing them. This shows that working on polynomials as black boxes or straight line programs is easy. Since  $\phi(p_2) = 0$ , we have  $\phi(p) = \phi(p_1 + my^\alpha)$ , that has a small number of monomials, and sparse

interpolation can recover it. Since  $y^\alpha$  by assumption is not in the support of  $p_1$ , recovering  $m$  is immediate.

In both attacks the key ingredient is discovering a polynomial with few monomials through its value at some points. This is known as *Sparse interpolation*. Algorithms of the type of Ben-Or and Tiwari (1988) may be used to solve efficiently the problem. See also Grigoriev et al. (1990), Kaltofen and Trager (1990).

## 6 Non-commutative Gröbner Cryptosystems? No Thanks!

Another urban legend to dispel is that “Gröbner bases are impossible to compute being infinite”.

Polly Cracker cryptosystems can obviously be proposed on structures different from polynomial rings and their ideals, provided that a concept of Gröbner basis exists. So, for example, free  $\mathcal{P}$ -modules and their submodules can be used instead of polynomials and ideals. This is of little interest, since this can be modeled with suitable polynomial rings and orderings, see Caboara and Silvestri (1999).

Straightforward extensions can be made with other structures: reduction rings, subalgebras, non-commutative algebras, etc; everything is extended naturally, with increased difficulties for the legitimate users, because of the difficulties of the generalization of Gröbner bases, but with unchanged power of the attacks, since linear algebra always remains the same.

Some of these extensions have been proposed; we will first analyze the weaknesses that they share with the other Polly Cracker systems, and their own weaknesses, derived from the difficulty of computing a private key, the Gröbner basis being often infinite; then we will show, through Pritchard’s (1996) algorithm that these difficulties do not extend to the simpler attacks.

### 6.1 Non-commutative Polly Cracker

Rai (2004) proposes to use two-sided ideals in free non-commutative algebras for a generalization of Barkee’s cryptosystem. The theory of Gröbner bases and Buchberger’s algorithm generalizes to two-sided ideals in non-commutative polynomial rings, (Green et al. 1998; Reinert 2003) but the Gröbner bases are usually infinite. This fact is quoted as a factor of security in the obvious generalization of Barkee’s Polly Cracker to non-commutative polynomials. While it is true, as pointed in Rai (2004), that Ufnarowski’s Ideal  $(xx - xy) \in k\langle x, y \rangle$  has an infinite Gröbner basis, Rai (2004) carefully avoids to remark that such infinite basis is  $\{g_i, i \in \mathbb{N}\}$  where  $g_i := xy^i x - xy^{i+1}$  and that ‘proving’ this statement simply requires (Green et al. 1998) to verify that the S-polynomials among  $g_i$  and  $g_j$

$$S(g_i, g_j) = xy^i g_j - g_i y^j x = xy^{i+j+1} x - xy^i xy^{j+1}$$

has the Gröbner representation  $S(g_i, g_j) = -g_i y^{j+1} + g_{i+j+1}$ . Of course, all the attacks to Polly Cracker schemes that do not rely on the knowledge of a Gröbner basis remain unchanged. But the fact that the Gröbner basis of an ideal is infinite does not mean that it is not computable; indeed, the system BERGMAN (Backelin et al. 2005) is able to compute such infinite Gröbner bases representing *finite state automata* (Cojocaru and Ufnarowski 1995). Moreover, even without this, to solve any normal form problem, a partial Gröbner basis is sufficient.

Coming to explicit examples, the fact that Gröbner bases of non-commutative polynomials are usually infinite is instead a serious obstacle to the construction of explicit Polly Cracker schemes. In particular, the only explicit systems that have been proposed are systems in which the private ideal has one generator (Gröbner bases of principal non-commutative ideals can be infinite, but it is easy to compute, describe and show that are Gröbner bases).

### 6.1.1 Factoring Attacks

In this part, we will discuss of some ideas that might allow to attack such a scheme. Principal ideals however allow easy recovery of the private key from the public key through a factoring attack. If  $f$  is the generator of the private ideal  $J$ , then the polynomials of the public key are two-sided multiples of  $f$ ,  $g_i = h_i f \ell_i$ . It is hence easy to recover  $f$  through a factorization, via one of the following methods:

- Map the non-commutative polynomials  $g_i$  to the corresponding commutative polynomial ring, compute the GCD, factor it and lift to the non-commutative ring. This procedure might not work if all the  $g_i$  reduce to 0 in the commutative polynomial ring, but this is not the case in the examples described in Rai (2004).
- If the  $g_i$  are 0 when made commutative, find maps to skew-commutative polynomial rings such that the  $g_i$  do not vanish when mapped, and factor the images through a “brute force” approach using linear algebra, then lift to the non-commutative polynomials.

Direct factorization of non-commutative polynomials: algorithms for this factorization exist (Davenport 1991), and, although it is not unique (and in general quite expensive), the fact that in this case one seeks a common divisor of a set of polynomials can be exploited in the algorithm.

## 6.2 Monoid Algebras

In Ackermann and Kreuzer (2006) Barkee’s cryptosystem in monoid algebras are discussed. The basis is the extension of Gröbner bases to monoid rings described in Madlener and Reinert (1993), Reinert (1995). It is interesting that any cryptosystem can be described in terms of this extension of Polly Cracker, since every map on a set can be extended to a free monoid generated by the same set (although the

complexity increases considerably), in particular this is shown for RSA and Diffie-Hellmann. The proposal however does not include concrete cryptosystems, except Polly2 and Rai systems, that can be seen as special cases of monoid algebra Polly Cracker, hence can be safely ignored.

### 6.3 Pritchard's Decryption Algorithm

Generalizing Gröbner bases to non-commutative settings is simple, but lacking a generalization of Noetherianity, they may be infinite. The status of such infinite non-commutative Gröbner bases is not so obvious; the existence of such infinite bases implies that Buchberger's algorithm becomes a semi-decision procedure which terminates returning a finite Gröbner basis if and only if such basis is finite.

More relevant, Pritchard (1996) adapted such version of Buchberger's algorithm into a semi-decision procedure which, given a basis  $G \subset k\langle X_1, \dots, X_n \rangle$  and a polynomial  $f \in k\langle X_1, \dots, X_n \rangle$  terminates if and only if  $f \in \mathbb{I}(G)$ .

It is a trivial exercise to adapt Pritchard's Procedure in order to produce a procedure which, given a basis  $G \subset \mathbb{F}\langle X_1, \dots, X_n \rangle$ , a polynomial  $C \in \mathbb{F}\langle X_1, \dots, X_n \rangle$  and a finite set of terms

$$T \subset \mathbf{N}(\mathbb{I}(F)) \subset \langle X_1, \dots, X_n \rangle,$$

terminates if and only if  $M := \text{Can}(C, \mathbb{I}(G)) \subset \text{Span}_{\mathbb{F}}(T)$ , in which case it returns such a canonical form, thus reading the message  $M := \text{Can}(C, \mathbb{I}(F))$  encrypted as  $C$ .

There are today two wide generalizations of the notion of non-commutative Gröbner bases which cover all specific instances discussed in literature (in particular Gröbner bases in the free monoid ring (Rai 2004) and over monoid rings (Ackermann and Kreuzer 2006)): Reinert's *function rings* (Reinert 2003) and Apel's *pseudovaluation rings* (Apel 2000). It has been proved (Mora 2009b):

**Theorem 1** *Let  $\mathcal{R} := \mathbb{F}[T]$  be a Reinert function ring,  $F \subset \mathcal{R}$  a finite set,  $f \in \mathcal{R}$  an element,  $T \subset \mathbf{N}(\mathbb{I}(F)) \subset \mathcal{T}$  a finite set of terms; there is a semi-decision procedure which terminates if and only if there exists  $g \in \text{Span}_{\mathbb{F}}(T)$  such that  $f - g \in \mathbb{I}(F)$  and which, in case of termination, returns such a  $g$ .*

We moreover conjecture that the same statement holds even if  $\mathcal{R}$  is just a pseudovaluation ring. Therefore, on the basis of these results, we consider that a claim of solidity on the basis of the (false) urban legend that “non-commutative Gröbner bases are impossible to be computed being infinite” is not acceptable: it must be up to the claimers to prove that their instantiation of a Polly-Cracker system is solid against a Gröbner basis computing, convincingly showing that a mock-version of the proposed scheme resists against a *good* non-commutative implementation of Pritchard's Procedure.

## 7 Conclusion

From a cryptanalytic point of view, the study of Polly Cracker-type systems gives rise to interesting mathematical and algorithmic problems. But the structural—as well as oracle—attacks that we have presented show evidence that those schemes are not suited for the design of secure cryptosystems. Moreover, they suffer from efficiency problems, namely a bad encryption rate and an often large public key size. Therefore, we do not think they deserve further investigation with respect to design concerns. However, very recently, another Polly Cracker system based on binomial ideals has been proposed (Caboara et al. 2008). Considering the fact that those ideals are very special inside Gröbner bases theory, it might be the case that this late proposal is more secure than its predecessors. Thus, the hope in secure Polly Cracker design might now lie in here.

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

The authors want to thank Moriarty, that has not only inspired but actually cooperated to the preparation of the paper. But, because of his never-ending struggle with his archi-enemy, eventually declined to appear as author.

## References

- P. Ackermann and M. Kreuzer, *Gröbner basis cryptosystems*, AAECC **17** (2006), nos. 3–4, 173–194.
- M. E. Alonso and M. G. Marinari, *Oracle-supported drawing of the Gröbner escalier*, preprint, 2008.
- J. Apel, *Computational ideal theory in finitely generated extension rings*, Theoret. Comput. Sci. **244** (2000), nos. 1–2, 1–33.
- J. Backelin, S. Cojocaru, and V. Ufnarowski, *Mathematical computations using Bergman*, 2005, Lund University, Sweden, 2005. – 206 p.
- W. Banks, D. Lieman, and I. Shparlinski, *Cryptographic applications of sparse polynomials over finite rings*, Proc. of ICISC 2000, LNCS, vol. **2015**, Springer, Berlin, 2001, pp. 206–220.
- F. Bao, R. H. Deng, W. Geiselmann, G. Schnorr, Steinwand R., and H. Wu, *Cryptanalysis of two sparse polynomial based public key cryptosystems*, Proc. of PKC 2001, LNCS, vol. **1992**, Springer, Berlin, 2001, pp. 153–164.
- B. Barkee, D. C. Can, J. Moriarty, and R. F. Ree, *Why you cannot even hope to use Gröbner bases in public key cryptography: an open letter to a scientist who failed and a challenge to those who have not yet failed*, J. Symbolic Comput. **18** (1994), no. 6, 497–501.
- M. Ben-Or and P. Tiwari, *A deterministic algorithm for sparse multivariate polynomial interpolation*, Proc. of ACM Symp. Theory Comput., ACM, New York, 1988, pp. 301–309.
- A. M. Bigatti, R. La Scala, and L. Robbiano, *Computing toric ideals*, J. Symbolic Comput. **27** (1999), no. 4, 351–365.
- O. Billet and J. Ding, *Overview of cryptanalysis techniques in multivariate public key cryptography*, this volume, 2009, pp. 263–283.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.

- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- S. Bulygin, *Chosen-ciphertext attack on noncommutative Polly Cracker*, 2005, <http://arxiv.org/abs/cs/0508015v2+>.
- S. Bulygin and T. S. Rai, *Countering chosen-ciphertext attacks against noncommutative Polly Cracker cryptosystems*, 2006, talk at Special Semester on Gröbner Bases, Linz, Austria.
- M. Caboara and M. Silvestri, *Classification of compatible module orderings*, J. Pure Appl. Algebra **142** (1999), no. 1, 13–24.
- M. Caboara, F. Caruso, and C. Traverso, *Gröbner bases in public key cryptography*, Proc. of ISSAC 2008, to appear, 2008.
- F. Caruso, P. Conti, and C. Traverso, *Non-commutative factorisation and GCD with applications to public-key cryptography*, 2008, Proc. of Differential Algebra and Related Computer Algebra, Le Matematiche, LXIII (1), pp. 37–39.
- S. Cojocaru and V. Ufnarovsky, *Noncommutative Gröbner basis, Hilbert series, Anick's resolution and BERGMAN under MS-DOS*, Computer Science Journal of Moldova **3** (1995), 24–39.
- P. Conti and C. Traverso, *Buchberger's algorithm and integer programming*, Proc. of AAECC, LNCS, vol. **539**, Springer, Berlin, 1992, pp. 130–139.
- P. Conti and C. Traverso, *Homomorphism attacks to non-commutative Polly Cracker*, 2007, preprint.
- J. H. Davenport, *Factorisation of polynomials in non-commuting variables*, 1991, Personal communication.
- A. Dickenstein, N. Fitchas, M. Giusti, and C. Sessa, *The membership problem for unmixed polynomial ideals is solvable in single exponential time*, Discrete Appl. Math. **33** (1991), nos. 1–3, 73–94.
- R. Endsuleit, W. Geiselmann, and R. Steinwandt, *Attacking a polynomial-based cryptosystem: Polly Cracker*, Int. J. Inf. Secur. **1** (2002), no. 3, 143–148.
- M. Fellows and N. Koblitz, *Combinatorial cryptosystems galore!*, Contemp. Math. **168** (1994), 51–61.
- D. Grant, K. Krastev, D. Lieman, and I. Shparlinski, *A public key cryptosystem based on sparse polynomials*, Proc. of ICCC 1998, Springer, Berlin, 2000, pp. 114–121.
- E. Green, T. Mora, and V. Ufnarovsky, *The non-commutative Gröbner freaks*, Symbolic rewriting techniques, Progr. Comput. Sci. Appl. Logic, vol. **15**, Birkhäuser, Basel, 1998, pp. 93–104.
- D. Y. Grigor'ev, M. Karpinski, and M. F. Singer, *Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields*, SIAM J. Comput. **19** (1990), no. 6, 1059–1063.
- E. A. Hirsch, <http://logic.pdmi.ras.ru/~hirsch/sat.html>, 2009.
- D. Hofheinz and R. Steinwandt, *A “differential” attack on Polly Cracker*, Proc. of ISIT 2002, 2002, pp. 211–211.
- E. Kaltofen and B. M. Trager, *Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators*, J. Symbolic Comput. **9** (1990), no. 3, 301–320.
- N. Koblitz, *Algebraic aspects of cryptography*, Algorithms and Computation in Mathematics, vol. **3**, Springer, Berlin, 1998.
- F. Levy-dit-Vehel and L. Perret, *A Polly Cracker system based on satisfiability*, Coding, cryptography and combinatorics, Progr. Comput. Sci. Appl. Logic, vol. **23**, Birkhäuser, Basel, 2004, pp. 177–192.
- L. V. Ly, *Polly Two: a new algebraic polynomial-based public-key scheme*, AAECC **17** (2006), nos. 3–4, 267–283.
- K. Madlener and B. Reinert, *Computing Gröbner bases in monoid and group rings*, Proc. of ISSAC 1993, ACM, New York, 1993, pp. 254–263.

- T. Matsumoto and H. Imai, *Algebraic methods for constructing asymmetric cryptosystems*, Proc. of AAECC, LNCS, vol. **229**, Springer, Berlin, 1985, pp. 108–119.
- T. Mora, A 15/01/94 communication to M.R. Fellows and N. Koblitz, 1994
- F. Mora, *De nugis Groebnerialium. II. Applying Macaulay's trick in order to easily write a Gröbner basis*, AAECC **13** (2003), no. 6, 437–446.
- T. Mora, *Solving polynomial equation systems. II. Macaulay's paradigm and Gröbner technology*, Encyclopedia of Mathematics and its Applications, vol. **99**, Cambridge University Press, Cambridge, 2005.
- T. Mora, *Gröbner technology*, this volume, 2009a, pp. 11–25.
- T. Mora, *Solving polynomial equation systems. III, algebraic solving and beyond*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2009b, to appear.
- F. L. Pritchard, *The ideal membership problem in non-commutative polynomial rings*, J. Symbolic Comput. **22** (1996), no. 1, 27–48.
- T. S. Rai, *Infinite Gröbner bases and noncommutative Polly Cracker cryptosystems*, Ph.D. thesis, Virginia Polytech. Inst. and State Univ., 2004.
- B. Reinert, *On Gröbner bases in monoid and group rings*, Ph.D. thesis, Kaiserslautern, 1995.
- B. Reinert, *A systematic study of Gröbner basis methods*, Ph.D. thesis, Kaiserslautern, 2003, Habilitationschrift.
- R. Steinwandt, *A ciphertext-only attack on Polly Two*, 2006, preprint.
- R. Steinwandt and W. Geiselmann, *Cryptanalysis of Polly Cracker*, IEEE Trans. on Inf. Th. **48** (2002), no. 11, 2990–2991.

# Block Ciphers: Algebraic Cryptanalysis and Gröbner Bases

Carlos Cid and Ralf-Philipp Weinmann

**Abstract** Block ciphers are one of the most important classes of cryptographic algorithms in current use. Commonly used to provide confidentiality for transmission and storage of information, they encrypt and decrypt blocks of data according to a secret key. Several recently proposed block ciphers (in particular the AES (Daemen and Rijmen in *The Design of Rijndael*, Springer, Berlin, 2002)) exhibit a highly algebraic structure: their round transformations are based on simple algebraic operations over a finite field of characteristic 2. This has caused an increasing amount of cryptanalytic attention to be directed to the algebraic properties of these ciphers. Of particular interest is the proposal of the so-called *algebraic attacks* against block ciphers. In these attacks, a cryptanalyst describes the encryption operation as a large set of multivariate polynomial equations, which—once solved—can be used to recover the secret key. Thus the difficulty of solving these systems of equations is directly related to the cipher’s security. As a result computational algebra is becoming an important tool for the cryptanalysis of block ciphers. In this paper we give an overview of block ciphers design and recall some of the work that has been developed in the area of algebraic cryptanalysis. We also consider a few computational and algebraic techniques that could be used in the analysis of block ciphers and discuss possible directions for future work.

## 1 Introduction

Confidentiality is the traditional goal of cryptography. For centuries encryption algorithms have been used by parties wishing to communicate securely. The traditional<sup>1</sup> way to use an encryption algorithm is to share a secret key, which is used together with the message as input to the algorithm. Accordingly, these algorithms are called *symmetric-key encryption algorithms*. The two main types of symmetric-key encryption algorithms are block ciphers and stream ciphers (Menezes et al. 1997).

---

<sup>1</sup>See Billet and Ding (2009) for public key cryptography.

C. Cid

Information Security Group, Royal Holloway, University of London, London, UK  
e-mail: [carlos.cid@rhul.ac.uk](mailto:carlos.cid@rhul.ac.uk)

R.-P. Weinmann

University of Luxembourg, Luxembourg City, Luxembourg  
e-mail: [weinmann@cdc.informatik.tu-darmstadt.de](mailto:weinmann@cdc.informatik.tu-darmstadt.de)

With block ciphers, data is broken up into blocks of fixed length, and each block is encrypted according to the secret key. With stream ciphers, individual elements of the message alphabet (e.g. characters or bits) are individually encrypted, using an encryption transformation that varies with time. In practice, most stream ciphers produce a *keystream*, and encryption is achieved by XORing the keystream with the message (Armknecht and Ars 2009).

While in cryptography one studies the design and construction of algorithms to secure the transmission and storage of information, cryptanalysis focuses on *breaking* cryptographic algorithms. Block cipher design and analysis have been very active areas of research since the 1970's, culminating in the selection of Rijndael (Daemen and Rijmen 2002) as the new Advanced Encryption Standard (AES) in 2000. The AES represents the state of art in block cipher design and provides an unparalleled level of assurance against most standard cryptanalytic techniques, such as differential and linear cryptanalysis. However the AES, as well as some related block ciphers, exhibit a highly algebraic structure. This has motivated researchers to investigate whether algebraic properties of ciphers can be exploited in their cryptanalysis. Of particular interest is the proposal of so-called *algebraic attacks* against block ciphers. The main idea behind an algebraic attack is to write the encryption operation as a large system of low degree, often sparse, multivariate polynomial equations. Solving this system leads to a recovery of the secret key. This is a reasonably recent area for symmetric-key cryptanalysis, where recognising statistical patterns of bits has usually been the most effective form of cryptanalysis, and potentially opens the possibility of applying well-known techniques from computational algebra, such as Gröbner basis algorithms, in symmetric-key cryptanalysis. In this paper we further explore this subject.

Section 2 gives a brief overview of block cipher design principles. Section 3 reviews some scenarios and techniques in conventional cryptanalysis of block ciphers. In Sect. 4 we discuss the principles of algebraic cryptanalysis. Section 5 gives an overview of experimental ciphers that have been proposed to investigate algebraic attacks in practice. Section 6 provides a summary of experimental results obtained for these ciphers. In Sect. 7 we give examples of attack strategies that make use of the structure of the systems in the attempt to reduce the complexity of computations. Finally, in Sect. 8 we give pointers to alternative approaches that have recently been proposed for solving the systems of equations that appear in the context of block ciphers. Section 9 concludes this paper.

## 2 Design of Block Ciphers

Mathematically, a block cipher can be described as a function

$$\mathcal{E} : K \times P \rightarrow C$$

$$(k, p) \mapsto c,$$

such that the mapping  $\mathcal{E}(k, \cdot) : P \rightarrow C$  is invertible. We say that  $p \in P$  is the plaintext (message),  $k \in K$  is the secret key, and  $c = \mathcal{E}(k, p) \in C$  is the resulting ciphertext (encrypted message). The sets  $P$ ,  $C$  and  $K$  are called the *plaintext space*, the *ciphertext space* and the *key space*, respectively. In practice, for modern ciphers we usually have  $C = P$  and  $2^{64} \leq |P|, |K| \leq 2^{256}$ . The best known examples of block ciphers are the Data Encryption Standard (DES) (National Bureau of Standards 1977) and its successor, the Advanced Encryption Standard (AES) (National Institute of Standards and Technology 2001).

When  $C = P$ , we can also consider a block cipher as an indexed set of permutations

$$\begin{aligned}\mathcal{E} : K &\rightarrow \text{Sym}(P) \\ k &\mapsto \varepsilon_k,\end{aligned}$$

where  $\varepsilon_k(p) = \mathcal{E}(k, p) = c$ . The theoretically ideal way to define a block cipher would be to construct such a set by randomly selecting a permutation from  $\text{Sym}(P)$  for every key  $k \in K$ . This process is of course impractical; instead, block cipher designers attempt to construct algorithms that mimic the behaviour of random permutations.

The roots of modern block cipher design can be traced to Claude Shannon. In his seminal article (Shannon 1949), Shannon discusses the design of block ciphers based on simple operations, which are iterated a number of times to achieve the desired security. With this goal in mind, he introduces the concepts of *diffusion* and *confusion*. The aim of diffusion is to spread the influence of all parts of the inputs of a block cipher, namely the plaintext and the key, to all parts of the output, the ciphertext. Diffusion is often provided by the use of (bit or byte) permutations or linear transformations. The aim of confusion is to make the relationship between plaintext, ciphertext, and key complicated. In most ciphers, confusion is provided by carefully chosen *substitution* or *S-boxes*. These make local substitutions of small sub-blocks of data which are then spread by the diffusion transformations (Cid et al. 2007).

A number of design structures have been proposed to implement the ideas of Shannon in practice. Most modern ciphers are designed as *substitution-permutation (SP-)networks* (Shannon 1949). In SP-networks, invertible layers of carefully chosen non-linear substitution (providing confusion) and linear/affine transformations (providing diffusion) are iterated a number of times. To make the encryption operation key dependent, secret key material is introduced at every iteration

Notably, the AES employs this design, with the basic version AES-128 employing 10 encryption rounds.

Another very common design is the *Feistel network* (Feistel 1973). In its basic form, Feistel ciphers modify only one half of the cipher state in each round. Although these ciphers restrict the speed of diffusion compared to SP-network ciphers, this particular structure allows the designer to select a transformation  $F$  that does not need to be a permutation. The most well-known example of a Feistel cipher is the DES (National Bureau of Standards 1977).

Besides its general structure, the security of a block cipher is obviously highly dependent on the properties of the different components used. Usually substitution layers consist of a number of highly non-linear Boolean functions. A particularly popular choice are transformations based on power functions over a finite field. For example, the AES S-Box is based on the inversion over a field of order  $2^8$  (Daemen and Rijmen 2002). For a discussion of the properties Boolean functions used in block ciphers, see Carlet (2009).

For the diffusion layer, linear (or affine) transformations are used in the attempt to maximise the level of mixing within each round. The AES diffusion layer has been designed in accordance to the *wide trail strategy* (Daemen and Rijmen 2002). This provides fast diffusion and, as a consequence, ensures the security of the AES against common cryptanalytic methods, such as differential and linear cryptanalysis (Daemen and Rijmen 2002).

### 3 Block Cipher Cryptanalysis

Cryptanalysis focuses on *breaking* ciphers. The exact notion of a “break” can however vary depending on the context. Traditionally, the goal of a cryptanalyst has been to recover an encrypted message. A much more ambitious attack is to recover the encryption key. Likewise, the adversary’s capabilities can vary. It may have access to only a single ciphertext, or to a large number of plaintext/ciphertext pairs. In either situation, one should always assume an adversary to have full knowledge of the algorithm details; the only secret piece of information is the encryption key<sup>2</sup>. Below we present an overview of the most common cryptanalytic attack models (adapted from Biryukov 1999). We provide a taxonomy of cryptanalytic attacks, ranging from the most practical to the most hypothetical.

- *ciphertext-only*: The adversary only has access to encrypted messages (and some information about the distribution of plaintexts). A cipher susceptible to ciphertext-only attacks cannot protect against passive eavesdropping. Although many of the classical ciphers could be broken by ciphertext-only attacks, this is a very unlikely scenario for modern block ciphers.
- *known plaintext*: This is a common form of attack, in which we assume the adversary has full or partial knowledge of messages being encrypted and of the corresponding ciphertexts. This is a very realistic scenario, where an adversary could assume the use of common words in the plaintext or exploit the fact that messages often have a lot of redundancy (e.g. common headers, etc).
- *chosen plaintext or ciphertext*: In the *chosen plaintext* scenario, the adversary can choose messages to be encrypted and sees the corresponding ciphertext. This is a less common scenario, although still quite realistic. A modern cipher is only

---

<sup>2</sup>This is known as *Kerckhoffs’ principle*, named after 19th century cryptographer Auguste Kerckhoffs, who stated that a cryptosystem should remain secure if all the details of the system, except the key, are of public knowledge (Kerckhoffs 1883a, 1883b).

considered secure if it can withstand chosen plaintext attacks. Likewise, in *chosen ciphertext* attacks, the adversary can select the ciphertext to be *decrypted* and has access to the corresponding plaintext. This also is a realistic scenario, especially with the growing number of network applications that automatically decrypt received messages.

- *adaptive chosen plaintext or ciphertext*: In these variants of the previous two attack scenarios, the adversary is able to adapt its choices of texts to be encrypted and decrypted, based on information learned during the attack. Although much less common, this is also a reasonable realistic scenario.
- *related key*: In this case, the adversary exploits the fact that different keys used fulfill a known relation (e.g. they may differ on a certain number of bits). This is usually employed in conjunction with some of the scenarios above.

By working on one of the modes of attack described above, the overall goal of a cryptanalyst could be classified in the following way.

- *distinguishing*: Modern block ciphers are designed to model a random permutation. If it is possible to design an algorithm that can efficiently distinguish an instantiation of the cipher with a fixed but unknown key from a random permutation, we say that the cipher is susceptible to distinguishing attacks. This is the most basic type of attack, and often indicates some structural weaknesses in the cipher. It may however allow more sophisticated forms of attacks.
- *decryption*: In this attack the adversary algorithmically obtains full (or partial) knowledge about the plaintext of an encrypted message. This is achieved without knowledge of the secret key, and thus it may not compromise other messages encrypted under the same key.
- *encryption*: This is also known as a forgery attack, where the adversary is able to successfully encrypt a message without knowledge of the secret key. Ciphers vulnerable to this type of attack are not suitable to authentication purposes.
- *partial key recovery*: In this attack the adversary is able to learn some information about the secret key (e.g. a number of key bits). The existence of an efficient partial key recovery attack is very undesirable and indicates some structural weaknesses in the cipher. In practical terms, the adversary could then simply try to guess the remaining bits (such that the overall attack complexity of the attack is still lower than full exhaustive search). More commonly, such an attack could be used as a first step in an elaborate full key recovery attack.
- *full key recovery*: This is the most ambitious and devastating attack. The adversary is able to fully recover the secret key, and can therefore decrypt all messages (past and future) encrypted under the key.

A cipher's security against the types of attacks described above is ultimately measured by the complexity of mounting such attack in practice. In a known plaintext scenario, an adversary could always mount a key recovery attack against the cipher by decrypting a small number of ciphertexts using all the possible keys. If the size of the key space  $K$  is  $2^n$ , the adversary is expected to succeed on average after  $2^{n-1}$  decryption operations. This type of attack is called *exhaustive key search*.

Modern ciphers are designed with  $n$  large enough so that exhaustive key search is infeasible in practice (e.g.  $n = 128$ ). Beside time complexity (i.e. the number of operations required), the complexity of attacks can also be measured by the number of plaintext/ciphertext pairs or memory required (data complexity). In such scenarios, adversaries are given (sometimes unrealistic) extensive capabilities, in the attempt to show some weakness in the design.

Modern ciphers are expected to withstand most of these attack scenarios; ultimately the goal of a block cipher designer is that exhaustive key search is in practice the most efficient attack against the cipher.

## 4 Algebraic Cryptanalysis

In contrast to conventional block cipher cryptanalysis, algebraic cryptanalysis exploits the intrinsic algebraic structure of a cipher. In its most common form, the attacker expresses the encryption transformation as a large set of multivariate polynomial equations, and subsequently attempts to solve the system to recover information about the encryption key. Algebraic attacks represent an exciting new development in cryptology, as they open new perspectives in block cipher cryptanalysis. For example, only a handful of plaintext–ciphertext pairs is usually required in algebraic cryptanalysis. Furthermore, it is expected that if an algebraic attack proves to be successful against a particular cipher, it might not be easily avoided by simply increasing the number of rounds.

Although different forms of algebraic methods used in cryptology may be considered as algebraic cryptanalysis, we will largely restrict ourselves to attacks in which (systems of) polynomial equations arising from the cipher are solved. Specifically, we will deal with attacks that make use of computational algebra techniques, such as Gröbner basis algorithms (Buchberger 1965, 1985, 2006), to compute solutions of polynomial systems.

The first documented use of Gröbner bases in symmetric-key cryptography was in fact not in a direct attack on a block cipher, but rather used for an improvement of the linear cryptanalysis of DES (Shimoyama and Kaneko 1998).

In 2002 Courtois and Pieprzyk proposed an algebraic attack on the block ciphers Rijndael (AES) and Serpent by solving large systems of quadratic polynomial equations with a method of their own invention (Courtois and Pieprzyk 2002a, 2002b): Extended Sparse Linearization (XSL). This method is based on the XL algorithm (Courtois et al. 2000) and attempts to exploit the structure and sparsity of the polynomial system. The XL algorithm in turn has been later shown to be a degenerated version of the  $F_4$  algorithm (Ars et al. 2004). These works gave rise to much speculation on the potential for algebraic attacks against block ciphers (in particular the AES). Cid and Leurent showed in 2005 that the XSL method as proposed in Courtois and Pieprzyk (2002b) does not work and a natural modification to fix the algorithm will turn it into an equivalent of XL (Cid and Leurent 2005). Furthermore, Lim and Khoo have shown that the XSL version proposed in Courtois and Pieprzyk (2002a) has a much higher complexity than expected and raised questions on whether the algorithm can work at all (Lim and Khoo 2007).

Although the XSL method itself is now widely recognized to be incorrect, its publication can be considered as a key event that fueled the interest of the cryptographic community in algebraic methods in symmetric-key cryptology. This led other researchers to the investigation of computational algebra techniques, in particular Gröbner bases, as a cryptanalytic tool (Ars 2005; Buchmann et al. 2006b; Cid et al. 2005b; Faugère 2007).

## 4.1 Polynomial Descriptions of Block Ciphers

In theory, most block ciphers afford a polynomial representation of the encryption process: by representing the encryption function as a vector of high-degree polynomials. The evaluation of such a vector with a fixed plaintext and key then yields the ciphertext.

In this case, the structure (i.e. the terms occurring) of the polynomials is known; merely the coefficients are unknown. This leads to interpolation attacks (Jakobsen and Knudsen 1997). This is a method for the cryptanalysis of a block cipher whose encryption function can be expressed in terms of a univariate polynomial function of moderate degree. The attacker is presented with a set of  $(d + 1)$  plaintext/ciphertext pairs  $(p_i, c_i)$  that are encrypted under one single key. If we consider  $p_i, c_i$  as elements of a field  $\mathbb{F}$  (e.g.  $\mathbb{F} = \mathbb{F}_{2^n}$ ), then the Lagrange Interpolation Formula states that the unique polynomial function  $f : \mathbb{F} \rightarrow \mathbb{F}$  mapping  $p_i$  to  $c_i$  is given by

$$f(x) = \sum_{i=0}^d c_i \prod_{\substack{j=0 \\ j \neq i}}^d \left( \frac{x - p_j}{p_i - p_j} \right).$$

If the block cipher encryption can be expressed as a polynomial function of degree  $d$ , then the encryption operation is given by the above polynomial function  $f$ . This function can then be used to encrypt any plaintext, or to decrypt any ciphertext, without knowledge of the secret key. However, we see this attack as effective only if the number of coefficients to be interpolated is substantially smaller than the number of entries in the code book (usually implying  $d$  being reasonably small).

Thus instead of attempting to describe the cipher as a single polynomial, perhaps a more promising approach is to express the encryption operation as a *system* of polynomial equations. Modelling the encryption process of a block cipher as a polynomial system presents us with several obvious questions. Should we have a system in few variables but with polynomials of high degree or should we rather have more variables but equations of lower degree? Since block ciphers can always be seen as vectorial Boolean functions, they can be described as a polynomial system over  $\mathbb{F}_2$ . Certain components are hard or almost impossible to express as “compact” polynomials, however. For instance, key-dependent S-Boxes such as those used in Blowfish (Schneier 1994) do not lend themselves to easy polynomial descriptions. In this particular case, the S-Boxes are generated algorithmically in a way that is most disadvantageous for compact polynomial descriptions. Similarly,

mixing operations over fields with different characteristics or—to a lesser extent—mixing modular arithmetic with bitwise operations can also be an effective measure against simple polynomial representations over a common field or ring.

In practice, systems of equations for block ciphers are either written over  $\mathbb{F}_2$  or—if all S-Boxes have both  $s$  input and  $s$  output bits—over  $\mathbb{F}_{2^s}$ . To keep the degree of the polynomials low, the encryption process is modelled by considering each layer (i.e. the linear and non-linear steps of each encryption round) separately.

In the following, let  $\mathbb{F}_q$  be the alphabet and  $N_\ell$  the number of layers of the cipher,  $n$  the block size of the cipher and  $m$  the cipher key size.<sup>3</sup> For each layer  $\ell$  of a block cipher, a set of so-called *state variables*  $\mathcal{X}_\ell := \{x_{\ell,1}, \dots, x_{\ell,n}\}$  is introduced. These variables represent the internal state of the cipher after the execution of the  $\ell$ -th layer. For the  $\ell$ -th layer these variables are also called *output variables*; correspondingly the state variables of the  $(\ell - 1)$ -th layer are called *input variables* of the  $\ell$ -th layer. The input and the output of the cipher are either fixed, or modelled by *plaintext variables*  $\mathcal{P} := \{x_{0,1}, \dots, x_{0,n}\}$  and *ciphertext variables*  $\mathcal{C} := \{x_{N_\ell,1}, \dots, x_{N_\ell,n}\}$ , respectively. For the encryption key, a set of *cipher key variables*  $\mathcal{K} := \{k_1, \dots, k_m\}$  is used. The set of all variables will from now on be denoted by  $\mathcal{X}$ .

The system of equations for a block cipher can typically be separated into three parts: linear equations describing the diffusion layer and key additions of the cipher, polynomial equations for the substitution layers of the cipher, and a set of *key-schedule equations*.

In some cases where key-schedule equations can be omitted, e.g. if round keys are generated by a selection of bits of the cipher. More commonly the key schedule has a similar structure to the encryption. In this case, we denote as  $\mathcal{K}_\ell := \{k_{\ell,1}, \dots, k_{\ell,j}\}$  the  $j$  key state variables for the layer  $\ell$ .

Note that linear layers can often be merged with neighbouring substitution layers without changing the degree of the system. In this case, the sparsity of the resulting equations will be reduced unless the linear layer simply permutes the elements of the internal state.

Let then  $I \subset \mathbb{F}_q[\mathcal{X}]$  be the ideal associated with the encryption process of a block cipher. The hope is thus that one can compute the Gröbner basis  $G$  of  $I$  to recover the encryption secret key (see however Sect. 4.7).

## 4.2 Field Equations

For cryptographic purposes, only solutions over the ground field are of importance; solutions in the algebraic closure are irrelevant. For a polynomial ring  $R = \mathbb{F}_q[\mathcal{X}]$  we can write the set of *field equations* of the form  $x^q - x = 0$  for all  $x \in \mathcal{X}$ . These equations in effect force all assignments of variables to be fixed under the automorphism  $x \mapsto x^q$ . We note that the left-hand side of the equations, the *field polynomials* form a universal Gröbner basis, and as a consequence, regardless of the term

---

<sup>3</sup>Note that  $n$  and  $m$  only denote bit sizes if  $q = 2$ .

ordering, the exponents of all polynomials can be reduced modulo  $q$  during Gröbner basis computations. Of particular interest is the case where  $q = 2$ , where we have the following simple proposition.

**Proposition 1** *The maximal degree of polynomials occurring in the computation of a Gröbner basis of a polynomial ideal in  $R = \mathbb{F}_2[\mathcal{X}]$  with  $n$  variables containing the field polynomials is at most  $n$ .*

Adding the field polynomials to a set  $P \subset R$  is equivalent to working over the quotient ring  $R' = R/J$ , where  $J = \langle x_0^q - x_0, \dots, x_n^q - x_n \rangle$ . For  $\mathbb{F}_2$ , this quotient ring is called the *ring of Boolean functions*. Computer algebra systems such as recent versions of MAGMA et al. (2008) (v2.11 onward) are able to detect the presence of field polynomials in the input and work with an exponent-reduced representation internally for the case of  $\mathbb{F}_2$ .

### 4.3 Polynomial Systems over $\mathbb{F}_2$

Modern block ciphers are designed to be either implemented in hardware or to be executed on computers. Henceforth choosing  $\mathbb{F}_2$  as ground field comes as a natural choice. Essentially the polynomial system is a decomposition of a Boolean circuit implementing the block cipher. For ciphers using S-Boxes over different fields of characteristic two, such as the MISTY family of ciphers, or ciphers with contracting or expanding S-Boxes, e.g. DES, writing equations over  $\mathbb{F}_2$  is the only obvious choice to obtain a polynomial system describing the complete cipher.

For a number of widely-used block ciphers, polynomial systems over  $\mathbb{F}_2$  have appeared in the literature. Courtois and Pieprzyk (2002b) initially presented systems of quadratic equations for Rijndael and Serpent (for a more thorough analysis of the systems arising from the AES, see Cid et al. 2007). Biryukov and de Cannière (2003) constructed systems of quadratic equations over  $\mathbb{F}_2$  for other block ciphers, such as Khazad, MISTY1, Kasumi and Camellia-128 together with explicit counts of the number of variables, equations and terms. These are usually very large, often sparse systems, with over a thousand of variables and equations (for example, the AES with 128-bit keys can be expressed as a system of 9600 quadratic and linear equations, of which 1600 are field equations Cid et al. 2007).

In all of these cases it makes sense to perform all computation directly in the ring of Boolean functions. Recently, Brickenstein and Dreyer have proposed novel strategies and polynomial representations dedicated to exactly this problem (Brickenstein and Dreyer 2007). These are described in further detail in Sect. 8.

### 4.4 Equations for Non-linear Components

A crucial part of generating a polynomial system for a block cipher is finding a suitable polynomial representation of the S-Boxes. Usually S-Boxes are given in the

form of look-up tables. These can be interpolated to obtain the corresponding polynomials. Equations over  $\mathbb{F}_2$  can be easily calculated by trace maps if a polynomial representations of an S-Box over an extensions field of  $\mathbb{F}_2$  is given. More generally, a linearly independent set of polynomials of maximum degree  $d$  for a  $s$ -bit S-Box can be found by triangulating a  $2^s \times t_d \times$  matrix where  $t_d = \sum_{i=0}^d \binom{2^s}{i}$  is the total number of terms (Biryukov and De Cannière 2003, Appendix A).

## 4.5 Equations for Inversion over $\mathbb{F}_{2^n}$

To thwart differential and linear cryptanalysis, certain classes of power functions over extension fields of  $\mathbb{F}_2$  have been proposed as S-Boxes (Nyberg 1994). These allow the designer to give upper bounds on the probabilities of the best linear approximations and the best differential characteristics. One of the functions proposed is the so-called *patched inversion*, which maps every element  $x \neq 0$  to  $x^{-1}$ . The zero element is mapped to itself. For a field  $\mathbb{F}_q$  with  $q > 2$ , this is equivalent to the power mapping  $x \mapsto x^{q-2}$ .

Due to its strong properties against conventional cryptanalysis, the patched inversion has become a popular choice for providing non-linearity to block ciphers and stream ciphers. To avoid easy algebraic descriptions when using the patched inverse function as S-Box, cipher designers often use functions that are *affinely equivalent* to Inv, i.e. a function  $\bar{I} = A_1 \circ \text{Inv} \circ A_2$ , where  $A_1, A_2$  are affine transformations over  $\mathbb{F}_2$ ; we call these *inversion-based* S-Boxes.

Boolean polynomials emanating from the Rijndael S-Box were considered in Courtois and Pieprzyk (2002a). When modelling inversion-based S-Boxes, one must consider whether the chosen representation holds for all input/output pairs of the S-Box, or whether it contains a defect for the zero element. For real-world ciphers consisting of many S-Box applications, such as the AES-128, the accumulated defect is considerable: one of the 24 equations listed for the S-Box in Courtois and Pieprzyk (2002b) is not valid when the input of the S-Box is zero. When including this equation for all 200 S-Boxes, the probability of the polynomial system holding for a random plaintext/ciphertext pair decreases to  $(\frac{255}{256})^{200} \approx 45.7\%$ .

Inversion-based S-Boxes may also allow us to express the encryption process as a system of polynomial equations over the extension field  $\mathbb{F}_{2^s}$ . In this case, one could either consider the function  $x \mapsto x^{2^s-2}$  or  $x \mapsto x^{-1}$ . The latter representation again makes the attack probabilistic, but may give much simpler equations.

## 4.6 Block Cipher Embeddings

Expressing the encryption process as a system of polynomial equations over a larger field  $\mathbb{F}_{2^s}$  may offer advantages, such as reducing the number of variables or increasing the sparseness of the polynomial system. For the particular example of the AES,

the main problem of obtaining such a “simple” polynomial description over  $\mathbb{F}_{2^8}$  lies in the affine transformation of the S-Box. Murphy and Robshaw proposed a simplified description of the AES by embedding the cipher into a larger cipher called BES (Murphy and Robshaw 2002). This embedding is achieved by applying a *vector conjugate mapping*  $\phi$  to each byte  $a \in \mathbb{F} = \mathbb{F}_{2^8}$  handled by the cipher, resulting in a *vector conjugate*  $\tilde{a} \in \mathbb{F}^8$

$$\phi : \mathbb{F} \mapsto \mathbb{F}^8, \quad a \rightarrow \tilde{a} = (a^{2^0}, a^{2^1}, \dots, a^{2^7}).$$

The result from the embedding of the AES-128 is a cipher that takes a 128-byte key and a 128-byte plaintext as input, has an 128-byte internal state and outputs a 128-byte ciphertext. The most striking effect of this embedding is that the affine transformation of the S-Box is moved into the diffusion layer of BES. Hence the non-linear layer consists of simple (patched) inversions. An advantage of this representation is that the overall number of terms occurring in the corresponding polynomial system is reduced, therefore making it considerably more sparse when compared to the usual description over  $\mathbb{F}_2$ . To make sure that all solutions to a BES system can be mapped back to solutions for the AES, equations to enforce the vector conjugate property for each variable can be added (this also ensures that all solutions of the system are in  $\mathbb{F}_{2^8}$ ). The real effectiveness of BES has however been questioned in Toli and Zanoni (2005).

We note that the embedding technique is not generally applicable to block ciphers but rather takes advantage of the AES rich algebraic structure (block cipher embeddings were considered in Cid et al. 2005a).

## 4.7 Direct Construction of Gröbner Bases

Surprisingly, for some block ciphers, a zero-dimensional Gröbner basis for the key-recovery ideal can be constructed with minimal computational effort—without performing a single polynomial reduction. Examples are AES-128 (Buchmann et al. 2006a) as well as ciphers of the Flurry and Curry families (Buchmann et al. 2006b).

This is achieved by constructing a polynomial system in which all leading terms are pairwise prime, allowing the first Buchberger criterion to be used to show that the resulting set of polynomials forms a Gröbner basis. The Gröbner basis is constructed from a direct description of the block cipher and its key schedule over  $\mathbb{F}_{2^n}$  merely by linearly combining polynomials and choosing an appropriate graded term ordering.

Observe that for an ideal  $I \subset \mathbb{F}_q[x_1, \dots, x_n]$  to be zero-dimensional, any set  $P$  of polynomials generating this ideal must have at least  $n$  elements; the number of solutions in the algebraic closure  $\overline{\mathbb{F}_q}$  otherwise is not finite. If the number of elements of  $P$  is exactly  $n$  and all leading terms of  $P$  are of the form  $x_i^e$  and pairwise prime,  $P$  is a zero-dimensional Gröbner basis of  $I$ . For the non-linear equations we avoid inversion-based representations of the S-Boxes by writing the inversion operation as a power polynomial.

The Gröbner basis of the key-recovery ideal for AES-128  $I_{\text{AES}} \in R_{\text{AES}}$  given in Buchmann et al. (2006b) consists of 200 polynomials of degree 254 and 152 linear polynomials in a ring of 352 variables. The vector space dimension  $\dim(R_{\text{AES}}/I_{\text{AES}})$  unfortunately is  $254^{200}$ , which makes the Gröbner basis unsuitable for cryptanalysis. This is due to the field equations not being captured by the Gröbner basis.

## 5 Small Scale and Experimental Ciphers

Algebraic attacks have received a lot of attention of the cryptographic community in the last few years. However there has not been much progress in assessing whether they can be effective against block ciphers in general. The main reason seems to be that the size of systems arising from block ciphers are completely out of reach for the current computational power. For most other methods of cryptanalysis it is quite straightforward to perform experiments on reduced-round versions of the cipher to understand how the attack might perform. This has not been the case for algebraic attacks on block ciphers.

One possible approach is to work on small scale variants of block ciphers, in order to test the effectiveness of the main algorithms in solving the systems of algebraic equations. While it is clearly not an easy task to design small versions that can replicate the main cryptographic and algebraic properties of a particular cipher, the hope is however that experiments on small versions can provide a preliminary insight into the behaviour of algebraic cryptanalysis on block ciphers.

With this goal in mind, a number of small scale ciphers have been proposed in recent years. Below we briefly describe the most relevant ones. Results of experiments using these ciphers are discussed in Sect. 6.

### 5.1 Small Scale Variants of the AES

A family of small scale variants of the AES aiming to provide a fully parameterised framework for detailed analysis of the cipher was proposed in Cid et al. (2005b).

The ciphers, denoted as  $\text{SR}(r, n_R, n_C, e)$  and  $\text{SR}^*(r, n_R, n_C, e)$ , are parameterised in the following way.<sup>4</sup>

- $r$  is the number of rounds,
- $n_R$  is the number of rows in the rectangular grid of the state,
- $n_C$  is the number of columns in the rectangular grid of the state,
- $e$  is the word size (in bits).

---

<sup>4</sup>The two variants differ insofar as the SR family uses the MixColumns operation in the last round, whereas the SR\* family omits it.

Both  $\text{SR}(r, n_R, n_C, e)$  and  $\text{SR}^*(r, n_R, n_C, e)$  have a block size of  $n_R n_C e$  bits and the full AES is modelled by  $\text{SR}^*(10, 4, 4, 8)$ . The data block is viewed as an  $n_R \times n_C$  array of words of  $e$  bits. Useful small scale variants exist when both  $n_R$  and  $n_C$  are restricted to 1, 2 and 4. The word sizes  $e = 4$  and  $e = 8$  are the most relevant and are defined with respect to the fields  $\mathbb{F}_{2^4}$  and  $\mathbb{F}_{2^8}$ . The field  $\mathbb{F}_{2^4}$  is defined by a primitive polynomial over  $\mathbb{F}_2$ , while small scale variants over  $\mathbb{F}_{2^8}$  use the same field as the AES (Daemen and Rijmen 2002).

A round of the small scale variants over a finite field is defined using small scale versions of the AES round operations (namely, SubBytes, ShiftRows, MixColumns and AddRoundKey) (Cid et al. 2005b). Furthermore, the corresponding key schedules is also defined.

These small scale variants seem to retain, as far as possible, the algebraic features of the AES. We note that they often have a small key space and can therefore be easily analysed by exhaustive key search or equivalent techniques. However, the main purpose of these small scale variants is to assist in the algebraic analysis of the AES. Experimental results based on these small scale variants are discussed in Sect. 6. A generator for SR and SR\* equation systems written by Martin Albrecht is contained in the SAGE computer algebra system (Stein 2008) since version 2.8.5.

## 5.2 Flurry and Curry

FLURRY and CURRY are two families of experimental block ciphers specifically designed to investigate algebraic attacks. The FLURRY family consists of Feistel networks while the ciphers in the CURRY family are SP-Networks.

An instance of the FLURRY family is denoted as  $\text{FLURRY}(k, m, r, f, D)$ , a CURRY instance as  $\text{CURRY}(k, m, r, f, D)$ . The parameters of these ciphers are as follows:

- $k$ : the extension degree for the base field  $\mathbb{F} := \mathbb{F}_{2^k}$ .
- $m$ : for Flurry, the block size consists of  $2m$  elements; for Curry, the internal state is a  $m \times m$  matrix.
- $r$ : number of rounds.
- $f$ : non-linear mapping (S-Box), which has to be bijective for CURRY.
- $D$ : diffusion matrix  $D \in \mathbb{F}^{m \times m}$ .

One of the underlying design goals of these cipher families is immunity against classical linear and differential cryptanalysis. In order to achieve this goal the S-Box and the diffusion matrix have obviously to be chosen accordingly. In Buchmann et al. (2006b) the designers propose MDS matrices for the diffusion and several low-degree power polynomials as well as the patched inversion function as choices for the S-Boxes. These parameters give very clean representations over  $\mathbb{F}_{2^k}$ .

### 5.3 Other Examples

Other small scale variants of the AES have also been proposed, though usually as an educational, rather than an experimental, tool (Musa et al. 2003; Phan 2002). For the particular purpose of experiments with algebraic cryptanalysis, a *toy* cipher CTC has also been proposed (Courtois 2006). This is a simple cipher with a minimal 3-bit S-Box which was claimed to be broken using algebraic techniques with the attack not being described in detail. CTC has been shown to be broken by linear and differential cryptanalysis (Dunkelman and Keller 2006), resulting in CTC2 being proposed (Courtois 2007). More experiments with CTC were presented in Albrecht (2007).

## 6 Experimental Results

Some of the small scale ciphers introduced in Sect. 5 have been used in experiments to evaluate the effectiveness of algebraic attacks against block ciphers. Although most of these simple experiments were performed using off-the-shelf software with limited computing resources, they are helpful as a preliminary assessment of algebraic attacks as a cryptanalytic technique against block ciphers. In particular, they may help understand how the various components of a cipher contribute to the complexity of algebraic attacks, and how the use of dedicated algebraic methods can improve the effectiveness of algebraic attacks.

### 6.1 Small Versions of the AES

Regarding the small scale variants of the AES (Sect. 5.1), experimental results on using Gröbner basis methods for solving the equation systems arising from these ciphers were presented in Cid et al. (2005b, 2007). The experiments were performed using an efficient implementation of the  $F_4$  algorithm (Faugére 1999), and the Gröbner bases were computed with respect to the *degrevlex* term ordering.

In general, the results showed that the computations became intractable quite early, and comparative tests performed indicated that some specific AES features appear to make algebraic attacks quite hard. In particular, it seems that the inter-word diffusion (which is highly efficient in the AES) plays an important role in the complexity of the computations.

Overall, despite the use of limited computer power, the experiments seem to indicate that general purpose Gröbner basis methods are unlikely to solve a full equation system emanating from the AES. However, systems arising from ciphers are very structured and with special properties. These may be explored in designing dedicated algebraic methods against block ciphers. We discuss some of these in Sect. 7.

Brickenstein and Dreyer later presented impressive results against small-scale variants of the AES using their PolyBoRi suite (Brickenstein and Dreyer 2007),

using significantly less memory than Magma while at the same time being orders of magnitude faster. It should be noted however that these equation systems were preprocessed before the actual basis computation.<sup>5</sup>

Other experiments on equation systems with a similar structure to the AES equation systems were presented in Ars (2005). These results on very small systems seem to indicate that the maximum degree of polynomials obtained during the running of the  $F_5$  (Faugère 1999) algorithm is bounded by a reasonably small value for any number of rounds. This would suggest that the complexity of solving such a system is not what we would expect from a comparable generic system. However the connection between the equation systems in Ars (2005) and the AES equation system is not sufficiently strong to conclude that an AES equation system would behave in a similar manner.

## 6.2 Flurry and Curry

In contrast to the small-scale AES variants, experiments for 128-bit instantiations of Flurry and Curry were carried out over fields of large order in Buchmann et al. (2006a). Henceforth the field polynomials were not taken into account. The equations considered described a key recovery scenario from a single plaintext/ciphertext pair. Again, Magma was used for conducting the experiments. For efficiency reasons, Magma first computes a degrevlex Gröbner basis which is then converted to a lexicographical Gröbner basis using either the FGLM algorithm (Faugère et al. 1993) or the Gröbner Walk (Stéphane Collart et al. 1997) algorithm. The degrevlex Gröbner basis was computed using the  $F_4$  algorithm. For ciphers with S-Boxes described by power polynomials, the bulk of the computational work for the key recovery to happened in the Gröbner basis conversion step. Benchmarking the two Gröbner basis conversion algorithms against each other gave inconclusive results. The method described in Sect. 4.7 allowed to reduce the key-recovery problem to a Gröbner basis conversion problem. Hence an upper bound on the space and time complexity of this step was derived, making use of FGLM.

In the experiments presented in Buchmann et al. (2006a), the inversion S-Box—in Flurry and Curry used without affine linear transforms on the input and output bits—offered less resistance against the algebraic attacks over large finite fields than a power polynomial. Theoretical results for this case were not obtained.

Faugère investigated different attack scenarios on Faugère (2007). In the following we summarize the results achieved with  $F_5$ . One approach investigated were computations on overdefined systems of equations that result from fixing a single variable to a guessed value. In this case, a speed-up over a direct attack only occurred for fields of small size. Systems of equations for an attack with multiple

---

<sup>5</sup>In Brickenstein and Dreyer (2007) the authors state that “we made some optimizations on the formulations of the equations on it [sic]”, without however going into further detail.

known plaintext/ciphertext pairs were found to be harder to solve than systems for a single plaintext/ciphertext pair.

For chosen plaintexts the situation looked different. Plaintext/ciphertext pairs were generated by adding unit vectors of the canonical basis to a random plaintext and encrypting them. In this case, the resulting systems of equations sometimes became much easier to solve than equations from a single pair. Up to 6 rounds were attacked for a Flurry cipher with the inversion S-Box in this scenario. For 7 rounds this attack works for S-Boxes represented by power functions, but not by the inverse function. Faugère claims the degree of polynomials in the Gröbner basis computations in the multiple chosen-plaintext scenario for Flurry with  $x \mapsto x^3$  to be bounded and conjectures an attack complexity polynomial in the number of rounds.

### 6.3 Other Experiments

For the experimental cipher CTC, its author claims that algebraic key-recovery attacks up to 6 rounds are practical for a 255-bit version (Courtois 2006). Multiple plaintext/ciphertext pairs were used for breaking the 6 round version. No further details are given on the method employed.

## 7 Attack Strategies

As remarked in Sect. 6, it seems very unlikely that general methods from computer algebra can be used in a straightforward manner to solve the systems of equations arising from modern block ciphers. We note however that these systems are highly structured (see below) and sparse. Thus a more promising approach may be to apply some *dedicated method*, based on techniques from computer algebra, but aiming to exploit the special properties of a target system. Below we discuss some of the methods proposed.

### 7.1 Meet-in-the-Middle and Incremental Techniques

The iterative nature of modern block ciphers means that the associated systems of equations are typically structured in blocks, with each block containing the equations for one round. Variables in one block only occur in neighbouring blocks or within the relevant part of the key schedule.

A promising technique to find the solution for systems with such structure is to employ a *meet-in-the-middle* approach (Cid et al. 2005b, 2007). The system consisting of  $r$  blocks (i.e. rounds) is divided into two subsystems for  $\frac{r}{2}$  rounds.<sup>6</sup> We

---

<sup>6</sup>We assume without loss of generality that  $r$  is even.

regard the output variables of the first equation subsystem as the input variables of the second equation subsystem. We can then compute the Gröbner bases of the two corresponding subsystems, using an appropriate elimination ordering. We then eliminate variables that do not appear in rounds  $\frac{r}{2}$  and  $\frac{r}{2} + 1$ . This gives two small systems of equations in variables from the two systems that are simply related by the round keys. These two equation systems can then be combined with some additional equations from the key schedule and solved to obtain the key. Experimental results using this approach on AES variants were presented in Cid et al. (2005b) and seem to confirm that a meet-in-the-middle technique may be more efficient than directly solving the full system of equations arising from a block cipher.

One possible drawback to this approach is that computations using elimination orderings are known to be less efficient than those with degree orderings, and we might expect that using an elimination ordering in both subsystems would give only limited advantages over using the degree reverse lexicographical ordering for the full system. An alternative approach would be to simply compute the Gröbner bases for the two subsystems using the most efficient ordering and then to combine both results to compute the solution of the full set equations. Some experimental results on this approach presented in Cid et al. (2005b) indicate that this approach can in fact be more efficient for larger examples of the small scale variants of the AES. This suggests the applicability of a more general *divide-and-conquer* approach to the problem of solving the equation system deriving from iterated block ciphers.

An incremental method named *Gröbner Surfing* related to this idea was proposed in Albrecht (2007). Here, a Gröbner basis of the key-recovery ideal is computed round by round: let  $\mathcal{P}_r$  be the equations for the  $r$ -th round of the cipher and GB the Gröbner basis algorithm with the plaintext variables fixed in the first round and the ciphertext variables fixed in the last, the  $N_r$ -th round. The idea then is to decompose the Gröbner basis computation as follows:

$$\text{GB}\left(\bigcup_{r=1}^{N_r} \mathcal{P}_r\right) = \text{GB}(\mathcal{P}_{N_r} \cup \text{GB}(\mathcal{P}_{N_{r-1}} \cup (\dots \cup \text{GB}(\mathcal{P}_1))).$$

Alternatively this method may be expressed as a selection strategy for the critical pairs in the Gröbner basis algorithm. For this method to succeed more efficiently than a direct computation of a Gröbner basis, a suitable term ordering is crucial. Block orderings with graded term orderings inside the blocks and block splits at the round or layer boundaries seem to be a suitable choice. Experimental results on the Gröbner Surfing technique applied to small instances of CTC are presented in Albrecht (2007). It was found that a Gröbner Surfing strategy on CTC performed better than a straightforward degrevlex Gröbner basis computation.

## 7.2 Differential-Algebraic Cryptanalysis

A recent trend in block cipher cryptanalysis is to combine algebraic approaches with traditional methods of cryptanalysis. In Albrecht and Cid (2008) an attack is pro-

posed that combines algebraic techniques with differential cryptanalysis. In differential cryptanalysis, given a *differential characteristic* covering  $r$  out of  $N_r$  rounds of a given block cipher, the cryptanalyst usually guesses subkey bits to overcome the last  $r_d = N_r - r$  rounds. This quickly becomes impractical as the number of rounds  $r_d$  grows. Albrecht and Cid (2008) investigate the block cipher PRESENT (Bogdanov et al. 2007), and are able to increase  $r_d$  from 2 to 4 rounds by using algebraic techniques. Specifically, the authors construct a system of polynomial equations for  $r_d$  rounds for pairs of plaintexts and use Gröbner basis algorithms to perform a consistency check. This allows them to determine whether a given pair satisfies the considered differential characteristic. Based on this observation, information about the encryption key could be recovered. The technique is in theory generally applicable to improve differential cryptanalysis although no experimental evidence of the feasibility of the attack against reduced versions of ciphers other than PRESENT are provided.

## 8 Alternative Methods for Solving Polynomial Systems

As stated in Sect. 4.3, representation of block ciphers in the ring of Boolean functions are of significant relevance for algebraic cryptanalysis. It therefore makes sense to optimize algorithms to perform computations over such rings. For these systems, alternative representations are possible. Brickenstein and Dreyer have recently proposed using a variant of binary decision diagrams (Lee 1959; Akers 1978), called zero-suppressed binary decision diagrams (ZDD) for representing Boolean functions in Gröbner basis computations (Brickenstein and Dreyer 2007). These make use of the fact that the Boolean functions dealt with have a sparse polynomial representation. One of the main ideas behind PolyBoRi is to preserve the sparsity of the representation throughout the computation, which keeps the memory consumption low. This is achieved by using an adapted version of the SlimGB algorithm (Brickenstein 2005), which was originally designed for a similar task.

Combinatorial methods have been investigated on reduced versions of the DES: Raddum and Semaev proposed a method using message-passing on a graph for solving multivariate equations from a reduced version of DES (Raddum and Semaev 2006). Up to 4 rounds of DES were successfully attacked using this approach. In Raddum and Semaev (2007), they studied the application of related techniques against some of the small scale variants of AES introduced in Sect. 5. Their results seem to be substantially better than the ones obtained in Cid et al. (2005b).

Courtois and Bard demonstrated the applicability of a state-of-the-art SAT-solver for cryptanalysing reduced round versions of DES. This was done by first setting up a  $\mathbb{F}_2$  polynomial system, converting it to CNF (Courtois and Bard 2007), assigning a number of variables to fixed values and then checking the satisfiability of the CNF clauses using the MiniSat solver (Een and Sorensson 2006). This allowed them to attack 6 rounds of DES with a single plaintext/ciphertext pair. The advantage of these methods lies in a significantly lower memory consumption than Gröbner basis algorithms based on linear algebra such as  $F_4$  and  $F_5$ .

Subsequently, Courtois, Bard and Wagner presented a combination of slide attacks and SAT-solver cryptanalysis against the KeeLoq cipher (Courtois et al. 2008).

## 9 Conclusions

In this paper we have presented an overview of the latest developments in the area of algebraic cryptanalysis against block ciphers. This is an area that has recently received a lot of attention from the cryptographic community. Many different methods have been considered, with however limited success so far in targeting modern block ciphers. In particular, to the authors' best knowledge, no modern block cipher with practical relevance has been successfully attacked using algebraic cryptanalysis faster than with other techniques. Nonetheless, algebraic cryptanalysis is a very active field of research, and should remain so in the coming years. We expect that algebraic techniques will continue to be used and become established as an important tool in the cryptanalysis of block ciphers.

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

## References

- S. B. Akers, *Binary decision diagrams*, IEEE Trans. on Computers **27** (1978), no. 6, 509–516.
- M. Albrecht, *Algebraic attacks on the Courtois Toy Cipher*, Master's thesis, Diplomarbeit—Universität Bremen, 2007.
- M. Albrecht and C. Cid, *Algebraic techniques in differential cryptanalysis*, Crypto. ePrint Arch., Rep. 2008/177, 2008, <http://eprint.iacr.org/>.
- F. Armknecht and G. Ars, *Algebraic attacks on stream ciphers with Gröbner bases*, this volume, 2009, pp. 329–348.
- G. Ars, *Applications of Gröbner bases to cryptography*, Ph.D. thesis, University of Rennes I, 2005.
- G. Ars, J. C. Faugère, H. Imai, M. Kawazoe, and M. Sugita, *Comparison between XL and Gröbner basis algorithms*, Proc. of Asiacrypt 2004 (P. J. Lee, ed.), LNCS, vol. **3329**, Springer, Berlin, 2004, pp. 338–353.
- O. Billet and J. Ding, *Overview of cryptanalysis techniques in multivariate public key cryptography*, this volume, 2009, pp. 263–283.
- A. Biryukov, *Methods of cryptanalysis*, Ph.D. thesis, Technion, 1999.
- A.A. Biryukov and C. De Cannière, *Block ciphers and systems of quadratic equations*, Proc. of FSE 2003, LNCS, vol. **2887**, Springer, Berlin, 2003, pp. 274–289.
- A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vinkelsoe, *PRESENT: An ultra-lightweight block cipher*, Proc. of CHES 2007, LNCS, vol. **7427**, Springer, Berlin, 2007, pp. 450–466.
- M. Brickenstein, *Gröbner bases with slim polynomials*, Reports in Comp. Alg. 35, Univ. Kaiserslautern, Kaiserslautern, 2005, <http://www.mathematik.uni-kl.de/>.
- M. Brickenstein and A. Dreyer, *PolyBoRi: A framework for Gröbner basis computations with Boolean polynomials*, Elec. Proc. of MEGA 2007, 2007, <http://www.ricam.oeaw.ac.at/mega2007/electronic/26.pdf>.

- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- J. Buchmann, A. Pyshkin, and R. P. Weinmann, *A zero-dimensional Gröbner basis for AES-128*, Proc. of FSE 2006, LNCS, vol. **4047**, Springer, Berlin, 2006a, pp. 78–88.
- J. Buchmann, A. Pyshkin, and R. P. Weinmann, *Block ciphers sensitive to Gröbner basis attacks*, Proc. of CT-RSA 2006, LNCS, vol. **3860**, Springer, Berlin, 2006b, pp. 313–331.
- C. Carlet, *Boolean methods and models*, ch. Boolean Functions for Cryptography and Error Correcting Codes, Cambridge University Press, 2009, to appear.
- C. Cid and G. Leurent, *An analysis of the XSL algorithm*, Proc. of ASIACRYPT 2005, LNCS, vol. **3788**, Springer, Berlin, 2005, pp. 333–352.
- C. Cid, S. Murphy, and M. J. B. Robshaw, *An algebraic framework for cipher embeddings*, Proc. of 10th IMA International Conference on Coding and Cryptography, LNCS, vol. **3796**, Springer, Berlin, 2005a, pp. 278–289.
- C. Cid, S. Murphy, and M. J. B. Robshaw, *Small scale variants of the AES*, Proc. of FSE 2005, LNCS, vol. **3557**, Springer, Berlin, 2005b, pp. 145–162.
- C. Cid, S. Murphy, and M. J. B. Robshaw, *Algebraic aspects of the Advanced Encryption Standard*, Springer, Berlin, 2007.
- N. T. Courtois, *How fast can be algebraic attacks on block ciphers?* Tech. Report Rep. 2006/168, Crypto. ePrint Arch., 2006, <http://eprint.iacr.org/>.
- N. T. Courtois, *CTC2 and fast algebraic attacks on block ciphers revisited*, Tech. Report Rep. 2007/152, Crypto. ePrint Arch., 2007, <http://eprint.iacr.org/>.
- N. T. Courtois and G. V. Bard, *Algebraic cryptanalysis of the data encryption standard*, Cryptography and Coding, LNCS, vol. **4887**, Springer, Berlin, 2007, pp. 152–169.
- N. Courtois and J. Pieprzyk, *Cryptanalysis of block ciphers with overdefined systems of equations*, Cryptology ePrint Archive 2002/044, 2002a, <http://eprint.iacr.org/2002/044/>.
- N. Courtois and J. Pieprzyk, *Cryptanalysis of block ciphers with overdefined systems of equations*, Proc. of ASIACRYPT 2002, LNCS, vol. **2501**, Springer, Berlin, 2002b, pp. 267–287.
- N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient algorithms for solving overdefined systems of multivariate polynomial equations*, Proc. of EUROCRYPT 2000, LNCS, vol. **1807**, Springer, Berlin, 2000, pp. 392–407.
- N. Courtois, G. V. Bard, and D. Wagner, *Algebraic and slide attacks on KeeLoq*, Proc. of FSE 2008, LNCS, vol. **5086**, Springer, Berlin, 2008, pp. 97–115.
- J. Daemen and V. Rijmen, *The design of Rijndael*, Springer, Berlin, 2002.
- O. Dunkelman and N. Keller, *Linear cryptanalysis of CTC*, Tech. Report Rep. 2006/250, Crypto. ePrint Arch., 2006, <http://eprint.iacr.org/>.
- N. Een and N. Sorensson, *MiniSat—a SAT solver with conflict-clause minimization*, 2006, <http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/Main.html>.
- J. C. Faugére, *A new efficient algorithm for computing Gröbner bases ( $F_4$ )*, J. Pure Appl. Algebra **139** (1999), nos. 1–3, 61–88.
- J. C. Faugére, *Gröbner bases. Applications in cryptology*, Talk at FSE 2007, 2007.
- J. C. Faugére, P. Gianni, D. Lazard, and T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, J. Symbolic Comput. **16** (1993), no. 4, 329–344.
- H. Feistel, *Cryptography and computer privacy*, Scientific American **228** (1973), no. 5, 15–23.
- T. Jakobsen and L. R. Knudsen, *The interpolation attack on block ciphers*, Proc. of FSE 1997, LNCS, vol. **1267**, Springer, Berlin, 1997, pp. 28–40.
- A. Kerckhoffs, *La cryptographie militaire*, Journal des Sciences Militaires (1883a), 161–191.
- A. Kerckhoffs, *La cryptographie militaire*, Journal des sciences militaires **IX** (1883b), 3–72.
- C. Y. Lee, *Representation of switching circuits by binary-decision programs*, Bell System Technical Journal **38** (1959), 985–999.

- C. W. Lim and K. Khoo, *Detailed analysis on XSL applied to BES*, Proc. of FSE 2007, LNCS, vol. **4593**, Springer, Berlin, 2007, pp. 242–253.
- MAGMA, J. J. Cannon, W. Bosma (eds.), *Handbook of MAGMA functions*, edition 2.15, 2008.
- A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*, CRC press series on discrete mathematics and its applications, CRC Press, Boca Raton, 1997.
- S. Murphy and M. J. B. Robshaw, *Essential algebraic structure within the AES*, Proc. of CRYPTO 2002, LNCS, vol. **2442**, Springer, Berlin, 2002, pp. 1–16.
- M. A. Musa, E. F. Schaefer, and S. Wedig, *A simplified AES algorithm and its linear and differential cryptanalysis*, Cryptologia **XXVII** (2003), no. 2, 148–177.
- National Bureau of Standards, *The Data Encryption Standard*, Federal Information Processing Standards Publication (FIPS) 46, 1977.
- National Institute of Standards and Technology, *The Advanced Encryption Standard*, Federal Information Processing Standards Publication (FIPS) 197, 2001.
- K. Nyberg, *Differentially uniform mappings for cryptography*, Proc. of EUROCRYPT 1993, LNCS, vol. **765**, Springer, Berlin, 1994, pp. 55–64.
- R. C. W. Phan, *Mini Advanced Encryption Standard (Mini-AES): A testbed for cryptanalysis students*, Cryptologia **XXVI** (2002), no. 4, 283–306.
- H. Raddum and I. Semaev, *New technique for solving sparse equation systems*, Cryptology ePrint Archive, Report 2006/475, 2006, <http://eprint.iacr.org/>.
- H. Raddum and I. Semaev, *Solving MRHS linear equations*, Proc. of WCC 2007, INRIA, 2007, pp. 323–332.
- B. Schneier, *The Blowfish encryption algorithm*, Dr. Dobb's Journal (1994), 38–40.
- C. E. Shannon, *Communication theory of secrecy systems*, Bell System Tech. J. **28** (1949), 656–715.
- T. Shimoyama and T. Kaneko, *Quadratic relation of S-box and its application to the linear attack of full round DES*, Proc. of CRYPTO 1998, LNCS, vol. **1462**, Springer, Berlin, 1998, pp. 200–211.
- W. Stein, *Sage: Open Source Mathematical Software (Version 2.8.5)*, The Sage Group, 2008, <http://www.sagemath.org>.
- S. Stéphane Collart, M. Kalkbrener, and D. Mall, *Converting bases with the Gröbner Walk*, J. of Symbolic Comput. **24** (1997), nos. 3–4, 465–469.
- I. Toli and A. Zanoni, *An algebraic interpretation of AES-128*, Proc. of AES 2004, LNCS, vol. **3373**, Springer, Berlin, 2005, pp. 84–97.

# Algebraic Attacks on Stream Ciphers with Gröbner Bases

Frederik Armknecht and Gwenolé Ars

**Abstract** Stream ciphers efficiently encrypt data streams of arbitrary length and are widely deployed in practice, e.g., in mobile phones. Consequently, the development of new mechanisms to design and analyze stream ciphers is one of the major topics in modern cryptography. Algebraic attacks evaluate the security of certain stream ciphers by exploring the question how an attack could be performed by generating and solving appropriate systems of equations. In this text, we give an introduction to algebraic attacks and provide an overview on how and to what extent Gröbner bases are useful in this context.

## 1 Introduction

Nowadays, it has become more and more common to exchange digital data over possibly long distances. As the involved communication channels are usually not within the control of the user, the risk of being eavesdropped by malicious third parties cannot be excluded. A usual countermeasure is to encrypt the data. Encryption is the process of obscuring information to make it unreadable without special knowledge. This means that the original data, the plaintext  $P$ , is modified before submission in such a way that the transmission, the ciphertext  $C$ , reveals no obvious information about the underlying message. Only a legitimate receiver should be able to undo the modification of the data, i.e., to decrypt it, to recover the original meaning. To prevent an outsider to decrypt the ciphertext, the actual encryption/decryption transformation require some additional information, a key  $K$ , which is kept secret from non-legitimate parties and has been exchanged previously between sender and receiver.

For certain practical applications, e.g. encryption in mobile phones, there is the need to encrypt data of arbitrary length as fast as possible. One widely approach is the use of stream ciphers based on keystream generators. Examples are the keystream generator  $E_0$  from the Bluetooth standard for wireless communication (Bluetooth specification v1.1 1999), A5/1 used in the GSM-encryption (Briceno

---

F. Armknecht  
Ruhr-Universität Bochum, Bochum, Germany  
e-mail: [frederik.armknecht@trust.rub.de](mailto:frederik.armknecht@trust.rub.de)

G. Ars  
Lycée Marie de Champagne, Troyes, Paris, France  
e-mail: [gwenole.ars@gmail.com](mailto:gwenole.ars@gmail.com)

et al. 1998; Zenner et al. 2000), and RC4 used in SSH, HTTPS, and WLAN (Fluhrer et al. 2001).

The cryptanalysis of encryption mechanisms is one of the major goals in modern cryptography. This means the evaluation of possible risks and attacks from malicious third parties. By and by, numerous possible attacks against keystream generators have been found and analyzed. In 2003, algebraic attacks against keystream generators were considered publicly for the first time, e.g. see Courtois and Meier (2003), Armknecht and Krause (2003), Ars (2005). As they outmatched for several keystream generators, e.g. for  $E_0$ , all previously published attacks, they gained more and more attention. Today, algebraic attacks are an established tool for the cryptanalysis of keystream generators.

Algebraic attacks are based on generating and solving a system of non-linear equations over a finite field. Thus, it is not surprising that the application of Gröbner bases (see Buchberger 1965, 1970, 1985, 1998, 2006) for algebraic attacks has been and still is subject of ongoing research, e.g., see Ars and Faugère (2003, 2005), Ars (2005). This text provides a survey on the topics for which Gröbner bases proved to be useful for algebraic attacks (see Mora 2009 for a basic explanation of Gröbner basis theory).

The text is structured as follows. After giving in Sect. 2 an introduction into the type of keystream generators considered in algebraic attacks, the attacks themselves are explained in Sect. 3. As we will see, the two most important questions in the context of algebraic attacks are the search for useful equations and the task of solving the system of equations. Consequently, Sect. 4 will show how Gröbner bases can be used for finding equations and Sect. 5 how they can be used for computing the solution. Section 6 gives a final conclusion.

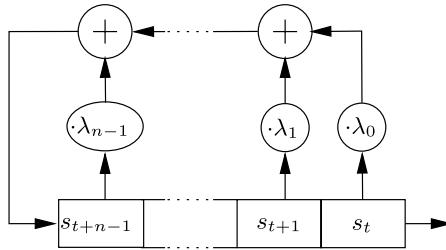
## 2 Keystream Generators

Keystream generators are mechanisms that generate arbitrary long sequences, called the keystream, depending on an initial value which is equal to or derived from the secret key. Therefore, only parties which know the key are (or at least should be) able to generate the keystream. The keystream is then used to encrypt the plaintext stream.

Keystream generators comprise an internal state, an update function, and an output function. At the beginning, the internal state is set to an initial state  $S_0$ . The internal state is modified in regular time intervals, called clocks, according to the update function. The state at clock  $t$  is denoted by  $S_t$ . At the beginning of a clock  $t$ , the current keystream element  $z_t$  is generated by evaluating the output function  $f$  on the actual state  $S_t$ , that is  $z_t = f(S_t)$ .

To encrypt a plaintext stream  $P = (p_0, p_1, \dots)$ , one generates a keystream  $Z = (z_0, z_1, \dots)$  determined by  $S_0$  and encrypts  $P$  to a ciphertext stream  $C =$

**Fig. 1** Schematic picture of an LFSR



$(c_0, c_1, c_2, \dots)$  via  $c_t := E_{z_t}(p_t)$ .<sup>1</sup> A legitimate receiver can initialize the keystream generator with the same initial state  $S_0$  as well and generate the same keystream  $Z$  to decrypt  $C$  via  $p_t = D_{z_t}(c_t)$ . The most common case in practice is that  $p_t$ ,  $c_t$ , and  $z_t$  are bit vectors and that encryption and decryption is simply the bitwise XOR-operation.

In practice, linear feedback shift registers (LFSRs) turned out to be good building blocks for keystream generators. In the following, we recapitulate some basic facts on LFSRs. For further reading, we recommend Lidl and Niederreiter (1986).

**Definition 1** Let  $\mathbb{F}_q$  denote the finite field of size  $q$ , with  $q$  being a prime power. A *linear feedback shift register* of length  $n$  is a finite state machine with an internal state, being an element from the vector space  $\mathbb{F}_q^n = (\mathbb{F}_q)^n$ , and a linear feedback function  $\sum_{i=1}^n \lambda_{i-1} \cdot x_i \in \mathbb{F}[x_1, \dots, x_n]$  with  $\lambda_0 \neq 0$ .

An LFSR is regularly clocked. At each clock, one field element is given out and the internal state is changed. Let  $S_t = (s_t, \dots, s_{t+n-1}) \in \mathbb{F}_q^n$  be the internal state at clock  $t$ . The output at clock  $t$  is defined to be  $s_t$ , that is the left-most entry of  $S_t$ . The internal state is updated according to

$$S_t = (s_t, \dots, s_{t+n-1}) \mapsto S_{t+1} := \left( s_{t+1}, \dots, s_{t+n-1}, \sum_{i=0}^{n-1} \lambda_i \cdot s_{t+i} \right). \quad (1)$$

We call the sequence  $(s_t)_{t \geq 0}$  the *LFSR sequence*. As the update function (1) is linear, there exists a matrix  $L$  such that  $S_t := S_0 \cdot L^t$ . The matrix  $L$  is called the *feedback matrix*. Observe that  $\lambda_0 \neq 0$  implies that the change of the internal state is reversible.

Figure 1 displays a schematic figure of an LFSR. LFSRs can be used to produce streams  $(s_t)_{t \geq 0}$  of arbitrary length. The advantage of LFSRs is that they can be implemented efficiently in hardware (at least for the case  $\mathbb{F}_2$ ), making them particularly interesting for restricted devices as mobile phones. Another advantage is that LFSRs and their sequences are mathematically well understood. Unfortunately, from a cryptographic point of view, LFSRs alone are extremely weak as they the initial state can be reconstructed from the outputs by solving a system of linear equations.

<sup>1</sup>According to the established notation, we write  $E_K(\cdot)$  for the encryption using a key  $K$  and similarly  $D_K(\cdot)$  for the decryption.

Thus, to strengthen LFSR-based keystream generators, one has to incorporate some kind of non-linearity.

The most popular approach is to apply a non-linear function to the outputs of several LFSRs (or several outputs of one LFSR) or to include a second finite state machine with a non-linear update function and to combine the contents of both to compute the keystream. These approaches can be described by a general kind of keystream generator, which we call  $(m, \ell)$ -combiners. A formal definition is the following:

**Definition 2** A  $(m, \ell)$ -combiner consists of the following components:

- an internal state  $S \in \mathbb{F}_q^m \times \mathbb{F}_q^n$  with
  - the first part of  $S$ , being from  $\mathbb{F}_q^m$ , is the content of the memory register of length  $m$  and
  - the second part of  $S$ , being from  $\mathbb{F}_q^n$ , is the internal state of  $s$  LFSRs of lengths  $n_1, \dots, n_s$  with  $n = n_1 + \dots + n_s$ ,
- a (projection) matrix  $P$  over  $\mathbb{F}_q$  of size  $n \times \ell$ , used to select some elements from the LFSRs internal state for further computation,
- a non-linear next memory state function  $\Psi : \mathbb{F}_q^m \times \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q^m$ , used to update the memory register, and
- an output function  $f : \mathbb{F}_q^m \times \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q^o$ , used to compute the keystream.

If  $m \geq 1$ , then we speak of a *combiner with memory*, else of a *simple combiner*.

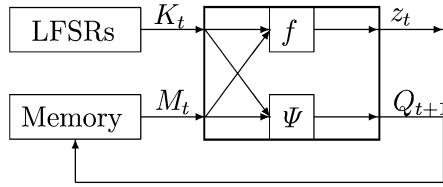
Let  $L_1, \dots, L_s$  be the LFSR feedback matrices and  $L := \text{diag}(L_1, \dots, L_s)$ . The generation of the keystream works as follows. At the beginning, the memory and the LFSRs are initialized with some value  $S_0 \in \mathbb{F}_q^m \times \mathbb{F}_q^n$ . The first part is the initial state of the memory and the second part the (concatenation of) the internal states of the LFSRs. The goal of an algebraic attack, as we will explain in the next section, is to recover the LFSRs' initial state.<sup>2</sup> Therefore, this value is usually treated as the secret key and denoted by  $K$ . Thus, if we assign  $M_0 \in \mathbb{F}_q^m$  to be the memory's initial setting, it holds that  $S_0 = (M_0, K)$ .

At each clock  $t$ , the actual states of the LFSRs and of the memory register are used to compute the next keystream element  $z_t$  and to update the LFSRs and the memory register. In many cases, only a fraction of the LFSRs' internal states are used for the computation of  $z_t$  and the next state of the memory register. This means that only the values in  $K \cdot L^t \cdot P$  are involved in the computations at clock  $t$  with  $P \in \mathbb{F}_q^{n \times \ell}$  being an appropriate projection matrix. We abbreviate  $K \cdot L^t \cdot P$  to  $K_t$  and call it the *input* of the  $(m, \ell)$ -combiner at clock  $t$ . The memory is updated via  $M_{t+1} := \Psi(M_t, K_t)$ , whereas the keystream element  $z_t$  is computed by  $z_t = f(M_t, K_t)$ . Summing up, the state is updated as follows:

$$S_t \mapsto S_{t+1} = (\Psi(M_t, K_t), K \cdot L^{t+1}). \quad (2)$$

---

<sup>2</sup>Often, the memory is initialized with some public value or can be easily reconstructed, once the initializations of the LFSRs are known.

**Fig. 2** A  $(m, \ell)$ -combiner

For cryptographic reasons, we will assume that the state update transformation is bijective. This means that  $L$  is regular and  $\Psi(., X) : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^m$  is bijective for all  $X \in \mathbb{F}_q^\ell$ . A schematic picture of a  $(m, \ell)$ -combiner is given in Fig. 2.

As the outputs  $z_t, \dots, z_{t+r}$  depend only on  $M_t, K_t, \dots, K_{t+r}$ , we define the extended output function  $f_\Psi(M_t, K_t, \dots, K_{t+r}) = (z_t, \dots, z_{t+r})$ . For the sake of simplicity, we use the same notation  $f_\Psi$  for different values of  $r$ .

Observe that the key size of a given  $(m, \ell)$ -combiner can be easily altered by changing  $L$  and  $P$  but keeping the same update and output functions  $\Psi$  and  $f$ . Hence, it is natural to treat the production of the internal stream  $K_0, K_1, \dots$  and the computation of the keystream  $z_0, z_1, \dots$  separately. In particular, one could use other mechanisms to produce the internal stream as for example cellular automata e.g., see Wolfram (1986).

### 3 Algebraic Attacks

The goal of cryptanalysis is the evaluation of cryptographic schemes against several kinds of possible attackers. The first step is to define a concise attacker model which specifies the knowledge, capabilities, and goals of an attacker. An algorithm deployed by an attacker to reach her goal is called an *attack*. In this section, we give a brief description of the work principles of algebraic attacks against stream ciphers. A more comprehensive treatment can be found in Armknecht (2006).

Regarding the knowledge, the usual assumptions follow Kerckhoffs' principle (Kerckhoffs 1883). This means for  $(m, \ell)$ -combiners that an attacker knows both the structure of the combiner itself, including the definitions of the LFSRs, the output function  $f$ , etc., as well as parts of the keystream.

Different definitions of attacker models exist, e.g. Rueppel (1989, 1992), and (Zenner 2004, Chap. 2). For algebraic attacks, one considers attackers that are able to operate on a uniform computational model, like a Turing machine, whose computational behavior is similar to that of a programmable microprocessor. Or, less formally, the attacker has access to a personal computer to perform the computations of his attack. One single operation will be called a basic operation. The efficiency of an attack is measured by (at least) the minimum necessary number of keystream outputs, the number of basic operations, and the amount of memory required for the attack.

A variety of different attacker goals are imaginable. One possibility is, given some keystream elements  $z_0, \dots, z_t$ , to predict the next keystream elements

$z_{t+1}, \dots$ . This would allow the decryption of the whole ciphertext. Another possibility is a distinguishing attack, where the attacker's task is to distinguish the keystream from a truly random stream. In algebraic attacks, one tries to derive the initial states from the known information. Observe that this covers the goals mentioned above.

It has to be pointed out that the initial states and the secret key, shared between sender and receiver, are not necessarily equal. In practice, there exists a separate mechanism which derives the internal states from the common secret key. However, the design and analysis of good derivation mechanisms is a topic on its own. Therefore, we focus on the recovery of the initial state  $S_0 = (M_0, K)$  of the memory and the LFSRs. Usually, the size of the memory register is small compared to the lengths of the LFSR registers. Hence, once  $K$  has been found out,  $M_0$  can easily be reconstructed either by exploiting the structure of the keystream generators or by exhaustively trying all values. Therefore, we concentrate on attacks where the primal goal is to find out the value of  $K$  and refer to  $K$  as the secret key.

Observe that each known keystream part  $z_t, \dots, z_{t+r-1}$  reveals some information on the corresponding inputs  $K_t, \dots, K_{t+r-1}$  and hence on  $K$ . The basic idea of algebraic attacks is to encode this information into equations. This eventually leads to a system of equations with its solution being exactly the value of  $K = (k_1, \dots, k_n)$ . From now on we will assume, if not otherwise stated, that each function is expressed in its algebraic normal form with the degrees reduced to its minimum. This means the computations are done in the ring  $\mathbb{F}_q[k_1, \dots, k_n]/(k_1^q - k_1, \dots, k_n^q - k_n)$ . Furthermore we will use the abbreviation  $K^q - K$  for  $k_1^q - k_1, \dots, k_n^q - k_n$ . That is the ring mentioned above could be equally expressed by  $\mathbb{F}_q[K]/(K^q - K)$ .

Actually, the approach to express a cipher by a system of equations is not new. Already Shannon mentioned in his seminal paper (Shannon 1949) that breaking a good cipher should require “*as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type.*” The reason for this recommendation is that solving systems of nonlinear equations is difficult in general. For example, it has been proven that finding a solution of a system of  $n$  quadratic equations in  $n$  variables is an NP-hard problem (Håstad et al. 1993). This means that probably no polynomial time algorithm exists for solving general systems of non-linear equations over finite fields.

However, in Courtois and Meier (2003), Ars and Faugère (2003) it was pointed out for the first time that in the case of simple combiners, the secret key can be encoded into a system of equations which can be efficiently solved under certain properties. The generation of the system of equation depends on the knowledge of some kind of “local” equation. This idea has been later extended to combiners with memory in Armknecht and Krause (2003). The idea can be described in general as follows. Let a  $(m, \ell)$ -combiner be given with output function  $f$ , memory update function  $\Psi$ , and extended output function  $f_\Psi$ . Furthermore, assume for a fixed value  $r \geq 1$  that a function  $F : \mathbb{F}_q^{r \cdot \ell + r} \rightarrow \mathbb{F}_q$  is known such that it holds for all

$X_1, \dots, X_r \in \mathbb{F}_q^\ell$  and  $y_1, \dots, y_r \in \mathbb{F}_q^o$ :

$$\begin{aligned} \exists M \in \mathbb{F}_q^m : f_M(M, X_1, \dots, X_r) = (y_1, \dots, y_r) &\Rightarrow \\ F(X_1, \dots, X_r, y_1, \dots, y_r) = 0. \end{aligned} \tag{3}$$

Informally said, whenever there exists an assignment to the memory register such that the “LFSR-outputs”  $X_1, \dots, X_r$  lead to the “keystream outputs”  $y_1, \dots, y_r$  then  $F$  gives zero on  $(X_1, \dots, X_r, y_1, \dots, y_r)$ . In the case of simple combiners where no memory register is present, a candidate for  $F$  is  $F(X, y) := f(X) - y$ . However, observe that due to other cryptographic design criteria, e.g., to achieve high linear complexity, the degree of  $f$  will be probably high. This makes this choice rather unsuitable for algebraic attacks, but we will discuss in Sect. 4 different methods for finding better equations.

Once such functions are known, they can be used to setup a system of equations in the key. More precisely, whenever some keystream outputs  $z_t, \dots, z_{t+r-1}$  are known, one can insert the following equation to the system of equations:

$$0 = F(K_t, \dots, K_{t+r-1}, z_t, \dots, z_{t+r-1}). \tag{4}$$

It is important to stress that the more keystream outputs are known, the more equations can be set up using the “local” function  $F$ .

An important property of this kind of system of equations is that the degrees of the equations are bounded by some constant. Recall that  $K_t = K \cdot L^t \cdot P$ , that is the entries of  $K_t$  depend linearly on the key  $K$ . Hence, all equations of the form of (4) have a degree in  $K$  which is upper bound by

$$d := \max \{ \deg_{X_1, \dots, X_r} \underbrace{(F(X_1, \dots, X_r, y_1, \dots, y_r))}_{\in \mathbb{F}_q[X_1, \dots, X_r]} \mid (y_1, \dots, y_r) \in \mathbb{F}_q^{r \cdot o} \}. \tag{5}$$

In particular, only monomials of degree  $d$  or less can occur in the system of equations. Hence, if  $d$  is small, only a fraction of all possible monomials can be part of the system of equations. Let  $\mu$  denote this number where the constant term is not counted. Next assume that the generated system of equations contains  $\mu$  linearly independent equations. Then, the idea of the *linearization* approach is to replace each of the  $\mu$  monomials by a new identifier. By doing so, we transform a non-linear system of equations in  $n$  unknowns of degree  $\leq d$  into a *linear* system of equations in  $\mu$  unknowns where  $\mu$  linearly independent equations are given. This allows to solve the (newly created) system of linear equations with the usual methods from linear algebra, e.g., Gaussian elimination. Once this is done, the solution of the original system of equations can be easily extracted. In the case that Gaussian elimination is used, the time and memory effort for this approach are in  $\mathcal{O}(\mu^3)$  and  $\mathcal{O}(\mu^2)$ , respectively. The effort can be further reduced by using improved algorithms as for example the one from Strassen (1969). The interesting fact is that if  $\mathbb{F} = \mathbb{F}_2$ , being the most important case for practical applications, it holds that  $\mu \in \mathcal{O}(n^d)$  which implies that the attack effort is *polynomial* in  $n$ , i.e. the key size. Observe that (4)

is independent on how the values  $K_t$  are generated. This means that even if one increases the key size  $n$  (= the size of the LFSRs), (4) remains valid. Hence,  $d$  is independent of  $n$  and an increase of the key size results only in a polynomial increase in the attack effort. This is quite astonishing as the security should increase exponentially with the key size. For this reason, algebraic attacks outmatched, at least in theory, for several stream ciphers all other previously known attacks, e.g., Courtois and Meier (2003), Armknecht and Krause (2003), Cho and Pieprzyk (2004).

However, when it comes to the practicability of algebraic attacks, things are different. First of all, it might be difficult to find appropriate functions  $F$  as given in (3). Even if one such function is found, it might not necessarily be the best choice. For example, as the efforts for algebraic attacks (data, memory, computation) are all exponential in the value  $d$  specified in (5), an attacker is interested in finding equations with the lowest possible degree.

Another issue is the impractical huge amount of known keystream outputs required for an attack. For example, it was estimated in Armknecht and Krause (2003) that about  $2^{23}$  known keystream bits are necessary for a successful attack against  $E_0$ , being certainly unrealistic. This is even worse if one considers that only about 132 known keystream bits should be enough to determine the secret key from a information theoretical point of view.

In the following sections, we describe how and to what extent Gröbner bases are helpful to tackle these problems.

## 4 Finding Equations

As mentioned in the previous section, the effort of algebraic attacks depends both on the number of unknowns in the system of equations and on the degree of the equations. We discuss for different scenarios how Gröbner bases can be used to determine optimal equations in terms of degree and/or variables over few clocks.

### 4.1 Simple Combiners

First, we focus on simpler combiners, that is  $(0, \ell)$ -combiners with no memory register. Let  $f : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q^o$  be the output function with input variables  $x_1, \dots, x_\ell$  and outputs  $y_1, \dots, y_o$ . W.l.o.g., we can rewrite  $f$  as  $f = (f_1, \dots, f_o)$  with  $y_i = f_i(x_1, \dots, x_\ell)$ . A theoretical analysis of the existence of low degree equations has first been conducted in Meier et al. (2004) for the case of  $\mathbb{F}_q = \mathbb{F}_2$  and  $o = 1$ . The authors introduced the notion of the *algebraic immunity*  $AI(f)$  of a Boolean function  $f$  and showed that it is equal to the lowest possible degree of equations that can be used for an algebraic attack. That is, the value  $AI(f)$  indicates the resistance of the simple combiner against algebraic attacks.  $AI(f)$  is defined as follows where  $h \equiv 0$  means that a function  $h$  is the constant, all-zero function:

$$AI(f) := \min\{\deg(g) \mid g \not\equiv 0, g \cdot f \equiv 0 \text{ or } g \cdot (f - 1) \equiv 0\}. \quad (6)$$

This notion has later been extended to combiners with memory and block ciphers in Armknecht (2005). Several methods have been developed to compute  $AI(f)$ , e.g., in Meier et al. (2004), Armknecht (2004b), Armknecht et al. (2006), Didier and Tillich (2006). Before we proceed, we give a generalized definition of the algebraic immunity:

**Definition 3** Consider a function  $f : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q^o$ ,  $X = (x_1, \dots, x_\ell) \mapsto Y = (y_1, \dots, y_o)$  and the associated ideal of relations:

$$\mathcal{I} = \langle y_1 - f_1(X), \dots, y_o - f_o(X), X^q - X, Y^q - Y \rangle. \quad (7)$$

Observe that  $\mathcal{I}$  contains all relations between the inputs and outputs of  $f$ . The algebraic immunity of  $f$  is defined by

$$AI(f) := \min\{\deg_X(g) \mid g \in \mathcal{I} \setminus \{0\}\}. \quad (8)$$

The following proposition shows that this definition is indeed an extension of the original notion of algebraic immunity:

**Proposition 1** (Ars 2005) *For the case of  $o = 1$  and  $\mathbb{F}_q = \mathbb{F}_2$ , the definition of algebraic immunity from Definition 3 equals the original definition (6) of Meier et al. (2004).*

*Proof* First, we recall that any ideal over a finite field that contains the field equations is radical (Seidenberg 1974). Now let

$$AI_1(f) := \min\{\deg(g) \mid g \not\equiv 0, g \cdot f \equiv 0 \text{ or } g \cdot (f - 1) \equiv 0\} \quad \text{and} \quad (9)$$

$$AI_2(f) := \min\{\deg_X(g) \mid g \in \mathcal{I} \setminus \{0\}\}, \quad (10)$$

with  $\mathcal{I} := \langle f - y, y^2 - y, X^2 - X \rangle$ . We will show that  $AI_1(f) = AI_2(f)$ .

Let  $g_1 = g_1(X)$  be a Boolean function such that  $f \cdot g_1 \equiv 0$ . Hence,  $(f - 1) \cdot g_1 \equiv g_1$  and  $g_1 \in \mathcal{I}_1 := \langle f - 1, X^2 - X \rangle$  as  $\mathcal{I}_1$  is a radical ideal. In particular, one can express  $g_1$  by  $g_1 = P \cdot (f - 1) + \sum_{i=1}^n Q_i \cdot (x_i^2 - x_i)$  where  $P$  and  $Q_i$  are some polynomials. From this, it follows that

$$y \cdot g_1 = y \cdot P \cdot (f - y) + P \cdot (y^2 - y) + y \cdot \sum_{i=1}^n Q_i \cdot (x_i^2 - x_i) \in \mathcal{I}.$$

In a similar manner, one can show for any  $g'_1$  with  $g'_1 \cdot (f - 1) \equiv 0$  that  $(y - 1) \cdot g'_1 \in \mathcal{I}$ . This implies that  $AI_1(f) \geq AI_2(f)$ .

Let  $g_2 \in \mathcal{I}$ . We can write  $g_2$  as  $g_2(X, y) = P \cdot (f - y) + Q \cdot (y^2 - y) + \sum_{i=1}^n Q_i \cdot (x_i^2 - x_i)$  for some polynomials  $P$ ,  $Q$ , and  $Q_i$ . It follows that

$$g_2(X, 0) \cdot (f - 1) = \tilde{P} \cdot (f^2 - f) + \sum_{i=1}^n \tilde{Q}_i \cdot (f - 1) \cdot (x_i^2 - x_i).$$

Using the Frobenius relation ( $h^2 \equiv h$  on  $\mathbb{F}_2$ ), one has  $f^2 - f \equiv 0$  and  $x_i^2 - x_i \equiv 0$ , what implies  $g_2(X, 0) \cdot (f - 1) \equiv 0$ . Similarly, one can prove that  $g_2(X, 1) \cdot f \equiv 0$ . Because of  $\deg_X(g_2(X, y)) \geq \deg_X(g_2(X, 0))$  and  $\deg_X(g_2(X, y)) \geq \deg_X(g_2(X, 1))$ , one concludes that  $AI_2(f) \geq AI_1(f)$ .  $\square$

This extended definition allows to determine the algebraic immunity by computing a Gröbner basis of  $\mathcal{I}$  for an appropriate block ordering. Recall that for two ordered sets  $X$  and  $Y$  of variables and two orderings  $<_X$  and  $<_Y$  on  $\mathbb{F}[X]$  and  $\mathbb{F}[Y]$ , respectively, the block ordering  $<_{X,Y}$  on  $\mathbb{F}[X, Y]$  is defined by

$$X^\alpha Y^\beta <_{X,Y} X^{\alpha'} Y^{\beta'} \iff X^\alpha <_X X^{\alpha'} \text{ or } (X^\alpha = X^{\alpha'} \text{ and } Y^\beta <_Y Y^{\beta'}).$$

**Theorem 1** (Ars and Faugère 2005) *Consider an output function  $f : \mathbb{F}^\ell \rightarrow \mathbb{F}^o$ . A reduced Gröbner basis of  $\mathcal{I}$  for an elimination order on  $[X], [Y]$  contains a linear basis of polynomials  $g \in \mathcal{I}$  such that  $AI(f) = \deg_X(g)$ .*

Next, we recall several bounds on the algebraic immunity. For  $f : \mathbb{F}_2^\ell \rightarrow \mathbb{F}_2$ , it holds that  $AI(f) \leq \lceil \frac{\ell}{2} \rceil$  (Courtois and Meier 2003; Meier et al. 2004). For more general statements, let  $f$  and  $\mathcal{I}$  be as defined in Definition 3 and let  $\mathcal{R}$  be the ring  $\mathbb{F}_q[x_1, \dots, x_\ell, y_1, \dots, y_o]/\mathcal{I}$ . As  $\mathcal{I}$  is a zero dimensional ideal, the ring  $\mathcal{R}$  is a linear vector space with a finite dimension. If choosing the lexicographic order  $y_1 > \dots > y_o > x_1 > \dots > x_\ell$  for  $\mathcal{I}$ , the set  $\{X^\alpha \mid \alpha \in \mathbb{F}_q^\ell\}$  forms a basis of  $\mathcal{R}$ . This shows that  $\mathcal{R}$  has the dimension  $q^\ell$ . We deduce that the images of any  $q^\ell + 1$  monomials in  $\mathcal{R}$  are linearly dependent. In particular, there exists a linear relation between the first  $q^\ell + 1$  monomials and this relation corresponds to a polynomial in  $\mathcal{I}$ . Let  $M_\ell^d$  denote the number of monomials in  $\ell$  unknowns over  $\mathbb{F}_q$  which have a degree of  $d$ . Then, the number of monomials in  $\mathbb{F}_q[X, Y]/(X^q - X, Y^q - Y)$  with degree  $d$  in  $x_1, \dots, x_\ell$  is  $q^o M_\ell^d$  where  $q^o$  is the number of all possible monomials in the unknowns  $Y$ . Thus the algebraic immunity is upper bounded by the minimum value  $d$  such that the following holds:

$$\sum_{k=0}^d q^o M_\ell^k \geq q^\ell \iff 0 \geq q^{\ell-o} - \sum_{k=0}^d M_\ell^k. \quad (11)$$

For determining the smallest  $d$  which fulfills this inequation, one can make use of the fact that

$$\frac{q^{\ell-o}}{1-t} - \frac{(1-t^q)^\ell}{(1-t)^{\ell+1}} = \sum_{d \geq 0} \left( q^{\ell-o} - \sum_{k=0}^d M_\ell^k \right) t^d.$$

Hence, an upper bound for the algebraic immunity can be derived by developing the expression on the left hand side by  $t$  and determining the first negative or zero coefficient of this series.

Another question is the distribution of the algebraic immunity in the set of all possible functions, especially the percentage of functions that have the maximum

**Table 1** Distribution of  $AI(f)$  over all functions  $f : \mathbb{F}_2^\ell \rightarrow \mathbb{F}_2$

$AI(f)$	1	2	3
Balanced $f$	$2.9 \times 10^{-4}\%$	67.1%	32.9%
Unbalanced $f$	0.3%	99.7%	0%
Total	0.2%	90.6%	9.2%

algebraic immunity. Table 1 displays the distribution of  $AI(f)$  over all Boolean functions from  $\mathbb{F}_2^\ell$  to  $\mathbb{F}_2$ . Recall that  $1 \leq AI(f) \leq \lceil \frac{\ell}{2} \rceil = 3$ . The table shows that almost a third of all balanced functions have the maximum algebraic immunity of 3. Moreover, only balanced functions reach the maximum algebraic immunity.

An asymptotic bound for the algebraic immunity has been proven in Meier et al. (2004):

**Proposition 2** *Let  $f : \mathbb{F}_2^\ell \rightarrow \mathbb{F}_2$  be a balanced Boolean function. Then it holds that*

$$\Pr(AI(f) \leq d) \leq 2 \frac{2^{1+\ell+\dots+(d-1)} (2^{\binom{\ell}{d}} - 1) \left( \frac{2^\ell - 2^{\ell-d}}{2^{\ell-1} - 2^{\ell-d}} \right)}{\binom{2^\ell}{2^{\ell-1}}} \quad (12)$$

where  $\Pr(AI(f) \leq d)$  is the probability that  $AI(f) \leq d$ .

This yields the following theorem:

**Theorem 2** (Meier et al. 2004) *Let  $f : \mathbb{F}_2^\ell \rightarrow \mathbb{F}_2$  and  $(d_\ell)$  be a sequence that satisfies  $d_\ell \leq \mu\ell$  where  $\mu = \frac{1}{2}(1 + \frac{\ln 2}{2} - \sqrt{(1 + \frac{\ln 2}{2})^2 - 1}) \approx 0.22$ , then it holds that  $\lim_{\ell \rightarrow +\infty} \Pr(AI(f) \leq d_\ell) = 0$ .*

According to this theorem, almost all functions  $f : \mathbb{F}_2^\ell \rightarrow \mathbb{F}_2$  have asymptotically an algebraic immunity that is higher than  $0.22\ell$ . The theorem confirms the observations displayed in Table 1.

## 4.2 Combiners with Memory

Observe that the results stated so far concern only simple combiners. For a combiner with memory, that is, with a memory register of length  $m > 0$ , the keystream is computed from both the outputs of the LFSRs and from (parts of) the memory register. Hence, the degree of the equations depend both on the output function  $f$  and the next memory state function  $\Psi$ .

Before we consider the question on the minimum possible degree, we have to explain what type of equations are advantageous for algebraic attacks. A straightforward approach to generate equations would be to express equations in the inputs  $K_t$ ,

the keystream  $z_t$ , and some memory variables. In principle there are two different possibilities how to integrate the memory contents into the equations: either introducing new variables for each clock or expressing them by other variables. In the first case, the system of equations will be most likely unsolvable as the number of unknowns increases with the number of equations. Regarding the second case, observe that as opposed to the inputs  $K_t$ , which are linearly derived from  $K$ , the relation between the initial memory state  $M_0$  and the memory state  $M_t$  at clock  $t$  is non-linear. Hence, the degree of the equations would no longer be bounded by a constant  $d$  so that a polynomial effort for computing the solution is not guaranteed anymore. Therefore, the usual approach is to find equations which are *independent* of the memory variables in the system of equations. In Armknecht and Krause (2003), the authors proved for  $\mathbb{F} = \mathbb{F}_2$  and  $o = 1$  that if one considers  $m + 1$  successive outputs, there always exists a non trivial relation between the keystream outputs and the LFSR inputs  $K_t$ . An extension to general combiners with memory is the following:

**Theorem 3** (Ars and Faugère 2005) *Consider a combiner with memory as described in Definition 2 with output function  $f = (f_1, \dots, f_o)$  and a next memory state function  $\Psi = (\Psi_1, \dots, \Psi_m)$ . For  $r \geq 1$ , we define the ideal  $\mathcal{I}_r$  generated by all equations between the unknowns involved in generating  $r$  outputs. More precisely, let at clock  $t$  denote  $X_t = (x_{t,1}, \dots, x_{t,\ell})$  the inputs,  $M_t = (M_{t,1}, \dots, M_{t,m})$  the content of the memory register, and  $Y_t = (y_{t,1}, \dots, y_{t,o})$  the keystream output. Then  $\mathcal{I}_r$  is generated by*

$$\begin{cases} y_{t,j} - f_j(X_t, M_t), & j = 1, \dots, o, \quad t = 1, \dots, r, \\ M_{t+1,i} - \Psi_i(X_t, M_t), & i = 1, \dots, m, \quad t = 1, \dots, r-1 \end{cases}$$

and the field equations  $X_t^q - X_t$ , etc. over all variables. If  $r \geq \lceil \frac{m+1}{o} \rceil$ , then there exists a non-zero polynomial  $F \in \mathcal{I}_r$  which is independent of the memory values.

Summing up, a relation between the inputs  $K_t$  and keystream outputs  $z_t$  exists for sure if one considers sufficiently many successive clocks. From these, one can generate system of equations as described in Sect. 3. Notice that the degree of these equations are likewise bounded by some constant, making the linearization approach still possible. Moreover like in the previous section, there is a Gröbner basis method to compute equations with minimal degree:

**Proposition 3** (Ars 2005) *Consider a combiner with memory and the ideal  $\mathcal{I}_r$  as defined in Theorem 3. A reduced Gröbner basis of  $\mathcal{I}_r$  for an elimination order on  $[M_i][X_j], [Y_k]$  contains a linear basis of polynomials  $g \in \mathcal{I}_r$  with minimal degree in  $X_1, \dots, X_r$  and without any memory variables  $M_{t,i}$ .*

Actually, this result shows that the notion of algebraic immunity can be easily extended to combiners with memory. In particular, some results on the algebraic immunity of simple combiners can be transferred to the case of combiners with

memory by a simple change of variables. For example, similarly to the approach described in Sect. 4.2, one can derive an upper bound on the minimal degree of the equations by determining the first degree of a series which has a negative or zero coefficient. This series is  $\frac{q^{(\ell-o)r+m}}{1-t} - \frac{(1-t^q)^{\ell r}}{(1-t)^{\ell r+1}}$ .

A problem with this approach is that for a large number of variables, the computation of a Gröbner basis can be elaborate. The following theorem (Ars 2005) shows that this effort can be reduced in some cases if generators of the ideal  $\mathcal{J}_r := \mathcal{I}_r \cap \mathbb{F}[X_1, \dots, X_r, Y_1, \dots, Y_r]$  are known:

**Theorem 4** (Ars 2005) *Consider a combiner with memory with output function  $f : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q$  and with memory of size  $m$ . If*

$$\mathcal{J}_m = \mathcal{I}_r \cap \mathbb{F}[X_1, \dots, X_m, y_1, \dots, y_m] = \{0\},$$

*then there exists a non-trivial polynomial  $F \in \mathbb{F}_q[X_1, \dots, X_{m+1}, y_1, \dots, y_{m+1}]$  so that*

$$y_{m+1} = F(X_1, \dots, X_{m+1}, y_1, \dots, y_m). \quad (13)$$

*The ideals  $\mathcal{J}_r$ ,  $r \geq m$ , are generated by*

$$\{y_{t+m} - F(X_t, \dots, X_{t+m-1}, y_t, \dots, y_{t+m-1}) \mid t \in \{1, \dots, r-m\}\}$$

*and the field equations in these variables.*

As an example, this theorem can be applied to the summation generator (Rueppel 1985). In Lee et al. (2004), the authors found a polynomial  $F$  as described in (13). Using this polynomial, one can compute the minimal degree of relations independent of any memory variables. Below, we give a comparison (in dependence of the key size  $n$ ) of the minimal degree and the bounds derived in Lee et al. (2004):

$n$	2	3	4	5	6	7	8	9
Bound of Lee et al. (2004)	2	3	4	6	6	7	8	12
Minimal degree $d$	2	3	4	5	6	7	8	9

### 4.3 Considering Several Equations Simultaneously

So far, the focus was on directly deriving optimum equations over few clocks. In the following, we show how the *combination* of several equations can lead to even better results.

**Table 2** The average minimum degree when combining  $s$  appropriate functions  $F_t$  (1000 tests)

$s$	8	7	6	5	4	3	$s$	10	9	8	7	6
Few monomials	1.35	2	2.08	2.72	2.9	3	Few monomials	1.8	2.68	2.8	3	3
Small corr. coeff.	1.37	2	2.05	2.75	2.9	3	Small corr. coeff.	1.8	2.72	2.8	3	3
Random	1.45	2	2.13	2.74	2.9	3	1-resilient	1.8	2.56	2.78	3	3
Boolean functions with $\ell = 5$ variables							Random	1.8	2.39	2.77	3	3
							Boolean functions with $\ell = 6$ variables					

### 4.3.1 Reducing the Degree

Although the algebraic immunity gives the lowest possible degree of equations over some successive keystream elements, it is sometimes possible to find valid equations with a degree below this bound by combining many equations to form a new equation. This idea has been used the first time to mount *fast algebraic attacks*, introduced in Courtois (2003) and further improved in Armknecht (2004a), Hawkes and Rose (2004), Armknecht and Ars (2005). The idea is to reduce the degree of the equations by computing appropriate linear combinations of many successive equations. Successive equations means equations as defined in (4) for successive clocks  $t, t+1, \dots$ . However, an enormous number of successive equations is required for a successful attack,<sup>3</sup> making this approach rather interesting from a theoretical point of view. One possibility to extend the idea is to consider *non-linear* combinations of successive equations. Another possible extension is to look for a set of *non-successive* equations which share the same (preferably small) set of variables. Observe that this idea has been studied before for correlation attacks (Canteaut and Filiol 2002).

In Ars (2005), this approach has been explored for algebraic attacks on simple combiners and the minimum possible degree has been analyzed. More precisely, experiments have been conducted on output functions which were taken from the four following categories: (i) polynomials with only few monomials, (ii) polynomials with a small correlation coefficient, (iii) polynomials which are 1-resilient functions, and (iv) purely random polynomials. All considered functions have been chosen such that they have maximum algebraic immunity, that is  $\lceil \frac{\ell}{2} \rceil$ . The results are shown in Table 2. Somewhat surprising, the results were about the same for all categories. This indicates that the resistance against such approaches does not directly depend on the other criteria.

### 4.3.2 Reducing the Number of Variables

Another approach to improve algebraic attacks is to reduce the number of unknowns. In Armknecht and Ars (2005), it was described how fast algebraic attacks

<sup>3</sup>In Armknecht and Ars (2005), it was shown how to reduce this value to the minimum.

can be modified to achieve this. However, this likewise requires a huge amount of successive equations.

Alternatively, one could try to find “local” equations, that is over few clocks, which reduce the number of unknowns  $n$  to some smaller value  $n' < n$ . We illustrate this idea on the Geffe generator, a simple combiner over  $\mathbb{F}_2$  introduced in Geffe (1973). It uses three LFSRs  $L_A, L_B, L_C$  of lengths  $n_a, n_b$ , and  $n_c$ , respectively. Let  $a_t, b_t$ , and  $c_t$  be the outputs of each LFSR at time  $t$ . The output  $z_t$  is defined by  $z_t = a_t + c_t \cdot (a_t + b_t)$ . Thus, an algebraic attacks based on this equation would need to consider  $n_a + n_b + n_c$  variables. But multiplying the mentioned equation by  $c_t$  (and recalling that we are computing over the characteristic 2) gives a new relation  $0 = c_t z_t + c_t b_t$  which is independent of LFSR  $L_A$ . Thus, the second equation could be used to set up a system of equation for an algebraic attacks that aims for recovering the initial states from  $L_B$  and  $L_C$ . This would reduce the number of variables from  $n_a + n_b + n_c$  to  $n_b + n_c$ . Observe that any of the other LFSRs can be eliminated as well by multiplying with  $c_t + 1$  and  $a_t + b_t + 1$ , respectively.

The following theorem describes for general simple combiners if “local” equations exist which are independent of certain variables:

**Theorem 5** (Ars 2005) *Let  $f \in \mathbb{F}_2[x_1, \dots, x_\ell]$  and consider a simple combiner with output function  $f$ . Then, for any  $i \in \{1, \dots, \ell\}$ , there is a non-trivial relation  $g$  in  $\mathcal{I} = \langle f(X) - y, X^2 - X, y^2 - y \rangle$  which is independent of  $x_i$  if and only if the output function  $f$  cannot be written as  $f(X) = \hat{f}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_\ell) + x_i$ .*

Such functions  $g$  can be determined by computing an appropriate Gröbner basis with elimination order  $[x_i], [x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_\ell], [y_1, \dots, y_o]$ . This approach has some connections to the method for computing the algebraic immunity. Using the theory of Gröbner bases, one can prove that the restriction of a Gröbner basis to polynomials without the variable  $x_i$  gives a Gröbner basis in respect to the elimination order  $[x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_\ell], [y_1, \dots, y_o]$ , which in turn is the order used for computing the algebraic immunity. Hence, one has the same properties as in the case of the algebraic immunity. For example, this Gröbner basis contains a linear basis of relations with the lowest possible degree.

## 5 Computing Solutions

In this section, we resume some results on the minimum number of keystream outputs that are required to be known to have a unique solution and some results on the time effort for computing the solutions of the system of equations. We suppose in the following that an attacker has  $N$  linearly independent equations of degree  $d$  in  $n$  variables over  $\mathbb{F}_2$  at her disposal.

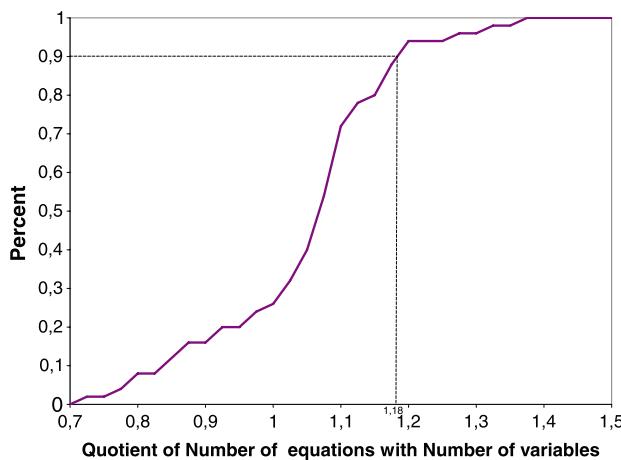
## 5.1 Minimum Number of Outputs

For a successful algebraic attack, it is necessary to know the minimum number of keystream outputs that is required to find a unique solution to the generated system of equations. A very rough upper bound is given by the following proposition:

**Proposition 4** (Ars 2005) *Let a simple combiner over  $\mathbb{F}_q$  be given which is composed of one LFSR of length  $n$  and a balanced output function  $f$ . For any keystream sequence  $(z_i)_{i=0}^{q^n-2}$  there exists at most one initial state which yields this keystream.*

For example, any system of equations over  $\mathbb{F}_2$  with at least  $N \geq 2^n - 1$  equations has one unique solution. Obviously, one is rather interested in a lower bound. Indeed, one can construct pathologic examples with balanced functions  $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$  such that at least  $2^n - 1$  keystream elements need to be known to be able to distinguish the key stream from the all-zero-sequence. However, in most cases one would expect that if the number of equations is close to the number of variables, only one unique solution should be possible.

Figure 3 represents the results of some computer simulations on how many outputs are needed to be known such that the solution is unique. On the horizontal axis, the ratio of the number of given outputs  $N$  by the length of the secret key  $n$  is given whereas the vertical axis shows the percentage of examples where the solution was uniquely determined. According to this graph, it seems that in the majority of the cases, already a little bit more than  $n$  equations provide enough information to determine the initial state. That is, the number of outputs has to be proportional to the length of the initial state to find a unique solution. Note that even if  $N$  is smaller than  $n$ , one can compute the Gröbner basis to get all possible solutions (this is similar to the list decoding concept in error correcting codes). However, when  $N$  is small the



**Fig. 3** Experimental analysis of the minimal number of output bits

**Table 3** Gröbner basis computation for  $N \approx n$  using the F5 algorithm

Example	Size $L$ of register	Number $N$ of outputs	Gröbner computation
$f_5$	40	44	22.3 s
$f_5$	50	54	46.0 s
$f_5$	60	66	89.0 s
$f_5$	70	77	221.0 s

computation of the Gröbner basis is much more difficult and it is not even clear if it can be done in polynomial time.

Very surprisingly, examples have been found, based on output functions given in Canteaut and Filiol (2000), for which the simple combiners can be efficiently attacked even if the number of outputs is small (see Table 3). It is an open issue to understand and to predict such a behavior from the Boolean function and the underlying LFSRs.

## 5.2 Time Effort

Observe that in the worst case, the time effort to compute a Gröbner basis can be double-exponential. However, we will see in this section that the time effort is better predictable if significantly more than  $n$  equations are available. All statements refer to the case that the  $F_5$  algorithm (Faugère 2002) is used. If  $N$  is large enough and  $\mathbb{F}_q = \mathbb{F}_2$ , one can show that during the Gröbner basis computation, none of the computed polynomials will have a degree that exceeds a certain constant  $d$ . From this, one can derive a time effort which is polynomial in the number of variables.

**Theorem 6** (Ars 2005) *If  $\mathbb{F} = \mathbb{F}_2$  and  $N \geq \sum_{i=0}^d \binom{n}{i}$ , where  $d$  is the value specified in (5), then the time effort for the Gröbner basis computation is in  $\mathcal{O}(\binom{n}{d}^\omega) = \mathcal{O}(n^{d\omega})$ , where  $\omega \leq 3$  is the effort for Gaussian elimination.*

This shows that Gröbner basis computation behaves like linearization in the considered cases. The experimental results presented in Table 4 confirm these estimations. The functions  $f_i$  are taken from Canteaut and Filiol (2002).

However, if  $N$  is smaller than the mentioned value, then the degree during the Gröbner basis computation can be higher. In Ars (2005) it was expected to have an effort of about  $n^{(d+1)\omega}$  if the number of outputs  $N$  satisfies the inequation  $\frac{\binom{n}{d+1}}{(n+1)} \leq N \leq \binom{n}{d}$ .

An important point is the comparison between the number of equations that are required for algebraic attacks and fast correlation attacks. For fast correlation attacks, the number of outputs needed is *exponential* in the key size (Canteaut and Filiol 2000) whereas only polynomial for algebraic attacks as discussed before.

**Table 4** Simulations performed on an Alpha DS25 1000 MHz with  $n$  being the number of variables,  $N$  being the number of outputs used, and  $d$  being the degree of the equations

Example	$n$	$N$	$d$	Effort	Time	Example	$n$	$N$	$d$	Effort	Time
$f_1$	40	1071	3	$n^{3\omega}$	18 s	$f_2$	40	3568	3	$n^{3\omega}$	19.3 s
$f_1$	80	8541	3	$n^{3\omega}$	1 h 32 m	$f_2$	70	19076	3	$n^{3\omega}$	32 m 12 s
$f_1$	89	11758	3	$n^{3\omega}$	4 h 28 m	$f_2$	80	28468	3	$n^{3\omega}$	1 h 44 s
$f_3$	80	6342	2	$n^{2\omega}$	1.1 s	$f_2$	89	39190	3	$n^{3\omega}$	4 h 32 m
$f_3$	128	12384	2	$n^{2\omega}$	10.2 s	$f_4$	80	6342	2	$n^{2\omega}$	1.1 s
						$f_4$	128	12384	2	$n^{2\omega}$	10.3 s

We illustrate this on some concrete functions over  $n = 40$  variables, taken from Canteaut and Filiol (2000) (these functions respect several design criteria for stream cipher). In the considered cases, the number of expected outputs for an algebraic attack is significantly less compared to a fast correlation attack. The exact results are below:

Example $n = 40$	$f_1, f_2, f_3$	$f_4, f_5, f_6$	$f_7$	$f_8, f_9, f_{10}$
Gröbner basis	821	412	274	1071
Correlation att.	7625	7625	7625	2725

Despite the results presented so far, it is very difficult in general to bound the effort of solving a system of equations generated for an algebraic attack on a combiner with memory. In Bardet et al. (2005), the authors derived some complexity results for solving “typical” overdetermined algebraic systems over  $\mathbb{F}_2$  using Gröbner bases. Interestingly, the complexity is sub-exponential if the number of equations  $N$  is higher than  $n \ln(n)$ .

## 6 Conclusions

Algebraic attacks consist in generating and solving systems of nonlinear equations over some finite field. While computing solutions is generally a hard problem, the specific structure of the system of equations makes attacks possible with an effort that is polynomial in the key size.

In this paper, we gave an overview on how Gröbner bases can be useful for algebraic attacks. Besides of finding the solutions, Gröbner bases can be helpful for reducing the amount of data and/or for looking for better equations. Therefore, it is a promising direction to further explore the usage of Gröbner bases and to find additional applications. Another open question is to analyze the efficiency of methods based on Gröbner bases to other approaches. For example, besides the methods

discussed in Sects. 4 and 5, alternative approaches exist. Although it is hard to predict the effort for computing a Gröbner basis, it turned out that these methods are often more efficient for practical applications. However, further analysis is required to understand this issue better.

**Acknowledgements** Part of these results have been presented at Linz D1 2006, which was a workshop within the Special Semester on Gröbner Bases, February–July 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

## References

- F. Armknecht, *Improving fast algebraic attacks*, Proc. of FSE 2004, LNCS, vol. **3017**, Springer, Berlin, 2004a, pp. 65–82.
- F. Armknecht, *On the existence of low-degree equations for algebraic attacks*, Cryptology ePrint Archive, Report 2004/185, 2004b, <http://eprint.iacr.org/>.
- F. Armknecht, *Algebraic attacks and annihilators*, Proc. of WEWORC 2005, LNI, vol. **74**, 2005, pp. 13–21.
- F. Armknecht, *Algebraic attacks on certain stream ciphers*, Ph.D. thesis, University Mannheim, Germany, 2006.
- F. Armknecht and G. Ars, *Introducing a new variant of fast algebraic attacks and minimizing their successive data complexity*, Proc. of Mycrypt, LNCS, vol. **3715**, Springer, Berlin, 2005, pp. 16–32.
- F. Armknecht and M. Krause, *Algebraic attacks on combiners with memory*, Proc. of CRYPTO 2003, LNCS, vol. **2729**, 2003, pp. 162–175.
- F. Armknecht, C. Carlet, P. Gaborit, S. Künzli, W. Meier, and O. Ruatta, *Efficient computation of algebraic immunity for algebraic and fast algebraic attacks*, Proc. of Eurocrypt 2006, LNCS, vol. **4004**, 2006, pp. 147–164.
- G. Ars, *Applications of Gröbner Bases to Cryptography*, Ph.D. thesis, University of Rennes I, 2005.
- G. Ars and J. C. Faugère, *An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases*, INRIA Report 4739, 2003, <http://www.inria.fr/rrrt/rr-4739.html>.
- G. Ars and J. C. Faugère, *Algebraic immunities of functions over finite fields*, Tech. report, INRIA, 2005, <ftp://ftp.inria.fr/INRIA/publication>.
- M. Bardet, J. C. Faugere, B. Salvy, and B. Y. Yang, *Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems*, Tech. report, Talk at MEGA 2005, 2005.
- Bluetooth specification v1.1, 1999, <http://www.bluetooth.com/>.
- M. Briceno, I. Goldberg, and D. Wagner, *A pedagogical implementation of A5/1*, 1998, <http://jya.com/a51-pi.htm>.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes Math. **4** (1970), 374–383.
- B. Buchberger, *Gröbner-bases: An algorithmic method in polynomial ideal theory*, Multidimensional systems theory, Reidel, Dordrecht, 1985, pp. 184–232.
- B. Buchberger, *An algorithmical criterion for the solvability of algebraic systems of equations*, London Math. Soc. LNS **251** (1998), 535–545.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- A. Canteaut and E. Filoli, *Ciphertext only reconstruction of stream ciphers based on combination generators*, Proc. of FSE 2000, LNCS, vol. **1978**, Springer, Berlin, 2000, pp. 165–180.

- A. Canteaut and E. Filiol, *On the influence of the filtering function on the performance of fast correlation attacks on filter generators*, Proc. of Symposium on Information Theory 2002, 2002.
- J. Cho and J. Pieprzyk, *Algebraic attacks on SOBER-t32 and SOBER-t116 without stuttering*, Proc. of FSE 2004, LNCS, vol. **3017**, Springer, Berlin, 2004, pp. 49–64.
- N. Courtois, *Fast algebraic attacks on stream ciphers with linear feedback*, Proc. of CRYPTO 2003, LNCS, vol. **2656**, Springer, Berlin, 2003, pp. 176–194.
- N. Courtois and W. Meier, *Algebraic attacks on stream ciphers with linear feedback*, Proc. of EUROCRYPT 2003, LNCS, vol. **2656**, Springer, Berlin, 2003, pp. 345–359.
- F. Didier and J. Tillich, *Computing the algebraic immunity efficiently*, Proc. of FSE 2006, LNCS, vol. **4047**, Springer, Berlin, 2006, pp. 359–374.
- J. C. Faugère, *A new efficient algorithm for computing Gröbner bases without reduction to zero ( $F_5$ )*, Proc. of ISSAC 2002, ACM, New York, 2002, pp. 75–83.
- S. Fluhrer, I. Mantin, and A. Shamir, *Weaknesses in the key scheduling algorithm of RC4*, Proc. of SAC 2001, Springer, Berlin, 2001, pp. 1–24.
- P. Geffe, *How to protect data with ciphers that are really hard to break*, Electronics **46** (1973), no. 1, 99–101.
- J. Håstad, S. Phillips, and S. Safra, *A well-characterized approximation problem*, Inf. Process. Lett. **47** (1993), no. 6, 301–305.
- P. Hawkes and G. Rose, *Rewriting variables: The complexity of fast algebraic attacks on stream ciphers*, Proc. of CRYPTO 2004, LNCS, vol. **3152**, Springer, Berlin, 2004, pp. 390–406.
- A. Kerckhoffs, *La cryptographie militaire*, Journal des Sciences Militaires (1883), 161–191.
- D. Lee, J. Kim, J. Hong, J. Han, and D. Moon, *Algebraic attacks on summation generators*, Proc. of FSE2004, LNCS, vol. **3017**, Springer, Berlin, 2004, pp. 34–48.
- R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press, Cambridge, 1986.
- W. Meier, E. Pasalic, and C. Carlet, *Algebraic attacks and decomposition of Boolean functions*, Proc. of EUROCRYPT 2004, LNCS, vol. **3027**, Springer, Berlin, 2004, pp. 474–491.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- R. Rueppel, *Correlation immunity and the summation generator*, Proc. of CRYPTO 1985, LNCS, vol. **218**, Springer, Berlin, 1985, pp. 260–272.
- R. Rueppel, *Security models and notions for stream ciphers*, Proc. of 2nd IMA Conference on Cryptography and Coding, Oxford University Press, London, 1989, pp. 213–230.
- R. Rueppel, *Stream ciphers*, Contemporary cryptology—The science of information integrity, IEEE Press, 1992, pp. 65–134.
- A. Seidenberg, *Constructions in algebra*, Trans. Amer. Math. Soc. **197** (1974), 273–313.
- C. E. Shannon, *Communication theory of secrecy systems*, Bell System Tech. J. **28** (1949), 656–715.
- V. Strassen, *Gaussian elimination is not optimal*, Numerische Mathematik **13** (1969), 354–356.
- S. Wolfram, *Random sequence generation by cellular automata*, Advances in Applied Mathematics **7** (1986), 123–169.
- E. Zenner, *On cryptographic properties of LFSR-based pseudorandom generators*, Ph.D. thesis, Universität Mannheim, 2004.
- E. Zenner, R. Weis, and S. Lucks, *Sicherheit des GSM-Verschlüsselungsstandards A5*, Datenschutz und Datensicherheit **24** (2000), no. 7, 405–407.

# Canonical Representation of Quasicyclic Codes Using Gröbner Bases Theory

Kristine Lally

**Abstract** The tools and techniques of Gröbner bases theory have proved useful in characterising quasicyclic codes and analysing their algebraic structure. A canonical generating set can be obtained from the reduced Gröbner basis of an associated module structure. The very particular form of this generating set allows straightforward determination of properties such as dimension, in manner directly analogous to the theory developed for cyclic codes.

## 1 Introduction

Quasicyclic (QC) codes of index  $\ell$  and length  $m\ell$  over a finite field  $\mathbb{F}$ , defined by the property that a cyclic shift of a codeword by  $\ell$  places is another codeword, are a natural generalization (Augot et al. 2009) of cyclic codes ( $\ell = 1$ ), and have closely linked algebraic structure. By a coordinate permutation, they are conventionally constructed as the rowspace of a block matrix consisting of some  $t$  rows of  $m \times m$  circulant submatrices. By the usual association of a circulant matrix with the polynomial formed by its top row, they can, in this setting, be regarded as  $\mathbb{F}[x]/I$ -submodules of  $(\mathbb{F}[x]/I)^\ell$ , where  $I = \langle x^m - 1 \rangle$  (Séguin and Drolet 1990).

Most of the literature on QC codes is largely concerned with the 1-generator (that is, cyclic submodule) case, when only  $t = 1$  row of circulants is present in this generator matrix. In this case, dimension can be read from the generating polynomials by the formulae given in Séguin and Drolet (1990), van Tilborg (1978). A large number of QC codes have been found, for example see Gulliver and Bhargava (1991), which achieve the highest known minimum distance of any linear code of the same length and dimension, and many in fact reach the maximum possible value. QC codes are known to be asymptotically good codes, moreover, it was recently showed that double-circulant QC codes meet an improved version of the Gilbert–Varshamov bound (Gaborit and Zemor 2008). Due to their compact representation and efficient encoding algorithm, QC codes are of on-going interest, and have more recently been used to construct good low density parity check (LDPC) codes (Fossorier 2004; Giorgatti et al. 2005).

The theory of Gröbner bases of modules (Mora 2009) has been applied in Lally and Fitzpatrick (1999, 2001), to provide insight into the algebraic structure of an

---

K. Lally

Department of Mathematics, RMIT University, Melbourne, Australia

e-mail: [kristine.lally@rmit.edu.au](mailto:kristine.lally@rmit.edu.au)

arbitrary QC code, where the code is initially specified by any number  $t$  of module generators. A canonical generating set is obtained, and used to determine dimension and also to construct a generator matrix for the code and its dual. Conversely all such canonical generating sets can be specified, allowing the classification and enumeration of all QC codes of index  $\ell$  and length  $m\ell$ , for all dimensions permissible by the degrees of the irreducible factors of  $x^m - 1$ .

## 2 Characterisation Using Gröbner Bases Theory

Let  $\mathcal{C}$  be a QC code of length  $\ell m$  and index  $\ell$ , generated as an  $\mathbb{F}[x]/I$ -submodule of  $(\mathbb{F}[x]/I)^\ell$ , by some arbitrary set of vectors  $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_t\}$  in  $(\mathbb{F}[x]/I)^\ell$ . The code  $\mathcal{C}$  is the image of an  $\mathbb{F}[x]$ -submodule  $\tilde{\mathcal{C}}$  of  $\mathbb{F}[x]^\ell$  containing  $\tilde{\mathcal{K}} = \langle (x^m - 1)\mathbf{e}_i, i = 1, \dots, \ell \rangle$  (where  $\mathbf{e}_i$  is the standard basis vector with 1 in  $i$ -th position and 0 elsewhere), under the natural homomorphism

$$\varphi: \quad \mathbb{F}[x]^\ell \rightarrow (\mathbb{F}[x]/I)^\ell, \quad (c_1, \dots, c_\ell) \mapsto (c_1 + I, \dots, c_\ell + I).$$

Since  $\tilde{\mathcal{C}}$  is a submodule of the finitely generated free module over the principal ideal domain  $\mathbb{F}[x]$  and contains  $\tilde{\mathcal{K}}$ , it is finitely generated by the set  $\tilde{\mathcal{M}} = \{\mathbf{a}_i, i = 1, \dots, t, (x^m - 1)\mathbf{e}_j, j = 1, \dots, \ell\}$  in  $\mathbb{F}[x]^\ell$ . Using the position-over-term (POT) order in  $\mathbb{F}[x]^\ell$ , where  $\mathbf{e}_1 > \mathbf{e}_2 > \dots > \mathbf{e}_\ell$ , and the terms  $x^i$  are ordered naturally in each component, the reduced Gröbner basis (RGB) of  $\tilde{\mathcal{C}}$  can be easily found (in this 1-variable module setting, by simply performing successive elementary row operations on the matrix with rows consisting of the set  $\tilde{\mathcal{M}}$ ) to be an upper triangular generating set  $\tilde{\mathcal{G}} = \{\mathbf{g}_1, \dots, \mathbf{g}_\ell\}$  containing exactly  $\ell$  vectors.

**Theorem 1** *A submodule  $\tilde{\mathcal{C}}$  of  $\mathbb{F}[x]^\ell$  containing  $\tilde{\mathcal{K}}$  (and thus corresponding to a QC code) has a reduced Gröbner basis (with respect to POT monomial ordering) of the form*

$$\tilde{\mathcal{G}} = \{\mathbf{g}_i = (g_{i1}, g_{i2}, \dots, g_{i\ell}), i = 1, \dots, \ell\} \tag{1}$$

where

- (i)  $g_{ij} = 0$  for all  $j < i$
- (ii) the diagonal components  $g_{ii}$  is a non-zero monic polynomial
- (iii)  $\partial g_{ki} < \partial g_{ii}$  for  $k < i$
- (iv) if the left-most non-zero component of an element of  $\tilde{\mathcal{C}}$  lies in the  $i$ -th place then it is divisible by  $g_{ii}$ ; in particular,  $g_{ii}$  divides  $x^m - 1$
- (v) if  $g_{ii} = x^m - 1$  then  $\mathbf{g}_i = (x^m - 1)\mathbf{e}_i$
- (vi) the  $\mathbb{F}$ -dimension of  $\mathbb{F}[x]^\ell/\tilde{\mathcal{C}}$  is  $\sum_{i=1}^{\ell} \deg g_{ii}$ , that is, the number of monomials  $x^s \mathbf{e}_i$  in  $\mathbb{F}[x]^\ell$  in normal form modulo  $\tilde{\mathcal{G}}$ .

Conversely any such upper triangular set of vectors  $\tilde{\mathcal{G}} = \{\mathbf{g}_i, i = 1, \dots, \ell\}$ , satisfying the monic and degree restrictions of (i)–(iii) above, is the unique RGB of

a submodule  $\tilde{\mathcal{C}}$  containing  $\tilde{\mathcal{K}}$  (and thus corresponds to a QC code) if there exists a matrix  $\tilde{A} \in \mathbb{M}_\ell(\mathbb{F}[x])$  satisfying  $\tilde{A}\tilde{G} = \tilde{G}\tilde{A} = (x^m - 1)I$ , where  $\tilde{G}$  is the upper triangular matrix with rows  $\mathbf{g}_i$ , and  $I$  is the identity matrix. It is immediate that  $\tilde{A}$  is also upper triangular, and the non-zero entries of  $\tilde{A}$  can be computed recursively from those of  $\tilde{G}$ . The entries of  $\tilde{G}$  satisfy an analogous system of equations in terms of those of  $\tilde{A}$ . This leads to a complete characterisation of all possible RGB for submodules of  $\mathbb{F}[x]^\ell$  corresponding to QC codes.

**Theorem 2** *The upper triangular set  $\tilde{\mathcal{G}}$  is a Gröbner basis of a submodule in  $\mathbb{F}[x]^\ell$  containing  $\tilde{\mathcal{K}}$  if and only if there exist  $a_{ij}$  for  $1 \leq i, j \leq \ell$  satisfying*

$$a_{ij} = \begin{cases} 0 & \text{if } j < i \\ \frac{x^m - 1}{g_{ii}} & \text{if } j = i \\ \frac{-1}{g_{jj}} (\sum_{k=i}^{j-1} a_{ik} g_{kj}) & \text{if } j > i. \end{cases} \quad (2)$$

The Gröbner basis is reduced if and only if  $\partial g_{ii} > \partial g_{ji}$  for all  $j < i$ , if and only if  $\partial a_{ii} > \partial a_{ij}$  for all  $j > i$ .

The QC code  $\mathcal{C}$  is the image of  $\tilde{\mathcal{C}}$  under  $\varphi$ . Dropping the coset notation, it follows that the set  $\mathcal{G}$  consisting of the elements of  $\tilde{\mathcal{G}}$  not mapped to zero under  $\varphi$  forms a  $\mathbb{F}[x]/I$ -generating set for the code  $\mathcal{C}$ . This image set  $\mathcal{G}$  is unique (with respect to the chosen monomial ordering) and is referred to as the RGB generating set of  $\mathcal{C}$ . The dimension of the code  $\mathcal{C}$  can be obtained directly from the diagonal elements of this canonical generating set.

**Theorem 3** *The dimension of the code  $\mathcal{C} \cong \tilde{\mathcal{C}}/\tilde{\mathcal{K}}$  with RGB generating set  $\mathcal{G} = \{\varphi(\mathbf{g}_i), i = 1, \dots, \ell\}$  is given by*

$$\ell m - \sum_{i=1}^{\ell} \partial g_{ii} = \sum_{i=1}^{\ell} (m - \partial g_{ii}).$$

The possible dimensions of QC codes can now be enumerated straightforwardly. Fixing the notation  $x^m - 1 = \prod_{n=1}^s f_n^\varepsilon$ , where  $m = (\text{char } \mathbb{F})^t m'$  with  $\gcd(m', \text{char } \mathbb{F}) = 1$  and  $\varepsilon = (\text{char } \mathbb{F})^t$ , for the decomposition of  $x^m - 1$  into irreducible factors  $f_n$  over  $\mathbb{F}$ .

**Corollary 1** *The QC codes of length  $\ell m$  and index  $\ell$  have dimensions  $\sum_{i=1}^{\ell} \sum_{n=1}^s t_{ni} \partial f_n$  where  $0 \leq t_{ni} \leq \varepsilon$ . Every such dimension arises in some code (for instance, in a code with block diagonal generator matrix).*

A generator matrix for the code  $\mathcal{C}$ , with linearly independent rows, can be constructed from the RGB generating set  $\mathcal{G} = \{\varphi(\mathbf{g}_i), i = 1, \dots, \ell\}$ , as an  $\ell \times \ell$  block upper triangular matrix consisting of rows of truncated  $m \times m$  circulants; the circulants in the  $i$ th row are formed from components of the vector  $\varphi(\mathbf{g}_i)$ , and truncated after the  $(m - \deg(g_{ii}) - 1)$ th cyclic shift in each component.

### 3 Parity Check Matrix and Dual Code

The rows of the matrix  $\tilde{A} \in \mathbb{M}_\ell(\mathbb{F}[x])$  determined in Theorem 2 can be used to form a parity check matrix for the code  $\mathcal{C}$ , and moreover, a minimal GB generating set for the dual code  $\mathcal{C}^\perp$ . The structure of  $\tilde{A}^T$  implies that its rows form a reduced Gröbner basis for the module they generate, with respect to the *reverse* POT term order (rPOT), that is, where  $\mathbf{e}_1 < \mathbf{e}_2 < \dots < \mathbf{e}_\ell$ , and that this module contains  $\tilde{\mathcal{K}}$ . Let  $\hat{f}$  denote the *transpose* of the polynomial  $f$ , that is, the top row polynomial of the transpose of the circulant matrix specified by  $f$ . It is easily seen that  $\hat{f}(x) = x^m f(x^{-1}) \bmod x^m - 1 = x^{m-\partial f} f^* \bmod x^m - 1$ , where  $f^*$  is the conventional reciprocal of  $f$ .

**Theorem 4** *The rows  $\tilde{\mathcal{H}} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_\ell\} \subseteq (\mathbb{F}[x])^\ell$  of the matrix*

$$\tilde{H} = \begin{pmatrix} a_{11}^* & 0 & \cdots & 0 \\ x^{\partial a_{22}} \hat{a}_{12} & a_{22}^* & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x^{\partial a_{\ell\ell}} \hat{a}_{1\ell} & x^{\partial a_{\ell\ell}} \hat{a}_{2\ell} & \cdots & a_{\ell\ell}^* \end{pmatrix}$$

*form an rPOT minimal Gröbner basis (usually not reduced) for the preimage  $\tilde{\mathcal{C}}^\perp$  of the dual code  $\mathcal{C}^\perp$  in  $\mathbb{F}[x]^\ell$ .*

The image set  $\mathcal{H} = \{\varphi(\mathbf{h}_i), i = 1, \dots, \ell\}$  forms a minimal GB generating set for the dual code  $\mathcal{C}^\perp$ . A block lower triangular matrix generator matrix for  $\mathcal{C}^\perp$ , and thus parity check matrix for  $\mathcal{C}$ , is formed by replacing each component of  $\varphi(\mathbf{h}_i)$  by the  $m \times m$  circulant it generates. Redundant rows can be removed by truncating all but the first  $m - \partial a_{ii}$  rows in each block row.

A change of monomial ordering, to form the POT RGB for the preimage module  $\tilde{\mathcal{C}}^\perp$  in  $\mathbb{F}[x]^\ell$ , can be achieved by appropriate  $\mathbb{F}[x]$ -elementary row operations on  $\tilde{H}$ . Comparison of the resulting generators, to the elements of  $\tilde{\mathcal{G}}$ , have lead, in the  $\ell = 2$  case, to a complete characterisation of all self-dual QC code of index 2 (Lally and Fitzpatrick 2001).

### 4 Recent Application to QC LDPC Codes

Parity check matrices comprising of low-weight circulant submatrices have been employed by many authors to define LDPC codes. For example, in Fossorier (2004) circulant permutation (1-weight) matrices are strategically arranged to achieve high girth. It is known that such matrices (and more generally those with a uniform circulant-weight configuration) cannot reach fullrank, and moreover, their actual rank is difficult to determine. A class of regular QC parity check matrices consisting

of 0-, 1- and 2-weight circulants, arranged in a predefined pattern, has been proposed more recently in Giorgetti et al. (2005). It was shown that the POT RGB of the dual code could be easily constructed by formula from the polynomial parameters, and hence the dimension determined straightforwardly as described earlier. Parity check matrices in this class were shown by this method to be fullrank, leading to the construction of a class of QC LDPC codes of rate  $(\ell - 1)/\ell$ .

*Remark* The “module approach” is used for some convolutional codes (Gluesing-Luerssen et al. 2009).

## References

- D. Augot, E. Betti and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- M. P. C. Fossorier, *Quasi-cyclic low-density parity-check codes from circulant permutation matrices*, IEEE Trans. on Inf. Th. **50** (2004), no. 8, 1788–1793.
- P. Gaborit and G. Zemor, *Asymptotic improvement of the Gilbert-Varshamov bound for linear codes*, IEEE Trans. on Inf. Th. **54** (2008), no. 9, 3865–3872.
- M. Giorgetti, M. Rossi and M. Sala, *On the Gröbner basis of a family of quasi-cyclic LDPC codes*, Bull. Iran. Math. Soc. **31** (2005), no. 2, 13–32.
- H. Gluesing-Luerssen, B. Langfeld and W. Schmale, *A short introduction to cyclic convolutional codes*, this volume, 2009, pp. 403–408.
- T. A. Gulliver and V. K. Bhargava, *Some best rate  $1/p$  and rate  $(p - 1)/p$  systematic quasi-cyclic codes*, IEEE Trans. on Inf. Th. **37** (1991), no. 3, part 1, 552–555.
- K. Lally and P. Fitzpatrick, *Construction and classification of quasicyclic codes*, Proc. of WCC 1999 (1999), 11–22.
- K. Lally and P. Fitzpatrick, *Algebraic structure of quasicyclic codes*, Discrete Appl. Math. **111** (2001), nos. 1–2, 157–175.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- G. E. Séguin and G. Drolet, *The theory of 1-generator quasi-cyclic codes*, Tech. report, Royal Military College of Canada, Kingston, Ontario, 1990.
- H. C. A. van Tilborg, *On quasi-cyclic codes with rate  $1/m$* , IEEE Trans. on Inf. Th. **24** (1978), no. 5, 628–630.

# About the $n$ th-Root Codes: a Gröbner Basis Approach to the Weight Computation

Marta Giorgetti

**Abstract** Recently some methods have been proposed to find the distance and weight distribution of cyclic codes using Gröbner bases (Sala in Appl. Algebra Engrg. Comm. Comput. 13(2):137–162, 2002; Mora and Sala in J. Symbolic Comput. 35(2):177–194, 2003). We identify a class of codes for which these methods can be generalized. We show that this class contains all interesting linear codes (i.e., with  $d \geq 2$ ) and we provide variants and improvements.

## 1 General $n$ th-Root Codes

We denote by  $\mathbb{F}_q$  the finite field with  $q$  elements,  $q$  is a power of a prime, and by  $n$  a natural number such that  $(q, n) = 1$ . Let  $k, N \in \mathbb{N}$  such that  $1 \leq k \leq N \leq n + 1$ . We refer to the vector space of dimension  $N$  over  $\mathbb{F}_q$  as to  $(\mathbb{F}_q)^N$ . The zeros of polynomial  $x^n - 1$ , which are called  $n$ -th roots of unity, lie in an extension field  $\mathbb{F}_{q^m}$  and in no smaller field. We denote the set of all these roots by  $R_n$ . From now on,  $q, n, k, N$  and  $m$  are understood. Our notation on linear and cyclic codes follows (Augot et al. 2009). All the following statements and definitions can be found in Giorgetti and Sala (2006), Giorgetti (2006), Giorgetti and Sala (2009).

**Definition 1** Let  $L$  be a subset of  $R_n \cup \{0\}$ ,  $L = \{l_1, \dots, l_N\}$  and  $\mathcal{P} = \{g_1(x), \dots, g_r(x)\}$  in  $\mathbb{F}_{q^m}[x]$  such that  $\forall i = 1, \dots, N$  there is at least one  $j = 1, \dots, r$  such that  $g_j(l_i) \neq 0$ . We denote by  $C = \mathcal{Q}(q, n, q^m, L, \mathcal{P})$  the linear code defined over  $\mathbb{F}_q$  having

$$H = \begin{pmatrix} g_1(l_1) & \dots & g_1(l_N) \\ g_2(l_1) & \dots & g_2(l_N) \\ \vdots & & \vdots \\ g_r(l_1) & \dots & g_r(l_N) \end{pmatrix} = \begin{pmatrix} g_1(L) \\ g_2(L) \\ \vdots \\ g_r(L) \end{pmatrix}$$

as its parity-check matrix. We say that  $C$  is an  $n$ th-root code.

**Remark 1** Code  $C = \mathcal{Q}(q, n, q^m, L, \mathcal{P})$  is linear over  $\mathbb{F}_q$ , its length is  $N = |L|$  and its distance  $d$  is greater than or equal to 2, because there are no columns in  $H$  composed only of zeros.

---

M. Giorgetti

Department of Mathematics, University of Milano, Milan, Italy

e-mail: [giorge@mat.unimi.it](mailto:giorge@mat.unimi.it)

**Definition 2** Let  $C = \Omega(q, n, q^m, L, \mathcal{P})$  be an  $n$ th-root code.

If  $\bar{L} = R_n \setminus L = \emptyset$ , we say that  $C$  is *maximal*.

If  $0 \notin L$ , we say that  $C$  is *zerofree*, non-*zerofree* otherwise.

We can accept in  $\mathcal{P}$  also rational functions of type  $f/g$ ,  $f, g \in \mathbb{F}_{q^m}$ , such that  $g(\bar{x}) \neq 0$  for any  $\bar{x} \in \mathbb{F}_{q^m}$ . We do so from now on.

*Example 1* Let  $q = 2$ ,  $n = 7$ ,  $q^m = 8$ ,  $L = \mathbb{F}_{2^3} = \langle \beta \rangle \cup \{0\}$  and  $\mathcal{P} = \{g_1(x) = \frac{1}{x^2+x+1}, g_2(x) = \frac{x}{x^2+x+1}\}$ . The seven 7th roots of unity are all the elements of  $\mathbb{F}_8^*$ ,  $R_7 = \mathbb{F}_8^*$ . The  $n$ th-root code  $C = \Omega(2, 7, 8, \mathbb{F}_8, \{g_1, g_2\})$  is non-*zerofree* ( $0 \in L$ ), maximal and its parity-check matrix is the following:

$$H = \begin{pmatrix} g_1(1) & g_1(\beta) & g_1(\beta^2) & g_1(\beta^3) & g_1(\beta^4) & g_1(\beta^5) & g_1(\beta^6) & g_1(0) \\ g_2(1) & g_2(\beta) & g_2(\beta^2) & g_2(\beta^3) & g_2(\beta^4) & g_2(\beta^5) & g_2(\beta^6) & g_2(0) \end{pmatrix}.$$

It is easy to see that  $C$  is an [8,2,5] code.

*Remark 2* In order to define the same  $n$ th-root code, it is possible to use different  $n$ . For example to define a linear code with length  $N = 5$ , we can use the five 5th roots of unity or five 7th roots of unity.

**Proposition 1** Let  $C$  be a linear code over  $\mathbb{F}_q$  of length  $N$  and  $d \geq 2$ . Then  $C$  is an  $n$ th-root code for any  $n \geq N - 1$  such that  $(n, q) = 1$ . In particular:

1. if  $n = N$ , then  $C$  can be maximal *zerofree*,
2. if  $n = N - 1$ , then  $C$  is maximal non-*zerofree*.

**Corollary 1** Let  $C$  be a linear code. Then  $C$  is an  $n$ th-root code if and only if  $d \geq 2$ .

## 1.1 Computing Distance and Weight Distribution for an $n$ th-Root Code

We provide a method to compute the distance and the weight distribution of a code  $C$ , given a representation of  $C$  as an  $n$ th-root code.

**Definition 3** Let  $C = \Omega(q, n, q^m, L, \mathcal{P})$  be an  $n$ th-root code,  $w$  and  $\hat{w}$  be natural numbers such that  $2 \leq w \leq N = |L|$ ,  $1 \leq \hat{w} \leq N - 1$ . We denote by  $J_w(C)$  and  $\hat{J}_{\hat{w}}(C)$  the following two ideals:

$$J_w = J_w(C) = J_w(q, n, q^m, L, \mathcal{P}) \subset \mathbb{F}_{q^m}[z_1, \dots, z_w, y_1, \dots, y_w],$$

$$\hat{J}_{\hat{w}} = \hat{J}_{\hat{w}}(C) = \hat{J}_{\hat{w}}(q, n, q^m, L, \mathcal{P}) \subset \mathbb{F}_{q^m}[z_1, \dots, z_{\hat{w}}, y_1, \dots, y_{\hat{w}}, v],$$

$$J_w = \left\langle \left\{ \sum_{h=1}^w y_h g_s(z_h) \right\}_{1 \leq s \leq r}, \{y_j^{q-1} - 1\}_{1 \leq j \leq w}, \right\rangle$$

$$\{p_{ij}(z_i, z_j)\}_{1 \leq i < j \leq w}, \left\{ \frac{z_j^n - 1}{\prod_{l \in \bar{L}} (z_j - l)} \right\}_{1 \leq j \leq w}, \quad (1)$$

$$\hat{J}_{\hat{w}} = \left\langle \left\{ \sum_{h=1}^{\hat{w}} y_h g_s(z_h) + v g_s(0) \right\}_{1 \leq s \leq r}, \{y_j^{q-1} - 1\}_{1 \leq j \leq \hat{w}}, \right. \\ \left. v^{q-1} - 1, \{p_{ij}(z_i, z_j)\}_{1 \leq i < j \leq \hat{w}}, \left\{ \frac{z_j^n - 1}{\prod_{l \in \bar{L}} (z_j - l)} \right\}_{1 \leq j \leq \hat{w}} \right\rangle \quad (2)$$

where  $p_{ij} = \sum_{h=0}^{n-1} z_i^h z_j^{n-1-h} = \frac{z_i^n - z_j^n}{z_i - z_j}$  are in  $\mathbb{F}_q[z_i, z_j]$ . We denote by  $\eta(J_w)$  and  $\hat{\eta}(\hat{J}_{\hat{w}})$  the integers  $\eta(J_w) = |\mathcal{V}(J_w)|$ ,  $\hat{\eta}(\hat{J}_{\hat{w}}) = |\mathcal{V}(\hat{J}_{\hat{w}})|$ .

*Remark 3* If we are in the binary case ( $q = 2$ ), variables  $y_j$ ,  $j = 1, \dots, w$ , and  $v$  are 1, and so we can omit them and the ideals  $J_w$  and  $\hat{J}_{\hat{w}}$  can be written in  $\mathbb{F}_{2^m}[z_1, \dots, z_w]$  and  $\mathbb{F}_{2^m}[z_1, \dots, z_{\hat{w}}]$ , respectively.

**Proposition 2** Let  $C = \Omega(q, n, q^m, L, \mathcal{P})$  be an  $n$ th-root code. In the zerofree case, there is at least one codeword of weight  $w$  in  $C$  if and only if there exists at least one solution of  $J_w(C)$ . In the non-zerofree case, there is at least one codeword of weight  $w$  in  $C$  if and only if there exists at least one solution of  $J_w(C)$  or of  $\hat{J}_{w-1}(C)$ . Moreover, the number of codewords of weight  $w$  is

$$A_w = \frac{\eta(J_w)}{w!} \quad \text{in the zerofree case and} \\ A_w = \frac{\eta(J_w)}{w!} + \frac{\hat{\eta}(\hat{J}_{w-1})}{(w-1)!} \quad \text{in the non-zerofree case}$$

Since the number of solutions of an ideal  $J$  is directly computed from any Gröbner basis of  $J$  (Mora 2009), we can compute the weight distribution (and the distance) of an  $n$ th-root code, by applying Proposition 2.

*Example 2* Consider the  $n$ th-root code  $C$  as in Example 1. We compute its weight distribution by applying Proposition 2. Setting  $w = 2$  we construct ideals  $J_2(C) \subseteq \mathbb{F}_2[z_1, z_2]$  and  $\hat{J}_1(C) \subseteq \mathbb{F}_2[z_1]$ :

$$J_2(C) = \langle g_1(z_1) + g_1(z_2), g_2(z_1) + g_2(z_2), z_1^7 - 1, z_2^7 - 1, p(z_1, z_2) \rangle \\ \hat{J}_1(C) = \langle g_1(z_1) + g_1(0), g_2(z_1) + g_2(0), z_1^7 - 1 \rangle$$

Their Gröbner bases  $\mathcal{G}_2$  and  $\hat{\mathcal{G}}_1$  are trivial and hence there are no words of weight 2 in this  $n$ th-root code. The same happens for  $w = 3$  and  $w = 4$ , so that  $A_3 = A_4 = 0$ . Setting  $w = 5$  we construct the ideals  $J_5$  and  $\hat{J}_4$ . Basis  $\mathcal{G}_5$  is trivial, but basis  $\hat{\mathcal{G}}_4$  has the following leading terms

$$\{z_1 z_2, z_1^2, z_1 z_3^2, z_2^3, z_1 z_4^3, z_3^4, z_2^2 z_3^2, z_4^5, z_2^2 z_4^3, z_3^3 z_4^3\}.$$

These monomials permit us to compute the number  $\hat{\eta}(\hat{J}_4) = 48$ . We get  $A_5 = \frac{\eta(J_5)}{5!} + \frac{\hat{\eta}(\hat{J}_4)}{4!} = \frac{48}{4!} = 2$  (note that the two words of weight 5 in  $C$  have the last component non-zero). Computing  $\mathcal{G}_6$  we have a non-trivial result,  $\eta(J_6) = 720$ , and for  $\hat{J}_5$  we get an empty variety. The words of weight 6 are then  $A_6 = \frac{\eta(J_6)}{6!} + \frac{\hat{\eta}(\hat{J}_5)}{5!} = \frac{720}{6!} = 1$ . Summarizing, we have:  $A_0 = 1$ ,  $A_w = 0$  for  $w = 1, 2, 3, 4, 7, 8$ ,  $A_5 = 2$  and  $A_6 = 1$ .

*Remark 4* A similar approach permits to compute the weight distribution and the distance for the cosets.

## 2 Conclusions and Further Research

The  $n$ th-root codes allows an extension to linear codes of some computational algebra techniques and some argument, that have been previously applied to cyclic codes (see also Mora and Orsini 2009). This translates in new tools, in particular algorithms to compute the weight distribution (and the distance), but also in new challenges, because it is not clear which  $n$ th root presentation fits better a given code.

**Acknowledgements** The author would like to thank her supervisors F. Dalla Volta and M. Sala.

The author acknowledges support from the Austrian Academy of Science during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- M. Giorgetti, *On some algebraic interpretation of classical codes*, Ph.D. thesis, University of Milan, 2006.
- M. Giorgetti and M. Sala, *A commutative algebra approach to linear codes*, BCRI preprint, [www.bcri.ucc.ie](http://www.bcri.ucc.ie), 58, UCC, Cork, Ireland, 2006.
- M. Giorgetti and M. Sala, *A commutative algebra approach to linear codes*, Journal of Algebra **321** (2009), no. 8, 2259–2286.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, this volume, 2009, pp. 69–91.
- T. Mora and M. Sala, *On the Gröbner bases of some symmetric systems and their application to coding theory*, J. Symbolic Comput. **35** (2003), no. 2, 177–194.
- M. Sala, *Gröbner bases and distance of cyclic codes*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 2, 137–162.

# Decoding Linear Error-Correcting Codes up to Half the Minimum Distance with Gröbner Bases

Stanislav Bulygin and Ruud Pellikaan

**Abstract** In this short note we show how one can decode linear error-correcting codes up to half the minimum distance via solving a system of polynomial equations over a finite field. We also explicitly present the reduced Gröbner basis for the system considered.

## 1 Introduction

In recent years a lot of attention was paid to the question of decoding and finding the minimum distance using Gröbner bases in particular in the case of cyclic codes, which form a particular subclass of linear codes. We mention just a few references in this field (Augot 1996; Augot et al. 2009; Chen et al. 1994a, 1994b; Orsini and Sala 2005) (but see also Mora and Orsini 2009). In this short note we give a method for decoding and finding the minimum distance for arbitrary linear codes.

## 2 Matrix in MDS Form

Let  $\mathbb{F}$  be a field. Let  $\bar{\mathbb{F}}$  be the algebraic closure of  $\mathbb{F}$ . Let  $\mathbf{b}_1, \dots, \mathbf{b}_n$  be a basis of  $\mathbb{F}^n$ . Now  $B$  is the  $n \times n$  matrix with  $\mathbf{b}_1, \dots, \mathbf{b}_n$  as rows.

**Definition 2.1** The (*unknown*) syndrome  $\mathbf{u}(B, \mathbf{e})$  of a word  $\mathbf{e}$  with respect to  $B$  is the column vector  $\mathbf{u}(B, \mathbf{e}) = B\mathbf{e}^T$ . It has entries  $u_i(B, \mathbf{e}) = \mathbf{b}_i \cdot \mathbf{e}$  for  $i = 1, \dots, n$ .

*Remark 2.2* The matrix  $B$  is invertible, since its rank is  $n$ . The syndrome  $\mathbf{u}(B, \mathbf{e})$  determines the error vector  $\mathbf{e}$  uniquely, since

$$B^{-1}\mathbf{u}(B, \mathbf{e}) = B^{-1}B\mathbf{e}^T = \mathbf{e}^T.$$

---

S. Bulygin

Department of Mathematics, University of Kaiserslautern, P.O. Box 3049, 67653  
Kaiserslautern, Germany  
e-mail: [bulygin@mathematik.uni-kl.de](mailto:bulygin@mathematik.uni-kl.de)

R. Pellikaan

Department of Mathematics and Computing Science, Eindhoven University of  
Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands  
e-mail: [g.r.pellikaan@tue.nl](mailto:g.r.pellikaan@tue.nl)

**Definition 2.3** Define the coordinatewise star product of two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$  by  $\mathbf{x} * \mathbf{y} = (x_1 y_1, \dots, x_n y_n)$ . Then  $\mathbf{b}_i * \mathbf{b}_j$  is a linear combination of the basis  $\mathbf{b}_1, \dots, \mathbf{b}_n$ , that is, there are constants  $\mu_{ijl} \in \mathbb{F}$  such that

$$\mathbf{b}_i * \mathbf{b}_j = \sum_{l=1}^n \mu_{ijl} \mathbf{b}_l.$$

The elements  $\mu_{ijl} \in \mathbb{F}$  are called the *structure constants* of the basis  $\mathbf{b}_1, \dots, \mathbf{b}_n$ .

**Definition 2.4** Let  $\mathbf{b}_1, \dots, \mathbf{b}_n$  be a basis of  $\mathbb{F}^n$ . Let  $B_r$  be the  $r \times n$  matrix with  $\mathbf{b}_1, \dots, \mathbf{b}_r$  as rows. Let  $B = B_n$ . We say that  $\mathbf{b}_1, \dots, \mathbf{b}_n$  is an *ordered MDS basis* and  $B$  an *MDS matrix* if all the  $t \times t$  submatrices of  $B_t$  have rank  $t$  for all  $t = 1, \dots, n$ . Let  $C_t$  be the code with  $B_t$  as parity check matrix.

*Remark 2.5* Let  $B$  be an MDS matrix. Then  $C_t$  is an MDS code for all  $t$ .

**Definition 2.6** Suppose  $n \leq q$ . Let  $\mathbf{x} = (x_1, \dots, x_n)$  be an  $n$ -tuple of mutually distinct elements in  $\mathbb{F}$ . Define

$$\mathbf{b}_i = (x_1^{i-1}, \dots, x_n^{i-1}).$$

Then  $\mathbf{b}_1, \dots, \mathbf{b}_n$  is called an *ordered Vandermonde basis* and the corresponding matrix is denoted by  $B(\mathbf{x})$  and called a *Vandermonde matrix*.

### 3 Decoding up to Half the Minimum Distance

It can be shown that if we have a linear  $[n, k, d]$  code  $C$  over the field  $\mathbb{F}_q$ , then a code  $C' = C\mathbb{F}_{q^m}$  has the same parameters. So without loss of generality we may assume, after a finite extension of the finite field  $\mathbb{F}_q$ , that  $n \leq q$ . Let  $\mathbf{b}_1, \dots, \mathbf{b}_n$  be a basis of  $\mathbb{F}_q^n$ . From now on we assume that the corresponding matrix  $B$  is an MDS matrix.

Let  $C$  be an  $\mathbb{F}_q$ -linear code with parameters  $[n, k, d]$ . Choose a parity check matrix  $H$  of  $C$ . The redundancy is  $r = n - k$ . Let  $\mathbf{h}_1, \dots, \mathbf{h}_r$  be the rows of  $H$ . The row  $\mathbf{h}_i$  is a linear combination of the basis  $\mathbf{b}_1, \dots, \mathbf{b}_n$ , that is, there are constants  $a_{ij} \in \mathbb{F}_q$  such that

$$\mathbf{h}_i = \sum_{j=1}^n a_{ij} \mathbf{b}_j.$$

In other words  $H = AB$  where  $A$  is the  $r \times n$  matrix with entries  $a_{ij}$ .

*Remark 3.1* Let  $\mathbf{r} = \mathbf{c} + \mathbf{e}$  be a received word with  $\mathbf{c} \in C$  a codeword and  $\mathbf{e}$  an error vector. The syndromes of  $\mathbf{r}$  and  $\mathbf{e}$  with respect to  $H$  are equal and known:

$s_i(\mathbf{r}) := \mathbf{h}_i \cdot \mathbf{r} = \mathbf{h}_i \cdot \mathbf{e} = s_i(\mathbf{e})$  and they can be expressed in the unknown syndromes of  $\mathbf{e}$  with respect to  $B$ :

$$s_i(\mathbf{r}) = \sum_{j=1}^n a_{ij} u_j(\mathbf{e}),$$

since  $\mathbf{h}_i = \sum_{j=1}^n a_{ij} \mathbf{b}_j$  and  $\mathbf{b}_j \cdot \mathbf{e} = u_j(\mathbf{e})$ .

**Definition 3.2** Let  $B$  be an MDS matrix with structure constants  $\mu_{ijl}$ . Define the linear functions  $U_{ij}$  in the variables  $U_1, \dots, U_l$  by

$$U_{ij} = \sum_{l=1}^n \mu_{ijl} U_l.$$

Let  $\mathcal{U}$  be the  $n \times n$  matrix with entries  $U_{ij}$ . Let  $\mathcal{U}_{u,v}$  be the  $u \times v$  matrix with entries  $U_{ij}$  with  $1 \leq i \leq u$  and  $1 \leq j \leq v$ .

**Definition 3.3** The ideal  $J(\mathbf{r})$  in the ring  $\mathbb{F}_q[U_1, \dots, U_n]$  is generated by the elements

$$\sum_{l=1}^n a_{jl} U_l - s_j(\mathbf{r}) \quad \text{for } j = 1, \dots, r$$

The ideal  $I(t, \mathcal{U}, V)$  in the ring  $\mathbb{F}[U_1, \dots, U_n, V_1, \dots, V_t]$  is generated by the elements

$$\sum_{j=1}^t U_{ij} V_j - U_{it+1} \quad \text{for } i = 1, \dots, n$$

Let  $J(t, \mathbf{r})$  be the ideal in  $\mathbb{F}_q[U_1, \dots, U_n, V_1, \dots, V_t]$  generated by  $J(\mathbf{r})$  and  $I(t, \mathcal{U}, V)$ .

**Remark 3.4** The ideal  $J(t, \mathbf{r})$  is generated by  $n - k$  linear functions and  $n$  quadratic polynomials.

**Theorem 3.5** Let  $B$  be an MDS matrix with structure constants  $\mu_{ijl}$  and linear functions  $U_{ij}$ . Let  $H$  be a parity check matrix of the code  $C$  such that  $H = AB$ . Let  $\mathbf{r} = \mathbf{c} + \mathbf{e}$  be a received word with  $\mathbf{c}$  in  $C$  the codeword sent and  $\mathbf{e}$  the error vector. Suppose that the weight of  $\mathbf{e}$  is not zero and at most  $(d(C) - 1)/2$ . Let  $t$  be the smallest positive integer such that  $J(t, \mathbf{r})$  has a solution  $(\mathbf{u}, \mathbf{v})$  over  $\bar{\mathbb{F}}_q$ . Then  $\text{wt}(\mathbf{e}) = t$  and the solution is unique satisfying  $\mathbf{u} = \mathbf{u}(\mathbf{e})$ .

**Corollary 3.6** Let  $\mathbf{r} = \mathbf{c} + \mathbf{e}$  be a received word with  $\mathbf{c}$  in  $C$  the codeword sent and  $\mathbf{e}$  the error vector. Suppose that the weight of  $\mathbf{e}$  is not zero and at most  $(d(C) - 1)/2$ . Let  $t$  be the smallest positive integer such that  $J(t, \mathbf{r})$  has a solution. Then

the solution is unique and the reduced Gröbner basis  $G$  for the ideal  $J(t, \mathbf{r})$  with respect to any monomial ordering is

$$U_i - u_i(\mathbf{e}), \quad i = 1, \dots, n,$$

$$V_j - v_j, \quad j = 1, \dots, t,$$

where  $(\mathbf{u}(\mathbf{e}), \mathbf{v})$  is the unique solution.

## 4 Conclusion and Future Work

In this short note we briefly described how the problem of decoding up to half the minimum distance can be translated to solving a system of polynomial equations over a finite field. Moreover, due to the uniqueness of the solution of such a system (and because the multiplicity of such a solution is one) we were able to explicitly write down the reduced Gröbner basis for the system. As a future work we would like to elaborate more on possible methods for solving such system and also estimate the complexity of the corresponding Gröbner basis computations. The full version of this short note is Bulygin and Pellikaan (2009). The ideas of the note stem from Høholdt et al. (1998). One can try some computations with the above system with the use of decodegb.lib,<sup>1</sup> which is the library for SINGULAR Computer Algebra System for polynomial computations (Greuel et al. 2007).

**Acknowledgements** The authors acknowledge support from the Austrian Academy of Sciences during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- D. Augot, Description of the minimum weight codewords of cyclic codes by algebraic system, *Finite Fields Appl.* (1996), no. 2, 138–152.
- D. Augot, M. Bardet, and J.-C. Faugère, *On the decoding of cyclic codes with Newton identities*, Special Issue “Gröbner Bases Techniques in Cryptography and Coding Theory” of *J. Symb. Comput.*, 2009.
- S. Bulygin and R. Pellikaan, *Bounded decoding of linear error-correcting codes with Gröbner bases*, Special Issue “Gröbner Bases Techniques in Cryptography and Coding Theory” of *J. Symb. Comput.*, to appear, 2009.
- X. Chen, I. S. Reed, T. Helleseth, and K. Truong, *General principles for the algebraic decoding of cyclic codes*, *IEEE Trans. on Inf. Th.* **40** (1994a), 1661–1663.
- X. Chen, I. S. Reed, T. Helleseth, and T. K. Truong, *Algebraic decoding of cyclic codes: a polynomial ideal point of view*, *Finite fields, Contemp. Math.*, vol. **168**, Am. Math. Soc., Providence, 1994b, pp. 15–22.
- G.-M. Greuel, G. Pfister, and H. Schönemann, *Singular 3.0. A computer algebra system for polynomial computations*, <http://www.singular.uni-kl.de>, 2007, Centre for Computer Algebra, University of Kaiserslautern.

---

<sup>1</sup><http://www.mathematik.uni-kl.de/~bulygin/files/decodegb.lib>.

- T. Høholdt, J. van Lint, and R. Pellikaan, *Algebraic geometry of codes*, Handbook of Coding Theory (V. S. Pless and W.C. Huffman, eds.), Elsevier, Amsterdam, 1998, pp. 871–961.
- T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, this volume, 2009, pp. 69–91.
- E. Orsini and M. Sala, *Correcting errors and erasures via the syndrome variety*, J. Pure Appl. Algebra **200** (2005), 191–226.

# Gröbner Bases for the Distance Distribution of Systematic Codes

Eleonora Guerrini, Emmanuela Orsini and  
Ilaria Simonetti

**Abstract** Coding theorists have been studying only linear codes, with a few exceptions (Preparata in *Inform. Control* 13(13):378–400, 1968; Baker et al. in *IEEE Trans. on Inf. Th.* 29(3):342–345, 1983). This is not surprising, since linear codes have a nice structure, easy to study and leading to efficient implementations. However, it is well-known that some non-linear codes have a higher distance (or a better distance distribution) than any linear code with the same parameters (Preparata in *Inform. Control* 13(13):378–400, 1968; Pless et al. (eds.) in *Handbook of Coding Theory*, vols. I, II, North-Holland, Amsterdam, 1998). This translates into a superior decoding performance (Litsyn in *Handbook of Coding Theory*, vols. I, II, North-Holland, Amsterdam, pp. 463–498, 1998).

Systematic non-linear codes are the most studied non-linear codes. We describe a Gröbner bases technique to compute the distance distribution for these codes.

## 1 Preliminaries

Throughout this paper  $m, k, n, q, p, r$  are integers such that  $m \geq 1$ ,  $1 \leq k \leq n$ ,  $q = p^r$ ,  $r \geq 1$  and  $p$  is a prime.

Let  $\mathbb{F}_q$  be the finite field with  $q$  elements and  $(\mathbb{F}_q)^m$  be the natural  $m$ -dimensional vector space over  $\mathbb{F}_q$ . Let  $\overline{\mathbb{F}}_q$  be the algebraic closure of  $\mathbb{F}_q$ . Given an ideal  $I$  in  $\mathbb{F}_q[Y] = \mathbb{F}_q[y_1, \dots, y_m]$ , we denote by  $\mathcal{V}(I) \subset (\overline{\mathbb{F}}_q)^m$  the set of all zeros of  $I$ . Let  $S \subseteq (\overline{\mathbb{F}}_q)^m$ , the set of all polynomials  $f \in \mathbb{F}_q[Y]$ , such that  $f(\mathbf{s}) = 0$  for any points  $\mathbf{s} \in S$ , is an ideal  $\mathcal{I}(S)$  in the polynomial ring  $\mathbb{F}_q[Y]$ , which is called the *vanishing ideal* of  $S$ . We denote by  $E_q[Y]$  the following set of polynomials in  $\mathbb{F}_q[Y]$ ,  $E_q[Y] = \{y_1^q - y_1, \dots, y_m^q - y_m\}$ .

---

E. Guerrini  
Department of Mathematics, University of Trento, Trento, Italy  
e-mail: [guerrini@science.unitn.it](mailto:guerrini@science.unitn.it)

E. Orsini  
Department of Mathematics, University of Pisa, Pisa, Italy  
e-mail: [orsini@posso.dm.unipi.it](mailto:orsini@posso.dm.unipi.it)

I. Simonetti  
Department of Mathematics, University of Milan, Milan, Italy  
e-mail: [simonetti@mat.unimi.it](mailto:simonetti@mat.unimi.it)

**Definition 1** Let  $1 \leq t \leq m$  and  $\mathbf{m} \in \mathbb{F}_q[y_1, y_2, \dots, y_m]$ . We say that  $\mathbf{m}$  is a *square-free t-monomial* if  $\mathbf{m} = y_{h_1} \cdots y_{h_t}$ , where  $h_1, \dots, h_t \in \{1, \dots, m\}$  and  $h_l \neq h_j$ ,  $\forall l \neq j$ , i.e.  $\mathbf{m}$  is a monomial in  $\mathbb{F}_q[Y]$  such that  $\deg_{y_{h_i}}(\mathbf{m}) = 1$  for any  $1 \leq i \leq t$ . We denote by  $\mathcal{M}_{m,t,q}$  the set of square-free  $t$ -monomials in  $\mathbb{F}_q[Y]$ .

We assume the reader familiar with Gröbner basis theory, as in the book chapter (Mora 2009).

We keep the reference to  $q$  implicit, so we use  $\mathcal{M}_{m,l}$  instead of  $\mathcal{M}_{m,l,q}$ .

**Definition 2** Let  $\phi : (\mathbb{F}_q)^k \rightarrow (\mathbb{F}_q)^n$  be an injective function and let  $C = \text{Im}(\phi)$ . We say that  $C$  is an  $(n, k, q)$  *code*. Any  $c \in C$  is called a *word* of  $C$ .

Let  $\pi$  be the projection  $\pi : (\mathbb{F}_q)^n \rightarrow (\mathbb{F}_q)^k$  such that  $\pi(a_1, \dots, a_n) = (a_1, \dots, a_k)$ . We say that  $C$  is *systematic* if  $(\pi \circ \phi)(v) = v$ , for any  $v \in (\mathbb{F}_q)^k$ . We denote by  $\mathcal{C}(n, k, q)$  the class of all systematic  $(n, k, q)$  codes.

For any  $C \in \mathcal{C}(n, k, q)$  and any  $0 \leq i \leq n$ , we denote by  $B_i(C)$  the number of words in  $C$  with weight  $i$  and by  $A_i(C)$  the number of word pairs in  $C$  with distance  $i$ .

## 2 Theoretical Results

All definitions and results of this section can be found in Guerrini et al. (2006) and Guerrini (2005).

If  $C \in \mathcal{C}(n, k, q)$ , then we can view  $C$  as a set of points in  $(\mathbb{F}_q)^n \subset (\overline{\mathbb{F}}_q)^n$  and hence as a 0-dimensional variety, so that  $\mathcal{I}(C)$  is its vanishing ideal in  $\mathbb{F}_q[X, Z] = \mathbb{F}_q[x_1, \dots, x_k, z_1, \dots, z_{n-k}]$ . We describe a Gröbner basis of  $\mathcal{I}(C)$ .

**Theorem 1** Let  $G$  be the reduced Gröbner basis for  $\mathcal{I}(C)$ , w.r.t. the lex order with  $x_1 < \dots < x_k < z_1 < \dots < z_{n-k}$ . Then  $G$  has the following structure:

$$G = \{E_q[X], z_1 - \mathbf{f}_1, \dots, z_{n-k} - \mathbf{f}_{n-k}\}$$

for some  $\mathbf{f}_j \in \mathbb{F}_q[X]$ ,  $1 \leq j \leq n-k$ .

From now on, we consider  $C$  and  $G(C)$  to be understood.

**Definition 3** Let  $t \in \mathbb{N}$ ,  $1 \leq t \leq n$ . We define the ideal  $\mathcal{W}_C^t \subseteq \mathbb{F}_q[x_1, \dots, x_k]$  as generated by:

$$E_q[X] \cup \{\mathbf{m}(x_1, \dots, x_k, \mathbf{f}_1(X), \dots, \mathbf{f}_{n-k}(X)) \mid \mathbf{m} \in \mathcal{M}_{n,t}\}.$$

A point in  $\mathcal{V}(\mathcal{W}_C^t)$  matches a codeword  $c$  in  $C$  with  $w(c) \leq t-1$ , so that we can compute the weight distribution  $\{B_0(C), \dots, B_n(C)\}$  of  $C$  as follows.

**Proposition 1** Let  $t \in \mathbb{N}$  such that  $2 \leq t \leq n$ . Then

$$\begin{cases} B_{t-1}(C) = |\mathcal{V}(\mathcal{W}_C^t)| \setminus |\mathcal{V}(\mathcal{W}_C^{t-1})| \\ B_0 = |\mathcal{V}(\mathcal{W}_C^1)|. \end{cases}$$

**Definition 4** Let  $\mathbb{F}_q[X, \tilde{X}] = \mathbb{F}_q[x_1, x_2, \dots, x_k, \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k]$ . We denote by  $L_{n,k,t} \subset \mathbb{F}_q[X, \tilde{X}]$  the following subset

$$L_{n,k,t} = \{x_1 - \tilde{x}_1, \dots, x_k - \tilde{x}_k\} \cup \{\mathbf{f}_1(X) - \mathbf{f}_1(\tilde{X}), \dots, \mathbf{f}_{n-k}(X) - \mathbf{f}_{n-k}(\tilde{X})\}.$$

We denote by  $\mathcal{I}_C^t$  the ideal in  $\mathbb{F}_q[X, \tilde{X}]$  generated by:

$$\{x_i^q - x_i, \tilde{x}_i^q - \tilde{x}_i \mid 1 \leq i \leq k\} \cup \{\mathbf{m}(L_{n,k,t}) \mid \mathbf{m} \in \mathcal{M}_{n,t}\}.$$

**Definition 5** In  $(\mathbb{F}_q)^k \times (\mathbb{F}_q)^k$  we denote by  $\mathcal{T}_k$  the *trivial variety*, i.e. the set of points  $a = (a_1, \dots, a_k, \tilde{a}_1, \dots, \tilde{a}_k)$  such that  $a_i = \tilde{a}_i$ ,  $1 \leq i \leq k$ .

**Theorem 2** Let  $t \in \mathbb{N}$  such that  $2 \leq t \leq n$ . Then

$$\mathcal{V}(\mathcal{I}_C^t) \neq \mathcal{T}_k \iff \exists c_1, c_2 \in C \text{ such that } d(c_1, c_2) \leq t - 1.$$

From Theorem 2, an algorithm is directly designed to compute the distance of any  $C \in \mathcal{C}(n, k, q)$ .

```

 $j = 1$ 
While  $\mathcal{V}(\mathcal{I}_C^j) = \mathcal{T}_k$  do
     $j := j + 1;$ 
Output  $j$ 

```

Theorem 2 can be improved to give the distance distribution  $\{A_1(C), \dots, A_n(C)\}$  of  $C$ , as follows.

**Theorem 3** Let  $2 \leq t \leq n$ . Then

$$\mathcal{V}(\mathcal{I}_C^t) = \{(c_1, c_2) \mid c_1, c_2 \in C, d(c_1, c_2) \leq t - 1\},$$

$$A_1 + A_2 + \dots + A_{t-1} = \frac{|\mathcal{V}(\mathcal{I}_C^t)| \setminus |\mathcal{T}_k|}{2},$$

and

$$A_{t-1} = \frac{|\mathcal{V}(\mathcal{I}_C^t)| \setminus |\mathcal{V}(\mathcal{I}_C^{t-1})|}{2}.$$

*Example 1* Let  $C = \{[0, 0, 0, 0], [0, 1, 0, 0], [0, 2, 0, 0], [1, 0, 0, 0], [1, 1, 0, 2], [1, 2, 0, 2], [2, 0, 0, 2], [2, 1, 0, 0], [2, 2, 0, 0]\}$ . Clearly,  $C$  is a code in  $\mathcal{C}(4, 2, 3)$ .

We want to compute all pairs of words  $(c_1, c_2)$  with  $d(c_1, c_2) \leq 2$ .

To do this, we start from the input basis of ideal  $\mathcal{I}_C^3 \subset \mathbb{F}_3[x_1, x_2, \tilde{x}_1, \tilde{x}_2]$ :

$$\begin{aligned} \mathcal{I}_C^3 = & \langle x_2^2 \tilde{x}_1^2 \tilde{x}_2 - x_1^2 \tilde{x}_1^2 \tilde{x}_2 - x_1 \tilde{x}_1^2 \tilde{x}_2 - \tilde{x}_1^2 \tilde{x}_2 - x_1^2 x_2^2 \tilde{x}_1 \tilde{x}_2 + x_2^2 \tilde{x}_1 \tilde{x}_2 + x_1^2 \tilde{x}_1 \tilde{x}_2 - \tilde{x}_1 \tilde{x}_2 \\ & - x_1^2 x_2^2 \tilde{x}_2 - x_1^2 \tilde{x}_2 + x_1 \tilde{x}_2 + x_1^2 x_2 \tilde{x}_1^2 + x_1 x_2 \tilde{x}_1^2 - x_1^2 x_2 - x_1 x_2, x_1^2 x_2 \tilde{x}_1^2 \tilde{x}_2 \\ & + x_1 x_2 \tilde{x}_1^2 \tilde{x}_2 - x_1^2 x_2 \tilde{x}_2 - x_1 x_2 \tilde{x}_2 - x_1^2 x_2^2 \tilde{x}_1^2 - x_1 x_2^2 \tilde{x}_1^2 + x_1^2 x_2^2 + x_1 x_2^2, \tilde{x}_2^3 \\ & - \tilde{x}_2, x_2^3 - x_2, \tilde{x}_1^3 - \tilde{x}_1, x_1^3 - x_1, x_2 \tilde{x}_1^2 \tilde{x}_2^2 + x_2 \tilde{x}_1 \tilde{x}_2^2 - x_1^2 x_2 \tilde{x}_2^2 - x_1 x_2 \tilde{x}_2^2 \\ & - x_1^2 \tilde{x}_1^2 \tilde{x}_2 - x_1 \tilde{x}_1^2 \tilde{x}_2 - \tilde{x}_1^2 \tilde{x}_2 - x_1^2 x_2^2 \tilde{x}_1 \tilde{x}_2 + x_1^2 \tilde{x}_1 \tilde{x}_2 - \tilde{x}_1 \tilde{x}_2 + x_1 x_2^2 \tilde{x}_2 \\ & - x_1^2 \tilde{x}_2 + x_1 \tilde{x}_2 + x_1^2 x_2 \tilde{x}_1^2 + x_1 x_2 \tilde{x}_1^2 - x_1^2 x_2 - x_1 x_2, x_1 \tilde{x}_1^2 \tilde{x}_2^2 - \tilde{x}_1^2 \tilde{x}_2^2 + x_2^2 \tilde{x}_1 \tilde{x}_2^2 \\ & + x_1^2 \tilde{x}_1 \tilde{x}_2^2 - \tilde{x}_1 \tilde{x}_2^2 - x_1 x_2^2 \tilde{x}_2^2 + x_1^2 \tilde{x}_2^2 - x_1 \tilde{x}_2^2 - x_2 \tilde{x}_1^2 \tilde{x}_2 - x_2 \tilde{x}_1 \tilde{x}_2 + x_1^2 x_2 \tilde{x}_2 \\ & + x_1 x_2 \tilde{x}_2 - x_1 x_2^2 \tilde{x}_1^2 - x_2^2 \tilde{x}_1^2 - x_1^2 x_2^2 \tilde{x}_1 + x_2^2 \tilde{x}_1 + x_1^2 x_2^2 + x_1 x_2^2, x_1 x_2 \tilde{x}_1 \tilde{x}_2^2 \\ & + x_2 \tilde{x}_1 \tilde{x}_2^2 - x_1^2 x_2 \tilde{x}_2^2 - x_1 x_2 \tilde{x}_2^2 - x_1^2 \tilde{x}_1^2 \tilde{x}_2 + \tilde{x}_1^2 \tilde{x}_2 - x_1^2 x_2^2 \tilde{x}_1 \tilde{x}_2 - x_1 x_2^2 \tilde{x}_1 \tilde{x}_2 \\ & - x_1^2 \tilde{x}_1 \tilde{x}_2 + \tilde{x}_1 \tilde{x}_2 + x_1^2 x_2^2 \tilde{x}_2 + x_1 x_2^2 \tilde{x}_2 + x_1^2 x_2 \tilde{x}_1^2 - x_2 \tilde{x}_1^2 - x_1^2 x_2 \tilde{x}_1 + x_2 \tilde{x}_1 \rangle. \end{aligned}$$

From the Gröbner basis we find that  $|\mathcal{V}(\mathcal{I}_C^3)| = 65$ . Since  $|\mathcal{T}_2| = 9$ , we have  $A_1 + A_2 = (65 - 9)/2 = 28$ .

### 3 Numerical Computations

We provide some numerical computations with time comparisons.

We have collected our results in three consecutive tables. In all tables, we consider codes with  $n = 2k$  and  $d \sim (n - k)$ , which is the hardest case due to the Singleton's bound. In the first table we consider random codes with  $\deg(f_i) = k$ ; in the second table we consider random quadratic codes ( $\deg(f_i) = 2$ ); in the third table we consider random linear codes ( $\deg(f_i) = 1$ ).

These tests have been done with MAGMA on a dual AMD Opteron 2 GHz, equipped with 8 Gb of RAM memory, at the computational centre MEDICIS.

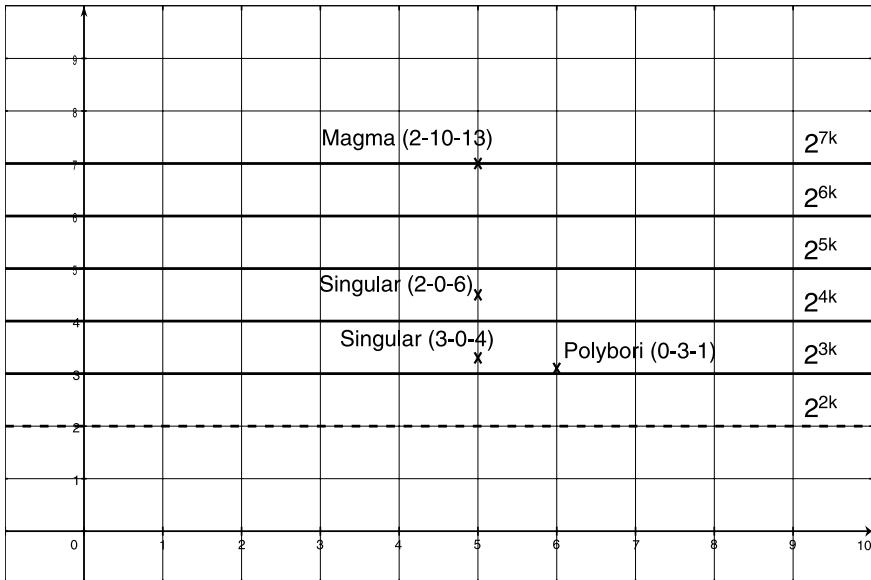
$n$	$k$	Time	$n$	$k$	Time	$n$	$k$	Time
4	2	<.001 s	4	2	<.001 s	4	2	<.001 s
6	3	<.001 s	6	3	<.001 s	6	3	<.001 s
8	4	<.001 s	8	4	<.001 s	8	4	<.001 s
10	5	~2.20 s	10	5	~1.49 s	10	5	<.001 s
12	6	~264.64 s	12	6	~22.73 s	12	6	~3.04 s

We tested our problem using different computer algebra systems (Magma 2.10.13, Polybori 0.3.1,<sup>1</sup> Singular 2.0.6, Singular 3.0.4). We say that for any system the time

---

<sup>1</sup>Polybori is a new software package, still in experimental phase, which is specialized to computing Gröbner bases over  $\mathbb{F}_2$ .

needed had a behavior of kind  $2^{\alpha k}$ , with  $\alpha$  depending on the system. In particular, we report the graph where  $x = k$  and  $y = \log(\frac{\text{time in } k}{\text{time in } k-1})$ , so that the  $y$  values represent the expected exponent.



That suggests us the following values for the computational costs:

- $2^{7k}$  for Magma 2.10.13,
- $2^{4.5k}$  for Singular 2.0.6,
- $2^{3.5k}$  for Singular 3.0.4,
- $2^{3k}$  for Polybori 0.3.1.

We note that a brute-force check of the distance has an asymptotic behaviour like  $2^{2k}$ . Admittedly, this looks much better than our estimate  $2^{3k}$ . However if we examine the drastic improvement obtained by the evolution of computer algebra systems (from  $2^{7k}$  to  $2^{3k}$  in few years), we can reasonably assume that our estimates are still pessimistic and that further improvements in software development will allow our method to run like  $2^{\alpha k}$ , with  $2 < \alpha < 3$ .

**Acknowledgements** The authors acknowledge support from the Austrian Academy of Sciences, during the Special Semester on Gröbner Bases, 2006, (Linz, Austria).

The authors would like to thank their supervisor M. Sala.

This work has been partially supported by STMicroelectronics contract “Complexity issues in algebraic Coding Theory and Cryptography”.

## References

- R. D. Baker, J. H. van Lint and R. M. Wilson, *On the Preparata and Goethals codes*, IEEE Trans. on Inf. Th. **29** (1983), no. 3, 342–345.

- E. Guerrini, *On distance and optimality in non-linear codes*, Master's thesis (laurea), Univ. of Pisa, Dept. of Math., 2005.
- E. Guerrini, M. Orsini, and M. Sala, *Computing the distance distribution of systematic non-linear codes*, BCRI preprint, [www.bcri.ucc.ie](http://www.bcri.ucc.ie) 50, UCC, Cork, Ireland, 2006.
- S. Litsyn, *An updated table of the best binary codes known*, Handbook of coding theory, vols. I, II, North-Holland, Amsterdam, 1998, pp. 463–498.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- V. S. Pless, W. C. Huffman and R. A. Brualdi (eds.), *Handbook of coding theory*, vols. I, II, North-Holland, Amsterdam, 1998.
- F. Preparata, *A class of optimum nonlinear double-error correcting codes*, Inform. Control **13** (1968), no. 13, 378–400.

# A Prize Problem in Coding Theory

Jon-Lark Kim

**Abstract** In this short note, we describe one of the long-standing open problems in algebraic coding theory, i.e., whether there exists a binary self-dual [72, 36, 16] code.

## 1 Introduction

Binary self-dual codes or self-dual codes over finite fields in general have been of great interest partly because many good linear block codes are either self-orthogonal or self-dual. It turns out that they satisfy a nonconstructive lower bound, analogous to the Gilbert-Varshamov bound in linear codes. Furthermore, they have nice algebraic properties; in particular, the weight enumerator of a self-dual code over a finite field is invariant under a certain finite matrix group, which restricts the minimum distance of a self-dual code over  $\mathbb{F}_2$ ,  $\mathbb{F}_3$ , or  $\mathbb{F}_4$ . We refer to Rains and Sloane (1998), Nebe et al. (2006) for a full discussion of self-dual codes.

A binary self-dual code  $C$  under the usual inner product is called a *Type II (or doubly-even) code* if all codewords have weight  $\equiv 0 \pmod{4}$ , and a *Type I (or singly-even) code* if there is a codeword whose weight  $\equiv 2 \pmod{4}$ . Given a binary Type I code  $C$ , one can obtain the doubly-even subcode  $C_0$  of  $C$  (consisting of all codewords whose weight  $\equiv 0 \pmod{4}$ ). The *shadow*  $S$  of  $C$  is defined by  $S := C_0^\perp \setminus C$  (Conway and Sloane 1990). The weight enumerator  $S(x, y)$  of the shadow of  $C$  is determined by the weight enumerator  $C(x, y)$  of  $C$  as  $S(x, y) = \frac{1}{|C|} C(x + y, i(x - y))$ , where  $i = \sqrt{-1}$ . This additional relation gives a further restriction on a possible weight enumerator of a binary self-dual code, often proving the nonexistence of a putative binary self-dual code (Conway and Sloane 1990).

Using  $C(x, y)$  and  $S(x, y)$  in a sophisticated way, Rains (1998) derived a tight upper bound on the minimum distance of a binary self-dual code. More precisely, if  $C$  is a binary self-dual code of length  $n$  with minimum distance  $d$  then  $d \leq 4\lfloor n/24 \rfloor + 4$  except when  $n \equiv 22 \pmod{24}$ , in which case  $d \leq 4\lfloor n/24 \rfloor + 6$  (see Rains 1998). Further if  $C$  is a Type I code of length  $n \equiv 0 \pmod{24}$ , then  $d \leq 4\lfloor n/24 \rfloor + 2$ . A Type I self-dual code whose minimum distance  $d$  attains this bound is called *extremal*. A Type II code of length  $n$  with minimum distance  $d = 4\lfloor n/24 \rfloor + 4$  is called *extremal*.

---

J.-L. Kim

Department of Mathematics, University of Louisville, Louisville, KY 40292, USA  
e-mail: [jl.kim@louisville.edu](mailto:jl.kim@louisville.edu)

It has been one of important problems in coding theory to find (binary) extremal self-dual codes (see Huffman 2005 for recent results on extremal self-dual codes over  $\mathbb{F}_2$ ,  $\mathbb{F}_3$ ,  $\mathbb{F}_4$ ,  $\mathbb{Z}_4$ ,  $\mathbb{F}_2 + u\mathbb{F}_2$ , and  $\mathbb{F}_2 + v\mathbb{F}_2$ ), due to their connection with other mathematical areas including designs, lattices, and modular forms (Pless et al. 1998; Nebe et al. 2006).

In particular, one of the most famous open problems is the following.

**Problem: Does there exist a Type II [24k, 12k, 4k + 4] code  $C(k)$  for  $k \geq 3$ ?**

We note the following results.

1. If  $k = 1$ , then  $C(1)$  is the Type II [24, 12, 8] code (the binary extended Golay code). In fact, any binary linear code with parameters [24, 12, 8] is equivalent to  $C(1)$  (Pless 1968).
2. If  $k = 2$ , then  $C(2)$  is the extended quadratic residue code  $XQ_{47}$  of length 48. This is unique up to equivalence among self-dual codes with parameters [48, 24, 12] (Houghten et al. 2003). It is not known whether there is a linear binary [48, 24, 12] code other than  $XQ_{47}$ .
3. **The existence of a Type II [72,36,16] code  $C(3)$  is one of the long-standing open problems in coding theory.** This was officially suggested by Sloane (1973). If it exists, then the codewords of weight 16 form a 5-(72, 16, 78) design whose existence is unknown.
4. If  $k \geq 154$ , then  $C(k)$  does not exist since  $A_{4k+8}$  (the number of codewords of weight  $4k + 8$ ) is negative (Zhang 1999).

## 2 Related Facts about a Putative Type II [72, 36, 16] Code

The weight enumerator of a putative Type II [72, 36, 16] code  $C(3)$  is:

$$\begin{aligned} W = & 1 + 249,849y^{16} + 18,106,704y^{20} + 462,962,955y^{24} + 4,397,342,400y^{28} \\ & + 16,602,715,899y^{32} + 25,756,721,120y^{36} + \dots \end{aligned}$$

One possible attack to prove or disprove the existence of  $C(3)$  is to investigate the order of the automorphism group of  $C(3)$ . The only possible *prime orders* of an automorphism of  $C(3)$  are 2, 3, 5, and 7. It is remarked (Huffman 2005) that Yorgov recently proved that the automorphism group has order a divisor of 72 or order 504, 252, 56, 14, 7, 360, 180, 60, 30, 10, or 5.

Another attack is to construct codes related to  $C(3)$ . The existence of  $C(3)$  is equivalent to that of a Type I [70, 35, 14] code (Rains 1998). The weight enumerator of a Type I [70, 35, 14] code is corrected in Huffman (2005) as follows:

$$W = 1 + 11,730y^{14} + 150,535y^{16} + 1,345,960y^{18} + \dots$$

Gulliver et al. (2003) showed that the existence of  $C(k)$  implies the existence of a Type I  $[24k, 12k, 4k + 2]$  code for  $k \geq 1$ . Hence if there is  $C(3)$ , then there is a Type I  $[72, 36, 14]$  code. Equivalently, if there is no Type I  $[72, 36, 14]$  code, there is no  $C(3)$ . No self-dual codes with parameters  $[72, 36, 14]$  are known to exist. There are exactly three possible weight enumerators for a Type I  $[72, 36, 14]$  code as follows.

$$W_1 = 1 + 7616y^{14} + 134,521y^{16} + 1,151,040y^{18} + \dots,$$

$$W_2 = 1 + 8576y^{14} + 124,665y^{16} + 1,206,912y^{18} + \dots,$$

$$W_3 = 1 + 8640y^{14} + 124,281y^{16} + 1,207,360y^{18} + \dots.$$

### 3 Future Work

There is a hope that  $C(3)$  might exist. For example, although it is not known yet whether there exists a binary linear  $[72, 36, 16]$  code, there is a  $[72, 36, 15]$  code by puncturing a  $[73, 36, 16]$  cyclic code and any  $[72, 36, d]$  code satisfies  $d \leq 17$  from Brouwer's Table.

A recent attempt to construct  $C(3)$  was made by Dougherty et al. (2007) by considering double circulant codes based on strongly regular graphs and doubly regular tournaments. In particular, SRG (Strongly Regular Graphs) with parameters  $(36, 15, 6, 6)$  produce a lot of Type II  $[72, 36, 12]$  codes. Similarly DRT (Doubly Regular Tournaments) of order 36 produce Type II  $[72, 36, 8 \text{ or } 12]$  codes. It is hoped that  $d = 16$  is possible if there is enough data for DRT of the above parameters.

We have also shown Kim and Solé (2008) that skew Hadamard matrices of order  $4m$  where a prime  $p$  divides  $m$  produce self-dual codes over  $\mathbb{F}_p$ . In particular, if  $m = 18$ , then we have plenty of Type II  $[72, 36, 12]$  codes with various weight enumerators from the 990 skew Hadamard matrices of order 72 in Kotsireas (2006). This motivates an active search for more skew Hadamard matrices of order 72.

From the viewpoint of Gröbner bases, it is shown in Guerrini and Sala (2007) how to construct the input basis of a zero-dimensional polynomial ideal, whose solutions correspond to binary systematic non-linear codes with fixed parameters. It is obvious how to specialize it to classify binary linear codes. By computing the Gröbner basis  $G$  of Guerrini-Sala's ideal  $B$  for parameters  $[72, 36, 16]$ , we would immediately have a complete classification for such codes, if they exist. In particular, if  $G$  is trivial ( $G = \{1\}$ ), then there are no such codes. Otherwise, its solutions can be tested whether they are self-dual. However, it is likely that the computation is infeasible, since  $I$  has  $36^2 = 1296$  variables.

## 4 Monetary Prizes

As far as we know, the existence of  $C(3)$  is the only coding problem with monetary prizes. The detail can be found from

<http://academic.scranton.edu/faculty/dougherty1/>

- N.J.A. Sloane offers \$10 (1973)—still valid (confirmed in 2006).
- F.J. MacWilliams offered \$10 (1977)—invalid now.

The following monetary prizes were announced in the Yamagata conference, October, 2000, and at WCC2001 in Paris.

- S.T. Dougherty offers \$100 for the existence of  $C(3)$ .
- M. Harada offers \$200 for the nonexistence of  $C(3)$ .

The prize is awarded only once and the result must be published in a refereed reputable mathematics journal. All decisions about the prize are decided by those offering the prize.

**Acknowledgements** The author would like to thank the Editorial Board of the book and, in particular, Massimiliano Sala.

The author acknowledges support from the Austrian Academy of Sciences during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- J. H. Conway and N. J. A. Sloane, *A new upper bound on the minimal distance of self-dual codes*, IEEE Trans. on Inf. Th. **36** (1990), no. 6, 1319–1333.
- S. T. Dougherty, J. L. Kim, and P. Solé, *Double circulant codes from two class association schemes*, Adv. Math. Commun. **1** (2007), no. 1, 45–64.
- E. Guerrini and M. Sala, *An algebraic approach to the classification of some non-linear codes*, Proc. of WCC 2007 INRIA (2007), 177–185.
- T. A. Gulliver, M. Harada, and J. L. Kim, *Construction of new extremal self-dual codes*, Discrete Math. **263** (2003), nos. 1–3, 81–91.
- S. K. Houghten, C. W. H. Lam, L. H. Thiel, and J. A. Parker, *The extended quadratic residue code is the only (48, 24, 12) self-dual doubly-even code*, IEEE Trans. on Inf. Th. **49** (2003), no. 1, 53–59.
- W. C. Huffman, *On the classification and enumeration of self-dual codes*, Finite Fields Appl. **11** (2005), no. 3, 451–490.
- I. Kotsireas, <http://www.medicis.polytechnique.fr/~kotsirea/>, 2006.
- J. L. Kim and P. Solé, *Skew Hadamard designs and their codes*, Des. Codes Cryptogr. **49** (2008), nos. 1–3, 1–11.
- G. Nebe, E. M. Rains, and N. J. A. Sloane, *Self-dual codes and invariant theory*, Algor. and Comput. in Math., vol. **17**, Springer, Berlin, 2006.
- V. Pless, *On the uniqueness of the Golay codes*, J. Combinatorial Theory **5** (1968), 215–228.
- V. S. Pless, W. C. Huffman, and R. A. Brualdi (eds.), *Handbook of Coding Theory*, vols. **I**, **II**, North-Holland, Amsterdam, 1998.
- E. M. Rains, *Shadow bounds for self-dual codes*, IEEE Trans. on Inf. Th. **44** (1998), no. 1, 134–139.

- E. M. Rains and N. J. A. Sloane, *Self-dual codes*, Handbook of Coding Theory (V. Pless and W.C. Huffman, eds.), Elsevier, Amsterdam, 1998, pp. 177–294.
- N. J. A. Sloane, *Is there a  $(72, 36)d = 16$  self-dual code?* IEEE Trans. on Inf. Th. **19** (1973), no. 2, 251.
- S. Zhang, *On the nonexistence of extremal self-dual codes*, Discrete Appl. Math. **91** (1999), nos. 1–3, 277–286.

# An Application of Möller's Algorithm to Coding Theory

M. Borges-Quintana, M.A. Borges-Trenard and  
E. Martínez-Moro

**Abstract** We show the use of Möller's Algorithm and related techniques for decoding and studying some combinatorial properties of linear codes. It is a concise summary of our previous results, with emphasis in illustrating the applications and comparing the developed method for computing the Gröbner basis associated with the code with the classical way to solve the same problem.

## 1 Introduction

The connection between Gröbner bases and linear algebra comes from the very beginning, i.e. from Buchberger's (1965, 2006) PhD thesis. In Faugère et al. (1993), Marinari et al. (1993) these techniques were generalized to different settings (change of orderings, ideal defined by functionals). In Borges-Quintana et al. (2006b, 2007) the algorithm for monoid and group algebras was specialized for the case of algebras associated to linear codes. This work is a concise presentation of some results in Borges-Quintana et al. (2008, 2006b, 2007).

## 2 An Ideal Associated with a Linear Code

A polynomial having the shape  $\tau_1 - \tau_2, \tau_i$  terms, is called *binomial*. A *binomial ideal* is an ideal generated by binomials. If an ideal  $I$  is binomial, then for each  $\tau \in T$ ,  $\text{Can}(\tau, I) \in N(I)$ ; in particular Gröbner and border bases consist of binomials. Therefore, if  $I \subset P$  is a binomial 0-dimensional ideal, denoting  $N(I) = \{\tau_0 = 1\}$ ,

---

M. Borges-Quintana · M.A. Borges-Trenard  
Dpto. de Matemática, FCMC, U. de Oriente, Santiago de Cuba, Cuba  
e-mail: [mijail@csd.uo.edu.cu](mailto:mijail@csd.uo.edu.cu)

M.A. Borges-Trenard  
e-mail: [mborges@csd.uo.edu.cu](mailto:mborges@csd.uo.edu.cu)

E. Martínez-Moro  
Dpto. de Matemática Aplicada, U. de Valladolid, Valladolid, Spain  
e-mail: [edgar@maf.uva.es](mailto:edgar@maf.uva.es)

$\tau_1, \dots, \tau_s\}$ , we have

1.  $\forall \ell, 1 \leq \ell \leq s, \exists ! h, l, 1 \leq h \leq n, 0 \leq l < s : h = \min\{i : X_i \mid \tau_\ell\}, \tau_\ell = X_h \tau_l.$
2.  $\forall h, l, 1 \leq h \leq n, 1 \leq l \leq s, \exists ! \ell : \text{Can}(X_h \tau_l, \mathbf{l}) = \tau_\ell.$

Such data have been applied to correct binary linear codes in Borges-Quintana et al. (2008, 2006b, 2007), where a linear  $[n, k]$ -code is encoded by expressing its generator matrix  $(a_{ij})$  by the binomial ideal  $\mathbf{l} = \langle \{\prod_{j=1}^n X_j^{a_{ij}} - 1, 1 \leq i \leq k\} \cup \{x_j^2 - 1, 1 \leq j \leq n\} \rangle$  and each codeword  $(a_1, \dots, a_n) \in \mathbb{F}_2^n$  as  $\prod_{j=1}^n X_j^{a_j}$ ; hence for any codeword  $\tau \in \mathcal{T}$ , the maximum likelihood decoding error is  $\text{Can}(\tau, \mathbf{l}, \prec)$ , where  $\prec$  is a total degree compatible term ordering. Thus we do three things, as follows. We use an improved version of Möller's (2009) algorithm for binomial ideals to deduce a reduced Gröbner basis, a border basis or a Gröbner representation for  $\mathbf{l}$  w.r.t. a total degree compatible ordering. We decode codewords using such data to compute the maximum likelihood decoding error. We use the data structure to solve other problems related with the combinatorics of a linear code.

## 2.1 A Second Way of Getting the Data for $\mathbf{l}$

A pattern algorithm was presented in Borges-Trenard et al. (2000) for the free monoid algebra, we will restrict here to the commutative case. Let  $M$  be a finite commutative monoid generated by  $g_1, \dots, g_n$ ;  $\xi : [X] \rightarrow M$ , the canonical morphism that sends  $X_i$  to  $g_i$ ;  $\sigma \subset [X] \times [X]$ , a presentation of  $M$  defined by  $\xi$  ( $\sigma = \{(w, v) \mid \xi(w) = \xi(v)\}$ ). Then, it is known that the monoid ring  $k[M]$  is isomorphic to  $\mathcal{P}/I(\sigma)$ , where  $I(\sigma)$  is the ideal generated by  $\{w - v \mid (w, v) \in \sigma\}$ ; moreover, any Gröbner basis  $G$  of  $I(\sigma)$  is also formed by binomials of the above form. In addition, it can be proved that  $\{(w, v) \mid w - v \in G\}$  is another presentation of  $M$ . Note that  $M$  is finite if and only if  $I(\sigma)$  is zero-dimensional.

For a binary code defined by a parity check matrix  $H$ , the monoid  $M$  is set to  $(\mathbb{F}_2)^{n-k}$  (the syndromes space) and  $g_i := \xi(X_i) = e_i H$  ( $e_i$  is the  $i$ -th coordinate vector in  $\mathbb{F}_2^n$ ). Note that  $M = \mathbb{F}_2^{n-k} = \langle g_1, \dots, g_n \rangle$  and  $I(\sigma) = \mathbf{l}$ . Starting from this setting, we can compute a reduced Gröbner basis, a border basis or a Gröbner representation for a total degree compatible ordering (Borges-Quintana et al. 2006b).

For a general linear code over  $\mathbb{F}_q$  ( $q = p^m$ ), where  $\alpha$  is a root of an irreducible polynomial of degree  $m$  over  $F_p$ ,  $H$  a parity check matrix of the code, each vector  $y = (\sum_{j=1}^m \beta_{1j} \alpha^{j-1}, \dots, \sum_{j=1}^m \beta_{nj} \alpha^{j-1}) \in \mathbb{F}_q^n$  is encoded as  $w = \prod_{i=1}^n \prod_{j=1}^m x_{ij}^{\beta_{ij}}$  (note that  $\psi(w) = y$  defines a surjective morphism  $\psi : [X] \rightarrow \mathbb{F}_q^n$ ). Now by extending the analysis of the binary case, let the monoid  $M = \mathbb{F}_q^{n-k}$  and let  $g_{ij} = \xi(X_{ij}) = \psi(X_{ij})H$ , note that  $M = \mathbb{F}_q^{n-k} = \langle g_{11}, \dots, g_{nm} \rangle$  and the ideal  $I(\sigma)$  coincides with the binomial ideal constructed from a generator matrix. In fact, the *binomial ideal associated with the code* is well defined despite the way it is constructed.

$$\mathbf{l} = \langle \{\tau_1 - \tau_2 \mid \psi(\tau_1)H = \psi(\tau_2)H, \tau_1, \tau_2 \in \mathcal{T}\} \rangle \subset \mathcal{P}.$$

To get the data structure that enables to solve the decoding problem we introduce the error vector ordering  $\prec_e$ , which is not a semigroup ordering. The set of canonical forms for  $\prec_e$  associated to the code need to be redefined as  $N = \{1 = \tau_1, \dots, \tau_{q^{n-k}}\} \subset \mathcal{T}$  such that

1. If  $\tau_1, \tau_2 \in N$  and  $\tau_1 \neq \tau_2$  then  $\xi(\tau_1) \neq \xi(\tau_2)$ .
2. For all  $\tau \in N \setminus \{1\}$  there exists  $X_i$  such that  $\tau = X_i \tau'$  and  $\tau' \in N$ .

In this case we can compute a Gröbner representation that can be used for canonical form computation. For this setting of general linear codes see Borges-Quintana et al. (2007).

We provide now some examples. All these applications are implemented in The GAP Group (2008) as a collection of functions that we have called GBLA\\_LC (Borges-Quintana et al. 2006a).

### 3 Examples

#### 3.1 Working out with a Gröbner Representation

For the binary code whose parity check matrix is

$$H^T := \begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$$

Since the code is binary,  $\prec_e$  is a degree compatible term ordering and we have  $N = \mathbf{N}(\mathbf{l}) = \{1, X_1, X_2, X_3, X_4, X_5, X_6, X_1 X_6\}$ , we encode this set as  $\tau_\ell = X_h \tau_l$  in the table

$\ell$	1	2	3	4	5	6	7
$h$	0	0	0	0	0	0	1
$l$	1	2	3	4	5	6	6

and whose corresponding FGLM-matrix is  $\text{Can}(X_h \tau_l, \mathbf{l}) = \tau_\ell$

	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1

For example, when the message  $X_2 X_3 X_6$  arrives

- read it and run on the second matrix we get the encoded error:

$$0 \xrightarrow{2} 2 \xrightarrow{3} 1 \xrightarrow{6} 7.$$

Note that the position 0 is always the starting point of the encoded process of the error. Then the canonical form of the received message is  $X_1 X_6$  (the last element of  $N(I)$ ). Note that the error capability of this code is 1 and in this case we got an error with weight 2.

- The maximum likelihood codeword is  $\psi(X_2 X_3 X_6) - \psi(X_1 X_6) = (1, 1, 1, 0, 0, 0)$ .

### 3.2 Combinatorial Properties of a Binary Code

Let us defined a [10, 4]-code over  $\mathbb{F}_2$ , with 16 codewords and 64 canonical forms by the parity check matrix

$$\mathbf{H} := \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}.$$

Starting from  $H$  we compute  $\mathcal{G}$  the reduced Gröbner basis of  $I$  for the degree reverse lexicographic ordering ( $\prec_{Drl}$ ). We show in the examples below some information about a binary code that one can get from the associated reduced Gröbner basis, in addition of another way of decoding.<sup>1</sup>

1. After the subset  $B := \{X_i^2 - 1 \mid 1 \leq i \leq 10\}$  the binomials of less degree  $D$  are those of degree  $D = 2$ , i.e.:

$$\begin{aligned} &\{X_2 X_5 - X_1 X_6, X_2 X_6 - X_1 X_5, X_3 X_5 - X_1 X_7, X_3 X_6 - X_2 X_7, \\ &X_3 X_7 - X_1 X_5, X_4 X_5 - X_1 X_8, X_4 X_6 - X_2 X_8, X_4 X_7 - X_3 X_8, \\ &X_4 X_8 - X_1 X_5, X_5 X_6 - X_1 X_2, X_5 X_7 - X_1 X_3, X_5 X_8 - X_1 X_4, \\ &X_6 X_7 - X_2 X_3, X_6 X_8 - X_2 X_4, X_7 X_8 - X_3 X_4\}. \end{aligned}$$

Since Borges-Quintana et al. (2007) the error-correcting capability  $t$  of the code is  $D - 1$  we deduce  $t = 1$ . Moreover the minimal distance of the code is  $d := \min\{\deg(u) + \deg(v) : u - v \in \mathcal{G} \setminus B\} = 4$  and these binomials are associated

---

<sup>1</sup>Although for this purpose the Gröbner representation is more efficient.

with codewords of minimal distance (Borges-Quintana et al. 2008), the vectors  $\{\psi(u) - \psi(v) : u - v \in \mathcal{G} \setminus B, \deg(u) + \deg(v) = d\}$ , namely:

$$\begin{aligned} & \{(1, 1, 0, 0, 1, 1, 0, 0, 0, 0), (1, 0, 1, 0, 1, 0, 1, 0, 0, 0), \\ & (0, 1, 1, 0, 0, 1, 1, 0, 0, 0), (1, 0, 0, 1, 1, 0, 0, 1, 0, 0), \\ & (0, 1, 0, 1, 0, 1, 0, 1, 0, 0), (0, 0, 1, 1, 0, 0, 1, 1, 0, 0)\}. \end{aligned}$$

The basis has 46 binomials.

2. *Decoding* (see Borges-Quintana et al. 2007): let us take now as a received vector  $v = (1, 1, 1, 1, 0, 0, 0, 0, 1, 1)$ , encoding  $w = X_1X_2X_3X_4X_9X_{10}$ . Let us reduce now  $w$  using the reduced Gröbner basis, by  $w_1 \xrightarrow{g} w_2$  we mean  $w_1$  is reduced to  $w_2$  modulo the polynomial  $g$  of  $\mathcal{G}$ .

$$w = x_1x_2x_3x_4x_9x_{10} \xrightarrow{x_2x_3x_4-x_9x_{10}} x_1x_9^2x_{10}^2 \xrightarrow{x_9^2-1} x_1x_{10}^2 \xrightarrow{x_{10}^2-1} x_1$$

$\text{weight}(\psi(x_1)) = 1$ , then the codeword is  $(0, 1, 1, 1, 0, 0, 0, 0, 1, 1)$ .

3. *Decomposition of a codeword* (see Borges-Quintana et al. 2008): let  $c = (1, 0, 0, 0, 1, 1, 1, 1, 1, 1)$ , encoding  $w_c = X_1X_5X_6X_7X_8X_9X_{10}$ . Reducing modulo  $\mathcal{G}$  and removing all occurrences of  $X_i^2$  in the reductions:

$w_c \xrightarrow{g_1} X_1X_2X_5X_6 \xrightarrow{g_2} 1$ , where  $g_1 = x_7x_9x_{10} - x_2x_8$ ,  $g_2 = x_2x_5 - x_1x_6$ . These sequence implies that  $c$  can be expressed as  $c = c_{g_1} + c_{g_2}$ , where  $c_{g_1}$  and  $c_{g_2}$  are the codewords associated to these binomials.

### 3.3 Example: the Golay Code

We use the GAP package GUAVA to construct a generator matrix of the Golay [23,12] code.

First we try our Möller's Algorithm approach to linear codes that is implemented in GAP Borges-Quintana et al. (2006a). We have got the reduced Gröbner basis of the ideal  $I$  for the ordering  $\prec_{\text{Drl}}$  in less than 15 minutes (with a Pentium 1.5 GHz).

From a generator matrix of the code, we use the first method for obtaining the ideal  $I$ , having this generating set and using the ordering  $\prec_{\text{Drl}}$  we have tried to compute the reduced Gröbner basis for  $I$  in **Mathematica**, **Maple**, and **GAP** and the computing process was interrupted after 4 hours. On the other hand, **Singular** succeeded in 2 hours.

### 3.4 GAP Computing Section

We write the sequence of commands used to get the reduced Gröbner basis.

```
gap> n:=23;;k:=12;;m:=1;;p:=2;;F:=GF(2);;
gap> alpha_prim:=RootOfDefiningPolynomial(GF(2));;
gap> alpha_ext:=RootOfDefiningPolynomial(F);;
```

```

gap> R:=PolynomialRing(Rationals,n*m) ;
      x:=IndeterminatesOfPolynomialRing(R) ;
gap> Read("D://gbla_lc.txt") ;
gap> LoadPackage( "guava", "2.4" ) ;
gap> H:=CheckMat(BinaryGolayCode()) ;
gap> Gr:=Greduce1(H,m,n,k,p) ;

```

There are 253 codewords associated to the binomials of  $\text{Gr}$ , but the Golay code have 2048 syndromes and 4096 codewords.

**Acknowledgement** The authors acknowledge support from the Austrian Academy of Science during the Semester on Gröbner bases (Linz, 2006).

## References

- M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro, *GBLA\_LC: Gröbner basis by linear algebra and linear codes*, 2006a, <http://www.math.arq.uva.es/~edgar/GBLAweb/>.
- M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro, *A general framework for applying FGLM techniques to linear codes*, LNCS, vol. **3857**, Springer, Berlin, 2006b, pp. 76–86.
- M. Borges-Quintana, M. A. Borges-Trenard, and E. Martínez-Moro, *On a Gröbner bases structure associated to linear codes*, J. Discrete Math. Sci. Cryptogr. **10** (2007), no. 2, 151–191.
- M. Borges-Quintana, M. A. Borges-Trenard, P. Fitzpatrick, and E. Martínez-Moro, *Gröbner bases and combinatorics for binary codes*, Appl. Algebra Engrg. Comm. Comput. **19** (2008), no. 5, 393–411.
- M. A. Borges-Trenard, M. Borges-Quintana, and T. Mora, *Computing Gröbner bases by FGLM techniques in a non-commutative setting*, J. Symbolic Comput. **30** (2000), no. 4, 429–449.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- J. C. Faugère, P. Gianni, D. Lazard, and T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, J. Symbolic Comput. **16** (1993), no. 4, 329–344.
- M. G. Marinari, H. M. Möller, and T. Mora, *Gröbner bases of ideals defined by functionals with an application to ideals of projective points*, AAECC **4** (1993), no. 2, 103–145.
- T. Mora, *The FGLM problem and Möller's algorithm on zero-dimensional ideals*, this volume, 2009, pp. 27–45.
- The GAP Group, GAP—groups, algorithms, and programming*, version 4.4.12, 2008, <http://www.gap-system.org>.

# Mattson Solomon Transform and Algebra Codes

Edgar Martínez-Moro and Diego Ruano

**Abstract** In this note we review some results of the first author on the structure of codes defined as subalgebras of a commutative semisimple algebra over a finite field (see Martínez-Moro in Algebra Discrete Math. 3:99–112, 2007). Generator theory and those aspects related to the theory of Gröbner bases are emphasized.

## Introduction

Some classical code constructions can be seen as ideals in a finite-dimensional commutative semisimple algebra  $\mathcal{A}$  (from now on, we denote it briefly by algebra) over a finite field  $\mathbb{F}_q$  with  $q = p^r$  elements and  $p$  prime (note that since  $\mathbb{F}_q$  is a perfect field  $\mathcal{A}$  is a separable algebra). Our notation on linear and cyclic codes follows (Augot et al. 2009). Consider a basis of  $\mathcal{A}$  as a  $\mathbb{F}_q$ -vector space given by  $\mathfrak{B} = \{b_1 = 1, \dots, b_n\}$ ,  $\mathcal{A}$  is equipped with a multiplication defined by the convolutional-like product given by

$$\left( \sum_{i=1}^n \alpha_i b_i \right) \left( \sum_{i=1}^n \alpha'_i b_i \right) = \left( \sum_{i=1}^n \alpha_i \alpha'_i b_{ij} \right), \quad \alpha_i, \alpha'_i \in \mathbb{F}_q, \text{ for } i = 1, \dots, n \quad (1)$$

where the  $b_{ij}$  correspond to the multiplication table of  $\mathcal{A}$  for the basis  $\mathfrak{B}$ , i.e.  $b_{ij} = b_i b_j = \sum_{k=1}^n m_{i,k} b_k$ , for  $1 \leq i, j, k \leq n$ ,  $m_i \in \mathbb{F}_q$ . For example, if  $G$  is a commutative finite group of order  $n$  with identity element 1 and  $\gcd(q, n) = 1$  (condition for semisimplicity) the group algebra  $\mathbb{F}_q[G]$  consists of elements of the form  $\sum_{g \in G} \alpha_g g$  (i.e.  $G$  is the basis) with  $\alpha_g \in \mathbb{F}_q$ , and the convolutional product is  $\sum_{g \in G} \alpha_g g \sum_{g \in G} \alpha'_g g = \sum_{g \in G} (\sum_{h \in G} \alpha_h \alpha'_{gh^{-1}}) g$ . For instance, for the cyclic

---

First author was partially supported by the Spanish MICINN through projects MTM2007-66842, MTM2007-64704, by AECI project A/7745/07 and by Junta de CyL project VA065A07. Second author was partially supported by the Spanish MICINN through project MTM2007-64704, by Junta de CyL project VA065A07 and by DASMOD-Cluster of Excellence in Rhineland-Palatinate (Germany).

E. Martínez-Moro

Departamento de Matemática Aplicada, Universidad de Valladolid, Valladolid, Castilla, Spain

e-mail: [edgar@maf.uva.es](mailto:edgar@maf.uva.es)

D. Ruano

Fachbereich Mathematik, Technische Universität Kaiserslautern, Kaiserslautern, Germany

e-mail: [ruano@mathematik.uni-kl.de](mailto:ruano@mathematik.uni-kl.de)

group of order  $n$ , i.e.  $G = C_n$ , one has the cyclic codes. We finally show how the well-known polynomial generator theory for cyclic codes can be extended to any finite-dimensional commutative semisimple algebra, following the main ideas in Martínez-Moro (2007).

## 1 Mattson–Solomon Transform

We consider *structure polynomials* of the algebra  $\mathcal{A}$  as the set of polynomials in  $\mathbb{F}_q[x_1, \dots, x_n]$  given by ( $m_i$  as before)

$$F = \left\{ x_i x_j - \left[ m_{i,1} + \sum_{k=2}^n m_{i,k} x_k \right] \right\}_{2 \leq i \leq j \leq n} \cup \{x_1 - 1\}. \quad (2)$$

Note that  $F$  is a Gröbner basis with respect to a monomial ordering compatible with the total-degree and it turns out (see Martínez-Moro 2004 for further details) that  $\mathcal{A} \cong \mathbb{F}_q[x_1, \dots, x_n]/\langle F \rangle$ , where  $b_i \mapsto x_i$  is an algebra isomorphism. Let  $\mathcal{V}(F) = (P_1, P_2, \dots, P_n)$  be the points in the variety defined by  $F$ , i.e. the roots of the system of the equations in  $F$  in some field extension  $\mathbb{F}$  (large enough) of  $\mathbb{F}_q$ . Denote the  $P_i$  by  $P_i = (p_{i1}, \dots, p_{in})$  as row vectors. We consider the *Mattson–Solomon* matrix  $M_{\mathcal{A}}$  defined as  $(M_{\mathcal{A}})_{ij} = p_{ij}$  which is non-singular and for  $a(\mathbf{x}) \in \mathbb{F}[x_1, \dots, x_n]/\langle F \rangle$  the map

$$\Theta : \mathbb{F}[x_1, \dots, x_n]/\langle F \rangle \rightarrow \mathbb{F}[x_1, \dots, x_n]/\langle F \rangle, \quad [\Theta(a)](\mathbf{x}) = \sum_{i=1}^n a(P_i) x_i \quad (3)$$

is the Mattson–Solomon transform. Let  $\circ$  be the multiplication of polynomials modulo the ideal  $\langle F \rangle$  and  $\star$  the component wise product  $(\sum a_i x_i) \star (\sum b_i x_i) = (\sum a_i b_i x_i)$  then one has that the map

$$\Theta : (\mathbb{F}[x_1, \dots, x_n]/\langle F \rangle, +, \circ) \rightarrow (\mathbb{F}[x_1, \dots, x_n]/\langle F \rangle, +, \star), \quad x_i \mapsto x_i$$

is a ring isomorphism (see, for example, Chillag 1995 for a proof) corresponding to the diagonalization by  $M_{\mathcal{A}}$  of the regular representation of  $a \in \mathcal{A}$ , i.e. if  $ab_i = \sum_{k=1}^m a_{ik} b_k$  then the product  $(a_{ij}) \cdot M_{\mathcal{A}}$  is a diagonal matrix. (Note that for the cyclic case  $\mathcal{A} = \mathbb{F}_q[C_n]$  and  $M_{\mathcal{A}} = (\xi^{ij})_{1 \leq i, j \leq n}$ , where  $\xi$  is a primitive  $n$ -th root of unity.)

## 2 Generator Theory

The purpose of Martínez-Moro (2007) is to devise a root-free theory (therefore there is no need to compute  $M_{\mathcal{A}}$ ) by generalizing the generator polynomial or parity-check polynomial construction of cyclic codes. Fix a field extension  $\mathbb{F}$  of  $\mathbb{F}_q$  and

consider the set

$$F' = F \cup \{g_1(x_1), \dots, g_n(x_n)\} \quad (4)$$

where  $F$  is the Gröbner basis in equation (2) associated to the algebra  $\mathcal{A}$  and  $g(x_i)$  is a factor  $g_i(x_i)|m_{\mathbb{F}}(x_i)$ , where  $m_{\mathbb{F}}(x_i)$  is the minimal polynomial of the element in  $\mathcal{A}$  corresponding to  $x_i$ , i.e.  $m_{\mathbb{F}}(x_i)$  is the generator of the elimination ideal  $\langle F \rangle \cap \mathbb{F}[x_i]$ ,  $1 \leq i \leq n$ . Note that possibly some or even every  $g_i(x_i)$  could be  $m_{\mathbb{F}}(x_i)$  and indeed, the factorization of this polynomial depends on the splitting field between  $\mathbb{F}_q$  and  $\mathbb{F}$  considered. A linear algebra procedure for computing  $m_{\mathbb{F}}(x_i)$  from the Gröbner basis in (2) can be found in (Martínez-Moro 2004). We call the ideal  $\langle F' \rangle$  the *generator ideal* of the code  $\mathcal{C} = \mathbb{F}_q[x_1, \dots, x_n]/\langle F' \rangle \subseteq \mathbb{F}_q[x_1, \dots, x_n]/\langle F \rangle \cong \mathbb{F}_q^n$  (where  $\subseteq$  is as  $\mathbb{F}_q$ -vector spaces) and the ideal  $(F')^\perp = F \cup \{h_i(x_i)\}_{i=1}^n$ , with  $h_i(x_i) = m_{\mathbb{F}_q}(x_i)/g_i(x_i)$  the *parity-check ideal* of the code  $\mathcal{C}$ . The points of the variety  $V((F')^\perp)$  considered as row vectors form a pseudo-parity-check matrix of the code (we say pseudo since the values of these points may not lie in  $\mathbb{F}_q$ ). Note that, as in the classical theory of cyclic codes, what we are considering as codes is the preimage by  $\Theta$  of the subalgebra given by the elements with some fixed zero positions in the Mattson–Solomon codomain.

The footprint (also called the Hilbert escalier) of  $\langle F' \rangle$  w.r.t. the ordering  $<$  is the set of monomials  $\Delta_{<}(\langle F' \rangle)$  in  $\mathbb{F}_q[x_1, \dots, x_n]$  that are not leading monomials in  $\langle F' \rangle$ . Since  $F'$  is a radical ideal (see Martínez-Moro 2004),  $|\Delta_{<}(\langle F' \rangle)|$  gives us the size of the variety  $V(\langle F' \rangle)$  in the algebraic closure of  $\mathbb{F}_q$  and therefore the dimension  $k$  of  $\mathbb{F}_q[x_1, \dots, x_n]/\langle F' \rangle$  as vector space.

Note that if there exists an element  $x_i$  such that the degree of  $m_{\mathbb{F}}(x_i)$  is  $n$  then  $\mathbb{F}_q[x_i]/\langle m_{\mathbb{F}}(x_i) \rangle$  and  $\mathbb{F}_q[x_1, \dots, x_n]/\langle F \rangle$  are isomorphic as vector spaces, in other words, the rest of the variables can be seen as polynomials in the variable  $x_i$  of degree less than or equal to  $n$  (one has this for cyclic codes). We call this element  $x_i$  (if it exists) *separating element*. The following table summarizes the above discussion and its comparison to cyclic codes

Cyclic codes	Semisimple codes
$\mathbb{F}_q[x]/\langle x^n - 1 \rangle$	$\mathcal{A} \cong \mathbb{F}_q[x_1, \dots, x_n]/\langle F \rangle$
$g(x) (x^n - 1)$	$F' = F \cup \{g_i(x_i)\}_{i=1}^n, g_i(x_i) m_{\mathbb{F}_q}(x_i)$
$h(x) = (x^n - 1)/g(x)$	$F'^\perp = F \cup \{h_i(x_i)\}_{i=1}^n, h_i(x_i) = m_{\mathbb{F}_q}(x_i)/g_i(x_i)$
$k = \deg g(x)$	$k =  \Delta(\langle F' \rangle) $

Bounds of BCH-type and Hartmann-Tzeng-Roos type can be established in terms of the roots of the polynomials  $g_i$  for  $i = 1, \dots, n$ , see Martínez-Moro (2007).

### 3 A Note on the Syndrome Variety

Given a semisimple code  $\mathcal{C}$  defined by a generator ideal  $\langle F \rangle$  one has that if  $r = \sum_{i=1}^n r_i x_i$  is a received word then the syndromes of  $r$  are  $s_j = \sum_{i=1}^n r_i P_{ji}$  where  $j$  ranges in some of the rows of the Mattson–Solomon matrix of the algebra that corresponds to the pseudo-parity-check matrix of the code. If all the syndromes are zero then  $r$  belongs to the code. Suppose that  $t$  errors have occurred and let  $\mathcal{G}$  be a Gröbner basis of  $\langle F \rangle$ , we consider the equations  $f_j = \sum_{l=1}^t y_l x_l - z_j$ ,  $\sigma_j = z_j^{q^m} - z_j$  and  $\lambda_i = y_i^{q-1} - 1$ , with indices depending on the field extension considered, where the  $z_i$ 's represent the syndromes and the  $y_i$ 's the error values. The variety generated by the polynomials

$$\{f_j, \sigma_j, \lambda_i\}_{i,j} \cup \mathcal{G} \quad (5)$$

is the *Chen-Reed-Helleseth-Truong syndrome variety* of the code used in the cyclic case for decoding (Loustaunau and York 1997) and for finding the minimum distance (Sala 2002). This variety is analyzed in the book chapter (Mora and Orsini 2009), where also a modified syndrome variety (for cyclic codes) can be found (see both Mora and Orsini 2009 and the references therein for a detailed study). A future line of research includes the study of this variety for semisimple algebra codes and the structure of its general error locator ideal.

**Acknowledgement** The authors acknowledge support from the Austrian Academy of Sciences during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- D. Augot, E. Betti, and E. Orsini, *An introduction to linear and cyclic codes*, this volume, 2009, pp. 47–68.
- D. Chillag, *Regular representations of semisimple algebras, separable field extensions, group characters, generalized circulants, and generalized cyclic codes*, Linear Algebra Appl. **218** (1995), 147–183.
- P. Loustaunau and E. V. York, *On the decoding of cyclic codes using Gröbner bases*, AAECC **8** (1997), no. 6, 469–483.
- E. Martínez-Moro, *Regular representations of finite-dimensional separable semisimple algebras and Gröbner bases*, J. Symbolic Comput. **37** (2004), no. 5, 575–587.
- E. Martínez-Moro, *On semisimple algebra codes: generator theory*, Algebra Discrete Math. **3** (2007), 99–112.
- T. Mora and E. Orsini, *Decoding cyclic codes: the Cooper philosophy*, this volume, 2009, pp. 69–91.
- M. Sala, *Gröbner bases and distance of cyclic codes*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 2, 137–162.

# Decoding Folded Reed–Solomon Codes Using Hensel-Lifting

Peter Beelen and Kristian Brander

**Abstract** A standard problem in coding theory is to construct good codes together with an efficient decoder. This paper addresses the construction of a class of codes (folded RS codes) for which one can give an efficient and (in a certain sense) optimal decoder, by adapting a list decoding algorithm.

## 1 Introduction

A standard problem in coding theory is to construct good codes together with an efficient decoder. This paper addresses the construction of a class of codes for which one can give an efficient and in a certain sense optimal decoder. A Reed–Solomon code of rate  $R$  can be list decoded up to a relative distance of  $1 - \sqrt{R}$  using the Guruswami–Sudan algorithm (Guruswami and Sudan 1999; Guerrini and Rimoldi 2009) (GS). On the other hand it is known that a code of rate  $R$  cannot be list decoded beyond a relative distance of  $1 - R$ , and an immediate question is whether one can construct decoders with performances in the gap between  $1 - \sqrt{R}$  and  $1 - R$ . In Guruswami and Rudra (2008) and Parvaresh and Vardy (2005) it was shown that using a folding construction on Reed–Solomon codes, similarly to the one used in Krachkovsky (2003), one can obtain codes with a list decoder, able to correct errors up to a relative distance of approximately

$$1 - R^{\frac{s}{s+1}}, \quad (1)$$

where  $s \geq 1$  is an integer parameter of the construction.

In this paper we sketch this construction and in more detail we show how to decode them up to the bound in (1). The resulting decoder is faster experimentally than the decoder in Guruswami and Rudra (2008). First of all, we outline how our decoder works. The overall approach is the same as in the GS algorithm: first we find an interpolation polynomial  $Q$  (as in Definition 2) and next we compute a certain type of roots of  $Q$  (similarly to Guruswami and Rudra 2008). Each

---

P. Beelen · K. Brander  
DTU Mathematics, Technical University of Denmark, Matematiktorvet 303S, 2800  
Kgs. Lyngby, Denmark

P. Beelen  
e-mail: [P.Beelen@mat.dtu.dk](mailto:P.Beelen@mat.dtu.dk)

K. Brander  
e-mail: [K.Brander@mat.dtu.dk](mailto:K.Brander@mat.dtu.dk)

such root corresponds to a codeword and the transmitted word is guaranteed to be among those. The main difference from the GS algorithm is that the interpolation polynomial is allowed to be *multivariate*, or more specifically of the form  $Q(x, z_1, \dots, z_s)$  (the  $s$  here is the same as in (1)). Due to the way the interpolation polynomial is chosen, it holds that if not too many errors occur, the transmitted word will be among the roots  $f(x)$  of the interpolation polynomial, of the form  $Q(x, f(x), f(\gamma x), \dots, f(\gamma^{s-1}x)) = 0$ , where  $\gamma$  is a primitive element (Guruswami and Rudra 2008). These roots are computed efficiently via Hensel-lifting (Beelen and Brander 2007).

## 2 Folded Reed–Solomon Codes

Informally, a folded Reed–Solomon (RS) code is a RS code over some  $\mathbb{F}_q$ , but viewed as a code over a larger alphabet by identifying consecutive  $m$  positions in the RS code as elements in  $\mathbb{F}_{q^m}$ .

**Definition 1** (Folded RS code) Given  $q, m, N, k$  s. t.  $mN \leq q - 1$ . Let  $\gamma$  be a primitive element of  $\mathbb{F}_q$  and let  $f(x) \in \mathbb{F}_q[x]$  with degree at most  $k - 1$ , then the folded RS code consists of all  $m \times N$  arrays of type

$$\begin{pmatrix} f(1) & f(\gamma^m) & \cdots & f(\gamma^{m(N-1)}) \\ f(\gamma) & f(\gamma^{m+1}) & \cdots & f(\gamma^{m(N-1)+1}) \\ \vdots & \vdots & \ddots & \vdots \\ f(\gamma^{m-1}) & f(\gamma^{2m-1}) & \cdots & f(\gamma^{mN-1}) \end{pmatrix}.$$

Using any fixed identification of  $(\mathbb{F}_q)^m$  with  $\mathbb{F}_{q^m}$ , we can consider the columns of the above array as elements of  $\mathbb{F}_{q^m}$ , and therefore we can consider the folded RS code as a code of length  $N$  over  $\mathbb{F}_{q^m}$ .

At this stage, it is not clear why it is an advantage to fold the RS codes, but as we shall see, this is exactly what makes the multivariate extension of the GS algorithm (Guruswami and Sudan 1999) work. We record some properties of the folded code.

**Proposition 1** *The folded RS code, with parameters as in Definition 1, is a (non-linear) code over  $\mathbb{F}_{q^m}$  of length  $N$ , rate  $R = \frac{k}{Nm}$  and minimum distance  $d = N - \lceil \frac{k}{m} \rceil + 1$ .*

## 3 Decoding of Folded Reed–Solomon Codes

In this section we describe a list decoder for the folded RS codes. As mentioned in the introduction, the decoder proceeds by first computing an interpolation polynomial, and next to compute certain roots of this. We first introduce the interpolation

polynomial. This will depend on an *interpolation parameter*  $s$  satisfying  $s \leq m$ , and the interpolation polynomial will be an element in  $\mathbb{F}_q[x, z_1, \dots, z_s]$ . We will need a special weighted degree on polynomials in this ring, namely the one defined by

$$w_{\deg} \left( x^i z_1^{j_1} \cdots z_s^{j_s} \right) = i + (k - 1) \sum_{l=1}^s j_l.$$

With these notions we can now describe the interpolation polynomial. Similarly to the GS algorithm the first variable  $x$  plays a special role and interpolates through the powers of  $\gamma$  (the primitive element in  $\mathbb{F}_q$ ), while the remaining variables interpolate through the positions in the received word. In the folded RS code each position consists of  $m$  subpositions from an ordinary RS code, and now we let the variables  $z_1, \dots, z_s$  interpolate through  $s$  consecutive subpositions for each position in the folded code. The definition of the interpolation polynomial depends on a *multiplicity parameter*  $r$ , which determines the multiplicity with which the polynomial is required to have zeroes at the interpolation points.

**Definition 2** (Interpolation polynomial) Let  $w$  be the received word

$$w = \begin{pmatrix} w_0 & w_m & \cdots & w_{m(N-1)} \\ w_1 & w_{m+1} & \cdots & w_{m(N-1)+1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m-1} & w_{2m-1} & \cdots & w_{mN-1} \end{pmatrix}, \quad (2)$$

then a non-zero polynomial  $Q(x, z_1, \dots, z_s) \in \mathbb{F}_q[x, z_1, \dots, z_s]$  is an *interpolation polynomial* for  $w$  if  $Q$  has a zero of multiplicity at least  $r$  for all points  $(\gamma^{mi+j}, w_{mi+j}, w_{mi+j+1}, \dots, w_{mi+j+s-1})$ , with  $0 \leq i < N$  and  $0 \leq j \leq m - s$ .

By translating the requirements on an interpolation polynomial in the above definition into linear equations, one can prove (see Beelen and Brander 2007)

**Proposition 2** Let  $w$  be as in (2). Let  $\Delta \in \mathbb{N}$  and let  $\lambda = \lfloor \frac{\Delta}{k-1} \rfloor$ , then if

$$\binom{r+s}{s+1} N(m-s+1) < (\Delta - \lambda(k-1) + 1) \cdot \binom{s+\lambda}{s} + (k-1) \cdot \binom{s+\lambda}{s+1}, \quad (3)$$

there exists an interpolation polynomial for  $w$  of weighted degree at most  $\Delta$ .

For the error-correcting radius of a folded RS code we now obtain (Beelen and Brander 2007):

**Proposition 3** Let  $w \in (\mathbb{F}_{q^m})^N$  be a code word of a folded RS code generated by the polynomial  $f(x)$ , possibly corrupted by at most  $t$  errors. Let  $Q$  be an interpolation polynomial for  $w$  of weighted degree  $\Delta$ , then if

$$t < N - \frac{\Delta}{r(m-s+1)} \quad (4)$$

it holds that

$$Q(x, f(x), f(\gamma x), \dots, f(\gamma^{s-1} x)) = 0. \quad (5)$$

Using Propositions 2 and 3 together, one can show that the decoder corrects errors up to a relative distance of at least

$$1 - R^{\frac{s}{s+1}} \left( \frac{m}{m-s+1} \right)^{\frac{s}{s+1}} \sqrt[s+1]{\prod_{i=1}^s \left( 1 + \frac{i}{r} \right)} - \frac{1}{N}.$$

By choosing the parameters of the code appropriately, this expression essentially reduces to (1). Also, letting  $s$  be large this quantity tends to  $1 - R$ , which as mentioned in the introduction, is the best possible relative decoding radius for any decoder (Guruswami and Rudra 2008). Proposition 3 shows that if  $t$  satisfies (4) and if no more than  $t$  errors occur during the transmission of  $w \in (\mathbb{F}_{q^m})^N$  then the polynomial generating  $w$ , will be among the solutions of (5) with  $f(x)$  of degree at most  $k-1$ . In the following we will refer to such solutions as  $z$ -roots of  $Q$ , and we now give a method for computing these, using a variation of the Hensel-lifting technique in Roth and Ruckenstein (2000). With such a root-finding method at hand, we have a way to compute a list known to contain the transmitted code word, and hence to *list decode* the folded RS code. Let  $w$  be some received word and let  $Q$  be an interpolation polynomial for it. The word  $w$  corresponds to a code word in the folded RS code generated by some polynomial, say  $f(x) = \sum_{l=0}^{k-1} f_l x^l$ . Now define the following shifted versions of this polynomial  $\psi_i^{(\sigma)}(x) = \sum_{l \geq i} f_l (\gamma^\sigma x)^{l-i}$ , for  $0 \leq i$  and  $0 \leq \sigma \leq s-1$ . By definition the polynomials  $\psi_i^{(\sigma)}(x)$  satisfy the following relations

$$\gamma^\sigma x \cdot \psi_{i+1}^{(\sigma)}(x) + f_i = \psi_i^{(\sigma)}(x). \quad (6)$$

Note that the constant term of  $\psi_i^{(\sigma)}(x)$  is  $f_i$ , and that  $f(\gamma^\sigma x) = \psi_0^{(\sigma)}(x)$ . Hence if  $Q(x, z_1, \dots, z_s)$  is an interpolation polynomial for a word generated by the polynomial  $f(x)$ , then

$$Q(x, \psi_0^{(0)}(x), \psi_0^{(1)}(x), \dots, \psi_0^{(s-1)}(x)) = 0, \quad (7)$$

which, when evaluated at  $x = 0$ , implies that  $Q(0, f_0, f_0, \dots, f_0) = 0$ . Thus  $f_0$  is a root of the univariate polynomial  $Q(0, z, \dots, z)$  and this fact restricts the *possible* values of  $f_0$ . Furthermore, if we also define shifted versions of the interpolation polynomial mirroring the relations satisfied by the  $\psi_i^{(\sigma)}(x)$ 's, we obtain polynomials having the different  $f_i$ 's as roots. More specifically we let  $Q_0 = Q$  and  $Q_{i+1}(x, z_1, \dots, z_s) = Q_i(x, xz_1 + f_i, \dots, \gamma^{s-1} x z_s + f_i)$ , for  $i \geq 0$ . Then we get the following result

**Proposition 4** For  $i \geq 0$  it holds that

$$Q_i(x, \psi_i^{(0)}(x), \dots, \psi_i^{(s-1)}(x)) = 0. \quad (8)$$

```

Reconstruct( $Q(x, z_1, \dots, z_s), i, f(x), \text{result}$ )
Let  $M(x, z_1, \dots, z_s) = x^{-r} Q(x, z_1, \dots, z_s)$ , with  $r$  largest possible s. t.
 $x$  divides  $M_i(x, z_1, \dots, z_s)$ .
for each distinct root  $\beta$  of  $M(0, z, \dots, z)$  do
    if  $i = k - 1$  then
         $\text{result} := \text{result} \cup \{f(x) + \beta x^i\}$ 
    else
        call Reconstruct( $Q(x, xz_1 + \beta, \dots, xz_s + \beta), i + 1, f(x) + \beta x^i, \text{result}$ )
    end if
end for

```

**Fig. 1** Procedure for computing the  $z$ -roots of  $Q(x, z_1, \dots, z_s)$

Since the constant term of  $\psi_i^{(\sigma)}(x)$  is  $f_i$  we get by evaluating the expression in (8) at  $x = 0$ , that  $Q_i(0, f_1, \dots, f_i) = 0$ , and thus  $f_i$  must be among the roots of the *univariate* polynomial  $Q_i(0, z, \dots, z)$ . Thus by recursively computing the polynomials  $Q_i(0, z, \dots, z)$  and their roots, we obtain restrictions on the possible values of  $f_i$ , and this allows us to compute a set of polynomials guaranteed to contain all the polynomials satisfying (5). There are certain points of this approach to consider. If the polynomial  $Q_i(x, z_1, \dots, z_s)$  is divisible by  $x$  we get that  $Q_i(0, z, \dots, z)$  is the zero polynomial, in which case no information about  $f_i$  can be inferred. To remedy this we introduce  $M_i(x, z_1, \dots, z_s) = x^{-r_i} Q_i(x, z_1, \dots, z_s)$ , where  $x^{r_i}$  is the highest power of  $x$  dividing  $Q_i(x, z_1, \dots, z_s)$ . Then  $M_i$  has the same property as  $Q_i$ , namely  $M_i(0, f_1, \dots, f_i) = 0$ . Thus the polynomial  $M_i$  can be used instead of  $Q_i$  to obtain restrictions on the possible values of  $f_i$ , and the power of  $x$  removed from  $Q_i(x, z_1, \dots, z_s)$  might prevent  $M_i(0, z, \dots, z)$  from being the zero polynomial, in cases where  $Q_i(0, z, \dots, z)$  is. It is however, still possible that the substitution  $z = z_1 = \dots = z_s$  and  $x = 0$  makes  $M_i(0, z, \dots, z)$  the zero polynomial, and if this happens we only have the trivial restriction that  $f_i$  must be an element in  $\mathbb{F}_q$ .

Putting all this together, we design a recursive procedure for computing a set containing the  $z$ -roots of a polynomial in  $\mathbb{F}_q[x, z_1, \dots, z_s]$ . The procedure is stated in Fig. 1 as pseudo-code. Parameter  $i$  determines the current level of the recursion, i.e. the power of  $x$  whose coefficient  $f_i$  is currently being computed. At level  $i$  the partial result  $f_i x^i + \dots + f_1 x + f_0$  is stored in  $f(x)$ , and when the recursion depth reached  $k - 1$ ,  $f(x)$  is added to the list `result` of potential  $z$ -roots of  $Q$ . Initially the procedure is called with  $i = 0$ ,  $f(x) = 0$  and `result` the empty list. At the end, the  $z$ -roots of  $Q$  lie in `result`.

## References

- P. Beelen and K. Brander, *Decoding of folded codes using Hensel-lifting*, Preprint, April 2007.
- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick's approach to recent developments*, this volume, 2009, pp. 197–218.
- V. Guruswami and A. Rudra, *Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy*, IEEE Trans. On Inf. Th. **54** (2008), no. 1, 135–150.
- V. Guruswami and M. Sudan, *Improved decoding of Reed–Solomon and algebraic geometric codes*, IEEE Trans. on Inf. Th. **45** (1999), no. 6, 1757–1767.
- V. Y. Krachkovsky, *Reed–Solomon codes for correcting phased error bursts*, IEEE Trans. on Inf. Th. **49** (2003), no. 11, 2975–2984.

- F. Parvaresh and A. Vardy, *Correcting errors beyond the Guruswami–Sudan radius in polynomial time*, Proc. of IEEE FOCS 2005, IEEE Computer Society, Alamitos, 2005, pp. 285–294.
- R. M. Roth and G. Ruckenstein, *Efficient decoding of Reed–Solomon codes beyond half the minimum distance*, IEEE Trans. on Inf. Th. **46** (2000), no. 1, 246–257.

# A Note on the Generalisation of the Guruswami–Sudan List Decoding Algorithm to Reed–Muller Codes

Daniel Augot and Michael Stepanov

**Abstract** We revisit the generalisation of the Guruswami–Sudan list decoding algorithm to Reed–Muller codes. Although the generalisation is straightforward, the analysis is more difficult than in the Reed–Solomon case. A previous analysis has been done by Pellikaan and Wu (List decoding of  $q$ -ary Reed–Muller codes, Tech. report, from the authors, 2004a; IEEE Trans. on Inf. Th. 50(4): 679–682, 2004b), relying on the theory of Gröbner bases. We give a stronger form of the well-known Schwartz–Zippel Lemma (Schwartz in J. Assoc. Comput. Mach. 27(4): 701–717, 1980; Zippel in Proc. of EUROSAM 1979, LNCS, vol. 72, Springer, Berlin, pp. 216–226, 1979), taking multiplicities into account. Using this Lemma, we get an improved decoding radius.

## 1 Definitions and Notation

We consider  $S = \{x_1, \dots, x_n\}$  a set of  $n$  distinct elements of  $\mathbb{F}_q$ . Let  $N, r$  be integers greater than or equal to one, we consider the evaluation map, defined on  $\mathbb{F}_q[X_1, \dots, X_n]$ :

$$\text{ev}^N : f(X_1, \dots, X_N) \mapsto (f(x_{i_1}, \dots, x_{i_N}))_{(x_{i_1}, \dots, x_{i_N}) \in S^N}.$$

We fix the following space of polynomials:  $L = \{f(X_1, \dots, X_N), \deg f \leq r\}$ . Then the code  $\text{ev}^N(L)$  is the Reed–Muller code of order  $r$  with  $N$  variables.

We say that a polynomial  $Q(X_1, \dots, X_N)$  has multiplicity  $s$  at the point  $(0, \dots, 0)$  if it does not contain any monomial of degree strictly less than  $s$ . We say that a polynomial  $Q(X_1, \dots, X_N)$  has multiplicity  $s$  at  $(x_{i_1}, \dots, x_{i_N})$  if the polynomial  $Q(X_1 + x_{i_1}, \dots, X_N + x_{i_N})$  has multiplicity  $s$  at  $(0, \dots, 0)$ . The weighted degree  $\text{wdeg}_{a_1, \dots, a_N}$  of a monomial  $X_1^{i_1} \cdots X_N^{i_N}$  is  $a_1 i_1 + \cdots + a_N i_N$ . The weighted degree of a polynomial is the maximum weighted degree of its monomials.

For a discussion on Sudan’s algorithm and its variants, see Guerrini and Rimoldi (2009).

---

D. Augot  
INRIA Paris-Rocquencourt, Project-Team Secret, Paris, France  
e-mail: [Daniel.Augot@inria.fr](mailto:Daniel.Augot@inria.fr)

M. Stepanov  
St. Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia  
e-mail: [mstepanov@gmail.com](mailto:mstepanov@gmail.com)

## 2 The Algorithm

The algorithm is as follows. Let  $\tau$  be the number of errors that will be corrected. The received word is a  $N$ -dimensional array  $y = (y_{i_1, \dots, i_N})_{(i_1, \dots, i_N) \in \{1, \dots, n\}^N}$ .

input  $(x_1, \dots, x_n) \in \mathbb{F}_q^n$ ,  $r, \tau \in \mathbb{N}$ ,  $y = (y_{i_1, \dots, i_N})$  the received word; auxiliary parameters: a degree  $d$  and  $s$  an order of multiplicity.

interpolation find a polynomial  $Q = Q(X_1, \dots, X_N, Z)$  such that

1.  $Q(X_1, \dots, X_N, Z) \neq 0$ ,
2.  $\text{wdeg}_{1, \dots, 1, r} Q(X_1, \dots, X_N, Z) \leq d$ ,
3.  $\text{mult}(Q; (x_{i_1}, \dots, x_{i_N}, y_{i_1, \dots, i_N})) = s$ ,  $(i_1, \dots, i_N) \in \{1, \dots, n\}^N$ .

factorisation Compute  $L = \{f = f(X_1, \dots, X_N) \mid Q(X_1, \dots, X_N, f) = 0\}$ .

verification return all  $f \in L$  such that  $\deg f \leq r$ , and  $d(f, y) < \tau$ .

The analysis of this family of interpolation based decoding algorithms is in two steps. First we must find conditions such that the polynomial  $Q(X_1, \dots, X_N, Z)$  always exists, and secondly analyze the conditions under which  $Q(X_1, \dots, X_N, f) = 0$ . For the existence of the polynomial  $Q$ , we will require that the number of unknowns is greater than the number of equations. Each condition  $\text{mult}(Q; (x_{i_1}, \dots, x_{i_N}, y_{i_1, \dots, i_N})) = s$  implies  $\binom{s+N}{N+1}$  linear equations on  $Q$ . On the other hand, the number of unknowns in the  $Q$  polynomial is roughly  $\frac{d^{N+1}}{(N+1)!r}$ , and a condition for the existence of  $Q$  is

$$\frac{d^{N+1}}{(N+1)!r} > \binom{s+N}{N+1} n^N,$$

Let  $Q_f$  be the polynomial  $Q(X_1, \dots, X_m, f)$ . We note that, since the condition  $\text{wdeg}_{1, \dots, 1, r} Q(X_1, \dots, X_N, Z) \leq d$  holds, we have that  $\deg Q_f \leq d$ . We need a Theorem to conclude that the polynomial  $\deg Q(X_1, \dots, X_N, f)$  has “more zeros than allowed”. In the univariate case, it is enough to state the a polynomial can not have more zeros than its degree. In the multivariate case, things are harder. Pellikaan and Wu have overcome this difficulty by relying on the theory of Gröbner bases and footprints (Gröbner *escalier*, see Mora 2009). They eventually get the following relative decoding radius:

$$\frac{\tau}{n^N} \leq \left(1 - \sqrt[N+1]{\frac{r}{n}}\right)^N. \quad (1)$$

## 3 The Analysis

**Lemma 1** Let  $Q(X_1, \dots, X_N)$  be of total degree less than  $d$ . Let  $x_1, \dots, x_n$  be  $n$  distinct points in  $\mathbb{F}_q$ . The sum of multiplicities of  $Q(X_1, \dots, X_N)$  over the  $n^N$  points  $(x_{i_1}, \dots, x_{i_N}) \in \mathbb{F}_q^N = (\mathbb{F}_q)^N$  is less than or equal to  $dn^{N-1}$ .

*Proof* By induction. The statement is true for  $N = 1$ . Let us consider the set  $I$  of points  $x_{i_1}, \dots, x_{i_l}$ , such that  $Q(X_1, \dots, X_{N-1}, x_{i_j})$  is identically zero,  $j = 1, \dots, l$ . Also let  $I'$  be  $\{1, \dots, n\} \setminus I$ . Then, for  $x_{i_j} \notin I$ , let  $\tilde{Q}_{i_j}$  be the polynomial  $Q(X_1, \dots, X_{N-1}, x_{i_j})$ . Then the number of zeros, counted with multiplicities of  $\tilde{Q}_{i_j}$ , over the points whose last coordinates is  $x_{i_j}$  is by induction bounded by  $dn^{N-2}$ . Now, for  $x_{i_j} \in I$ , we can write

$$Q(X_1, \dots, X_N) = (X_n - x_{i_j})^{t_{i_j}} \tilde{Q}_{i_j}(X_1, \dots, X_N)$$

for some  $t_{i_j} > 0$ , and where  $\tilde{Q}_{i_j}(X_1, \dots, X_N)$  is such that  $\tilde{Q}_{i_j}(X_1, \dots, X_{N-1}, x_{i_j})$  is not identically zero. The degree of  $\tilde{Q}_{i_j}(X_1, \dots, X_N)$  is  $d - t_{i_j}$ . Now the number of multiplicities of  $\tilde{Q}_{i_j}(X_1, \dots, X_N)$  over the points whose last coordinate is  $x_{i_j}$  is bounded by  $(d - t_{i_j})n^{N-2}$ , using the induction hypothesis. Let  $\Sigma$  be the sum of multiplicities. Let  $S_{i_j}$  be the set of points whose last coordinates is  $x_{i_j}$ . Then

$$\begin{aligned} \Sigma &= \sum_{x_{i_j} \in I'} \sum_{p \in S_{i_j}} \text{mult}(Q, p) + \sum_{x_{i_j} \in I} \sum_{p \in S_{i_j}} \text{mult}(Q, p) \\ &\leq |I'|dn^{N-2} + \sum_{x_{i_j} \in I} \sum_{p \in S_{i_j}} (t_{i_j} + \text{mult}(\tilde{Q}_{i_j}, p)) \\ &\leq |I'|dn^{N-2} + \sum_{x_{i_j} \in I} (t_{i_j}n^{N-2} + (d - t_{i_j})n^{N-2}) \\ &\leq |I'|dn^{N-2} + |I|dn^{N-2} = dn^{N-1}. \end{aligned}$$

□

To ensure that the polynomial  $Q_f$  is identically zero, we must have that  $Q_f$  has more than  $dn^{N-1}$  zeros counted with multiplicities. If  $s(n^N - \tau) > dn^{N-1}$ ,  $Q_f$  is identically zero. Working out the formulas leads to:

$$\tau \leq n^N - \sqrt[N+1]{rn^N(1 + \frac{1}{s}) \dots (1 + \frac{N}{s})} \leq n^N \left(1 - \sqrt[N+1]{\frac{r}{n}}\right). \quad (2)$$

This compares favourably to the Pellikaan–Wu radius. In conclusion, we note that, over the binary field, the Reed–Muller codes can be considered as subfield subcodes of classical Reed–Solomon codes (Kasami et al. 1968), and one can get a better decoding radius, using the univariate Guruswami–Sudan algorithm.

**Acknowledgement** The authors acknowledge support from the Austrian Academy of Sciences during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- E. Guerrini and A. Rimoldi, *FGLM-like decoding: from Fitzpatrick’s approach to recent developments*, this volume, 2009, pp. 197–218.

- T. Kasami, S. Lin, and W. W. Peterson, *New generalizations of the Reed–Muller codes. I. Primitive codes*, IEEE Trans. on Inf. Th. **14** (1968), 189–199.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- R. Pellikaan and X.-W. Wu, *List decoding of  $q$ -ary Reed–Muller codes*, Tech. report, from the authors, 2004a.
- R. Pellikaan and X.-W. Wu, *List decoding of  $q$ -ary Reed–Muller codes*, IEEE Trans. on Inf. Th. **50** (2004b), no. 4, 679–682.
- J. T. Schwartz, *Fast probabilistic algorithms for verification of polynomial identities*, J. Assoc. Comput. Mach. **27** (1980), no. 4, 701–717.
- R. Zippel, *Probabilistic algorithms for sparse polynomials*, Proc. of EUROSAM 1979, LNCS, vol. **72**, Springer, Berlin, 1979, pp. 216–226.

# Viewing Multipoint Codes as Subcodes of One-Point Codes

Gretchen L. Matthews

**Abstract** We consider ways in which multipoint algebraic geometry codes may be viewed as subcodes of the more traditionally studied one-point codes. Examples are provided to illustrate the impact of choices made on this embedding.

## 1 Introduction

An  $m$ -point algebraic geometry (AG) code is constructed by evaluating functions which are allowed to have poles at  $m$  specified points on a curve  $X$  over a finite field. While Goppa's construction (Goppa 1981; Leonard 2009) certainly encompasses multipoint codes, most subsequent work has focused on the one-point case. While multipoint codes can have better parameters than comparable one-point codes on the same curve (Matthews 2001), one-point codes are certainly better understood. Recently, there has been more work on multipoint codes (Beelen 2007; Carvalho and Torres 2005; Homma and Kim 2001, 2005, 2006a, 2006b). Here, we see that multipoint codes may be viewed as subcodes of the more traditionally studied one-point codes and illustrate the impact of choices made on this embedding.

*Notation* Let  $X$  be a smooth, projective, absolutely irreducible curve of genus  $g$  over a finite field  $\mathbb{F}$ . The divisor of a rational function  $f$  on  $X$  will be denoted by  $(f)$ . Given a divisor  $A$  on  $X$  defined over  $\mathbb{F}$ , let  $\mathcal{L}(A)$  be the set of rational functions  $f$  on  $X$  defined over  $\mathbb{F}$  with divisor  $(f) \geq -A$  together with the zero function. The dimension of  $\mathcal{L}(A)$  as an  $\mathbb{F}$ -vector space is denoted by  $\ell(A)$ . Clearly, if  $A \leq B$  for divisors  $A$  and  $B$  on  $X$ , then  $\mathcal{L}(A) \subseteq \mathcal{L}(B)$ .

Given distinct  $\mathbb{F}$ -rational points  $P_1, \dots, P_n, Q_1, \dots, Q_m$  on  $X$ , define divisors  $D := P_1 + \dots + P_n$  and  $G := a_1 Q_1 + \dots + a_m Q_m$  where  $a_i \geq 0$ . Then

$$C_{\mathcal{L}}(D, G) = \{(f(P_1), f(P_2), \dots, f(P_n)) : f \in \mathcal{L}(G)\}$$

is sometimes called an  $m$ -point code. We do not require the divisor  $D$  to be supported by all  $\mathbb{F}$ -rational points that are not in the support of  $G$ . Excellent references for algebraic geometry codes include (Høholdt et al. 1998; Stichtenoth 1993; Tsfasman and Vlăduț 1991).

---

G.L. Matthews

Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975,  
USA

e-mail: [gmatthe@clemson.edu](mailto:gmatthe@clemson.edu)

## 2 Embedding a Multipoint Code in a One-Point Code

Consider the multipoint code  $C_{\mathcal{L}}(D, G)$  from above. Since  $\mathbb{F}$  is finite, the group of divisor classes of degree zero is finite. Hence, there is a rational function  $f$  with divisor  $(f) = b_2 Q_2 + \cdots + b_m Q_m - b_1 Q_1$ , where  $b_i \geq a_i$  for all  $2 \leq i \leq m$  and  $b_1 = \sum_{i=2}^m b_i$ . Multiplication by  $f$  induces an isomorphism of  $\mathcal{L}$  spaces

$$\begin{aligned} \mathcal{L}\left(\sum_{i=1}^m a_i Q_i\right) &\rightarrow \mathcal{L}\left((a_1 + b_1) Q_1 - \left(\sum_{i=2}^m (b_i - a_i) Q_i\right)\right) \\ h &\mapsto fh \end{aligned}$$

which gives rise to an isometry of codes

$$C_{\mathcal{L}}\left(D, \sum_{i=1}^m a_i Q_i\right) \cong C_{\mathcal{L}}\left(D, (a_1 + b_1) Q_1 - \left(\sum_{i=2}^m (b_i - a_i) Q_i\right)\right).$$

As a consequence, the  $m$ -point code  $C_{\mathcal{L}}(D, G)$  is isometric to a subcode of the one-point code  $C_{\mathcal{L}}(D, (a_1 + b_1)P_1)$ .

## 3 Examples

While the existence of the function  $f$  above is guaranteed by the fact that the class number of  $X$  is finite, this may not be that helpful in finding the most appropriate function. To illustrate the effect of the choice of  $f$ , we give the two examples on the Hermitian curve  $X$  defined by  $y^q + y = x^{q+1}$  over  $\mathbb{F}_{q^2}$ .

*Example 1* Set  $G := 2(q + 1)P_\infty + \sum_{\beta^q + \beta = 0} P_{0\beta}$ , and let  $D$  be the sum of all other  $\mathbb{F}_{q^2}$ -rational points on  $X$ . Since the class number of  $X$  is  $(q + 1)(q^2 - q)$ , there exists a function  $f$  such that

$$(f) = (q + 1)(q^2 - q) \sum_{\beta^q + \beta = 0} P_{0\beta} - q(q + 1)(q^2 - q)P_\infty.$$

Multiplication by  $f$  gives

$$\begin{aligned} f\mathcal{L}(G) &= \mathcal{L}\left((q^4 - q^2 - 2q - 2)P_\infty - (q^3 - q - 1) \sum_{\beta^q + \beta = 0} P_{0\beta}\right) \\ &\subseteq \mathcal{L}((q^4 - q^2 - 2q - 2)P_\infty). \end{aligned}$$

Therefore, the  $(q + 1)$ -point code  $C_{\mathcal{L}}(D, G)$  is isometric to a subcode of the one-point code  $C_{\mathcal{L}}(D, (q^4 - q^2 - 2q - 2)P_\infty)$ . The dimension of superspace is  $\ell((q^4 - q^2 - 2q - 2)P_\infty) = q^4 - \frac{3q^2}{2} - \frac{3q}{2} - 1$  while the dimension of the original vector

space is  $\ell(G) = 9$ . Therefore, while  $C_{\mathcal{L}}(D, G) \subseteq C_{\mathcal{L}}(D, (q^4 - q^2 - 2q - 2)P_{\infty})$ , it is not easy to glean information about  $C_{\mathcal{L}}(D, G)$  by studying the larger code.

It may be possible to find a more appropriate function  $f$ . Given any  $\mathbb{F}_{q^2}$ -rational point  $P_{ab}$  on  $X$ , the rational function  $\tau_{ab} := y - b - a^q(x - a)$  has divisor  $(\tau_{ab}) = (q + 1)P_{ab} - (q + 1)P_{\infty}$  (Maharaj et al. 2005). Hence, a natural choice for the function  $f$  would be  $f = \prod_{\beta^q + \beta = 0} \tau_{0\beta}$ . This gives

$$f\mathcal{L}(G) = \mathcal{L}\left((q^2 + 3q + 2)P_{\infty} - q \sum_{\beta^q + \beta = 0} P_{0\beta}\right) \subseteq \mathcal{L}((q^2 + 3q + 2)P_{\infty}).$$

Here, the difference in dimensions of the Riemann–Roch spaces is much smaller as  $\ell((q^2 + 3q + 2)P_{\infty}) = \frac{q^2}{2} + \frac{7q}{2} + 3$ .

Taking  $f = x$  gives  $x\mathcal{L}(G) = \mathcal{L}((3q + 2)P_{\infty})$ . Now, we see that  $C_{\mathcal{L}}(D, G) \cong C_{\mathcal{L}}(D, (3q + 2)P_{\infty})$  and so the  $(q + 1)$ -point code  $C_{\mathcal{L}}(D, G)$  is isometric to the one-point code  $C_{\mathcal{L}}(D, (3q + 2)P_{\infty})$ . Hence, the parameters of  $C_{\mathcal{L}}(D, G)$  can be determined (Yang and Kumar 1992) and there is no need to consider the  $(q + 1)$ -point code. Not all multipoint codes are isometric to one-point codes (Matthews 2001).

*Example 2* Let  $c$  be a positive integer, and fix an  $\mathbb{F}_{q^2}$ -rational point  $P_{ab}$  on  $X$  with  $a \neq 0$ . Set  $G = cP_{\infty} + (q + 2)P_{ab} + \sum_{\beta^q + \beta = 0, \beta \neq 0} P_{0\beta} + \sum_{\beta^q + \beta = a^{q+1}, \beta \neq b} P_{a\beta}$ , and take  $D$  to be the sum of all other  $\mathbb{F}_{q^2}$ -rational points.

Taking  $f = \tau_{ab}^2 \prod_{\beta^q + \beta = 0, \beta \neq 0} (y - \beta) \prod_{\beta^q + \beta = a^{q+1}, \beta \neq b} (y - \beta)$  yields

$$f\mathcal{L}(G) = \mathcal{L}((2q^2 + 2q + c)P_{\infty} - qP_{ab} - A) \subseteq \mathcal{L}((2q^2 + 2q + c)P_{\infty})$$

where  $A := q \sum_{\beta^q + \beta = 0, \beta \neq 0} P_{0\beta} + \sum_{\beta^q + \beta = a^{q+1}, \beta \neq b, \alpha \neq a} P_{\alpha\beta}$ . This is a bit troubling as the subcode we are interested in is defined by the Riemann–Roch space of a divisor supported by many points. In particular, bases for this Riemann–Roch space are not known for arbitrary  $q$ . Moreover, the supports of  $A$  and  $D$  have points in common. While this could be corrected by redefining  $D$ , it changes the code length. In effect, this would require that one consider in advance the supports of the principal divisors in question to even know the code length. Thus, we instead multiply by  $x(x - a)\tau_{ab}$  to obtain

$$x(x - a)\tau_{ab}\mathcal{L}(G) = \mathcal{L}((3q + c + 1)P_{\infty} - P_{00}).$$

Bases for the Riemann–Roch space and code may be determined as in Maharaj et al. (2005).

As pointed out by a referee, the group structure of the Jacobian of the Hermitian curve (Rück and Stichtenoth 1994) may be useful here.

## 4 Conclusion

The idea of studying subcodes of one-point codes is not new (see Feng and Rao 1993, 1994, 1995; Høholdt et al. 1998). The thrust of our approach is that improved bounds on the parameters are known for certain multipoint codes, enabling one to identify subcodes with good parameters. Then, viewing a multipoint code  $C$  as a subcode of a one-point code  $C'$  may provide additional insight into  $C$ . Moreover, it may yield a simplified decoding algorithm for  $C$ , a topic to be addressed in another paper.

**Acknowledgements** This project was supported by NSF DMS-0201286 and NSA H-98230-06-1-0008.

The author acknowledges support from the Austrian Academy of Sciences during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- P. Beelen, *The order bound for general algebraic geometric codes*, Finite Fields Appl. **13** (2007), no. 3, 655–680.
- C. Carvalho and F. Torres, *On Goppa codes and Weierstrass gaps at several points*, Des. Codes Cryptogr. **35** (2005), no. 2, 211–225.
- G. L. Feng and T. R. N. Rao, *Decoding algebraic-geometric codes up to the designed minimum distance*, IEEE Trans. on Inf. Th. **39** (1993), no. 1, 37–45.
- G. L. Feng and T. R. N. Rao, *A simple approach for construction of algebraic-geometric codes from affine plane curves*, IEEE Trans. on Inf. Th. **40** (1994), 1003–1012.
- G. L. Feng and T. R. N. Rao, *Improved geometric Goppa codes, Part I: Basic theory*, IEEE Trans. on Inf. Th. **41** (1995), 1678–1693.
- V. D. Goppa, *Codes on algebraic curves*, Soviet Math. Dokl. **24** (1981), no. 1, 170–172.
- M. Homma and S. J. Kim, *Goppa codes with Weierstrass pairs*, J. Pure Appl. Algebra **162** (2001), nos. 2–3, 273–290.
- M. Homma and S. J. Kim, *Toward the determination of the minimum distance of two-point codes on a Hermitian curve*, Des. Codes Cryptogr. **37** (2005), no. 1, 111–132.
- M. Homma and S. J. Kim, *The two-point codes on a Hermitian curve with the designed minimum distance*, Des. Codes Cryptogr. **38** (2006a), no. 1, 55–81.
- M. Homma and S. J. Kim, *The two-point codes with the designed distance on a Hermitian curve in even characteristic*, Des. Codes Crypto. **39** (2006b), no. 3, 375–386.
- T. Høholdt, J. van Lint, and R. Pellikaan, *Algebraic geometry of codes*, Handbook of Coding Theory (V. S. Pless and W.C. Huffman, eds.), Elsevier, Amsterdam, 1998, pp. 871–961.
- D. A. Leonard, *A tutorial on AG code construction from a Gröbner basis perspective*, this volume, 2009, pp. 93–106.
- G. L. Matthews, *Weierstrass pairs and minimum distance of Goppa codes*, Des. Codes Cryptogr. **22** (2001), no. 2, 107–121.
- H. Maharaj, G. L. Matthews, and G. Pirsic, *Riemann-Roch spaces of the Hermitian function field with applications to algebraic geometry codes and low-discrepancy sequences*, J. Pure Appl. Algebra **195** (2005), no. 3, 261–280.
- H. G. Rück and H. Stichtenoth, *A characterization of Hermitian function fields over finite fields*, J. Reine Angew. Math. **457** (1994), 185–188.
- H. Stichtenoth, *Algebraic function fields and codes*, Universitext, Springer, Berlin, 1993.
- M. A. Tsfasman and S. G. Vlăduț, *Algebraic-geometric codes*, Math. and its Appl. (Soviet Series), vol. **58**, Kluwer Academic, Dordrecht, 1991.
- K. Yang and P. V. Kumar, *On the true minimum distance of Hermitian codes*, Proc. of AGCT 1991, LNM, vol. **1518**, Springer, Berlin, 1992, pp. 99–107.

# A Short Introduction to Cyclic Convolutional Codes

Heide Gluesing-Luerssen, Barbara Langfeld and  
Wiland Schmale

**Abstract** We introduce the notion of cyclic convolutional codes and briefly survey some recent results that were derived with the aid of Gröbner-type theory.

## 1 Introduction and Preliminaries

Throughout this text,  $\mathbb{F}$  denotes a finite field,  $n$  and  $k$  are natural numbers and  $k \leq n$ . As a standard assumption in the theory of cyclic block codes we will always assume that  $n$  and the characteristic of  $\mathbb{F}$  are coprime.

An  $(n, k)$ -block code is a  $k$ -dimensional subspace of the  $\mathbb{F}$ -vector space  $\mathbb{F}^n$ . Equivalently, an  $(n, k)$ -block code can be written as  $\{uG : u \in \mathbb{F}^k\}$  for some matrix  $G \in \mathbb{F}^{k \times n}$  of rank  $k$ . For some polynomial matrix  $G \in \mathbb{F}[z]^{k \times n}$  with rank  $k$  over  $\mathbb{F}(z)$  the set

$$\mathcal{C} := \{uG : u \in \mathbb{F}^k[z]\} \subseteq \mathbb{F}^n[z]$$

is called an  $(n, k)$ -submodule of the  $\mathbb{F}[z]$ -module  $\mathbb{F}^n[z]$  and  $G$  is called generator matrix of  $\mathcal{C}$ . If, in addition,  $G$  is right invertible over  $\mathbb{F}[z]$ , then  $\mathcal{C}$  is called a convolutional code or CC, for short.

Taking a closer look at the encoding process  $u \mapsto uG$ , the  $z^i$ -term of the code word  $uG$  uses certain  $z^j$ -terms of the message word  $u$ , where  $0 \leq j \leq i$ . In this sense, a CC has some kind of ‘memory’ which can be exploited to improve code properties. To be able to judge this potential we recall that  $\deg(G)$ , the degree of a

---

H. Gluesing-Luerssen

Department of Mathematics, University of Kentucky, 715 Patterson Office Tower,  
Lexington, KY 40506–0027, USA

e-mail: [heidegl@ms.uky.edu](mailto:heidegl@ms.uky.edu)

B. Langfeld

Zentrum Mathematik, TU München, Boltzmannstraße 3, 85747 Garching bei München,  
Germany  
e-mail: [langfeld@ma.tum.de](mailto:langfeld@ma.tum.de)

W. Schmale

Institut für Mathematik, Carl von Ossietzky Universität, 26111 Oldenburg, Germany  
e-mail: [wiland.schmale@uni-oldenburg.de](mailto:wiland.schmale@uni-oldenburg.de)

generator matrix  $G$ , is defined as the sum of the  $z$ -degrees of the rows of  $G$  (which are viewed as polynomials in  $\mathbb{F}^n[z]$ ). For example, consider

$$\mathcal{S} := \left\{ u \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} : u \in \mathbb{F}^k[z] \right\} = \left\{ u \begin{bmatrix} 1 & z^2 & z^2 \\ z & z^3 + 1 & z^3 + 1 \end{bmatrix} : u \in \mathbb{F}^k[z] \right\}.$$

The left generator matrix has degree 0, the right one has degree 5. We see that  $\mathcal{S}$  has a generator matrix with constant entries and thus it does not have ‘memory’. It is not better than a block code! We put this down more formally. For  $\mathcal{C}$  being an  $(n, k)$ -submodule, the number

$$\delta := \delta(\mathcal{C}) := \min\{\deg G : G \text{ is a generator matrix of } \mathcal{C}\}$$

is called *complexity* of  $\mathcal{C}$ . The set  $\mathcal{C}$  is also called an  $(n, k, \delta)$ -submodule or, if  $\mathcal{C}$  is a CC, an  $(n, k, \delta)$ -CC.

Submodules of complexity 0 are in a sense block codes because they have a generator matrix with constant entries. A nonzero complexity is therefore desirable or even necessary if we want to make use of the ‘memory’ of an  $(n, k)$ -submodule. For more information on CC’s consult, e.g., McEliece (1998).

## 2 How to Define Cyclic Convolutional Codes?

The following diagram briefly recalls the well known definition of cyclic block codes and the characterization of cyclicity in the ‘vector world’ and in the ‘polynomial world’. For more information and details see, e.g., Huffman and Pless (2003).

$$\begin{array}{ccc}
 \mathbb{F}^n & \xleftrightarrow[\mathfrak{p}]{\mathfrak{v}} & \mathbb{F}[x]/\langle x^n - 1 \rangle =: A \\
 \text{‘vectorize’} & & \text{‘polynomialize’} \\
 v = (v_0, \dots, v_{n-1}) & \leftrightarrow & \mathfrak{p}(v) = \sum_{i=0}^{n-1} v_i x^i \\
 \text{cyclic shift:} & \Leftrightarrow & \text{multiplication with } x: \\
 v \mapsto (v_{n-1}, v_0, \dots, v_{n-2}) & & \mathfrak{p}(v) \mapsto x \cdot \mathfrak{p}(v) \\
 \mathcal{C} \text{ is cyclic i.e., invariant} & \Leftrightarrow & \mathfrak{p}(\mathcal{C}) \text{ is an ideal in } A \\
 \text{under the cyclic shift} & &
 \end{array}$$

Since each ideal in  $A$  is principal, there exists some  $g \in \mathfrak{p}(\mathcal{C})$  such that  $\mathfrak{p}(\mathcal{C}) = \langle g \rangle$ . If  $g$  is chosen as a divisor of  $x^n - 1$ , which is always possible, then  $g$  contains all information about  $\mathcal{C}$  (dimension, generator matrix etc.) and is called *generator polynomial*. Again, see Huffman and Pless (2003) for details.

The first, naïve idea to generalize cyclicity to CC's is to use exactly the same definition as in the block code case, i.e., invariance under the cyclic shift. Here, of course, “cyclic shift” of a polynomial vector means shifting all coefficient vectors *simultaneously once*. In the polynomial world this reads as

$$g \in \mathfrak{p}(\mathcal{C}) \quad \Rightarrow \quad x \cdot g \in \mathfrak{p}(\mathcal{C}),$$

where the ‘polynomialize’-function  $\mathfrak{p}$  was extended canonically to  $A[z]$  via

$$\mathfrak{p}\left(\sum_{i=0}^t z^i u_i\right) = \sum_{i=0}^t z^i \mathfrak{p}(u_i) \quad \text{for } u_i \in A, \quad 1 \leq i \leq t.$$

But this definition is *not* fruitful, because it provides no new structure. Indeed, a CC invariant under the cyclic shift has complexity 0; see Piret (1976), Roos (1979).

Piret suggested to generalize the notion of “cyclic shift” of a polynomial vector  $v$ . His idea was to shift the coefficient vectors of  $v$  several times and differently often, but in a way that preserves a ‘nice’ structure; see Piret (1976). Roos generalized this approach as follows: He proposed to “shift” a polynomial vector  $v$  not only by shifting the coefficient vectors of  $v$  several times and differently often, but also to allow linear combinations of the shifted coefficient vectors. This generalized shift should again preserve a ‘nice’ structure; see Roos (1979). We adopt Roos’s definition. Let  $\mathcal{C}$  be an  $(n, k)$ -submodule and  $\sigma \in \text{Aut}_{\mathbb{F}}(A)$ , where  $\text{Aut}_{\mathbb{F}}(A)$  denotes the group of all  $\mathbb{F}$ -algebra automorphisms on  $A$ . Define  $\sigma$ -cyclicity of  $\mathcal{C}$  via the condition

$$g = \sum_{v \geq 0} z^v g_v \in \mathfrak{p}(\mathcal{C}) \quad \Rightarrow \quad x *_{\sigma} g := \sum_{v \geq 0} z^v \sigma^v(x) g_v \in \mathfrak{p}(\mathcal{C}).$$

The operation “ $*_{\sigma}$ ” can be canonically extended to a multiplication on  $A[z]$  and the set  $A[z; \sigma] := (A[z], +, *_{\sigma})$  is an  $\mathbb{F}$ -Algebra, called the *Piret-Algebra*, which is in general non-commutative. Observe that we put the coefficients on the right-hand side of  $z$ . This agrees with the definition of the map  $\mathfrak{p}$  defined above, where it is needed in order to make  $\mathfrak{p}$  an isomorphism of left  $\mathbb{F}[z]$ -modules. With this setting  $\sigma$ -cyclicity can be characterized as follows:

$$\mathcal{C} \text{ is } \sigma\text{-cyclic} \quad \Longleftrightarrow \quad \mathfrak{p}(\mathcal{C}) \text{ is a left ideal in } A[z; \sigma],$$

see Roos (1979), Gluesing-Luerssen and Schmale (2004). Therefore,  $\sigma$ -cyclicity appears as a natural generalization of block code cyclicity. We give a small example.

*Example* Let  $n = 3$ ,  $\mathbb{F} = \mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$ , and let  $\sigma \in \text{Aut}_{\mathbb{F}}(A)$  be defined via  $\sigma(x) = \alpha^2 x$ . Consider

$$G := [1+z+z^2, \alpha+z+\alpha^2z^2, \alpha^2+z+\alpha z^2] = [1, \alpha, \alpha^2] + z[1, 1, 1] + z^2[1, \alpha^2, \alpha].$$

We will use  $G$  both as matrix and as code word. The  $(3, 1)$ -submodule  $\mathcal{C} := \{uG : u \in \mathbb{F}[z]\}$  has complexity  $\delta = 2$ . Moreover,  $G$  is right invertible over  $\mathbb{F}[z]$ . The

submodule  $\mathcal{C}$  is even a  $\sigma$ -cyclic CC. To prove this, one can show

$$\begin{aligned} x *_{\sigma} \mathfrak{p}(G) &\in \mathfrak{p}(\mathcal{C}) \quad \text{and} \quad x^2 *_{\sigma} \mathfrak{p}(G) \in \mathfrak{p}(\mathcal{C}), \quad \text{since} \\ g := \mathfrak{p}(G) &= 1 + \alpha x + \alpha^2 x^2 + z(1 + x + x^2) + z^2(1 + \alpha^2 x + \alpha x^2), \\ x *_{\sigma} g &= \alpha^2 + x + \alpha x^2 + z\alpha^2(1 + x + x^2) + z^2(\alpha^2 + \alpha x + x^2) = \alpha^2 g \in \mathfrak{p}(\mathcal{C}), \\ x^2 *_{\sigma} g &= \alpha g \in \mathfrak{p}(\mathcal{C}). \end{aligned}$$

Therefore  $\mathcal{C}$  is  $\sigma$ -cyclic. One can show that  $\mathcal{C}$  is an MDS convolutional code with free distance 9; for these notions see McEliece (1998) and Rosenthal and Smarandache (1999).

*Remark* In Solomon and van Tilborg (1979) a link between quasicyclic block codes and convolutional codes was established with the main purpose of applying convolutional decoding techniques to certain block codes. It turns out that there is no apparent reason why a convolutional code associated with a given quasicyclic block code should be  $\sigma$ -cyclic for some automorphism  $\sigma$ . (In general, quasicyclicity for convolutional codes is not an instance of  $\sigma$ -cyclicity. The only exception is classical cyclicity, which does not lead to any convolutional code of positive complexity.) However, in Solomon and van Tilborg (1979) some quasicyclic block codes might give rise to a  $\sigma$ -cyclic convolutional code. But this is open to further investigation.

### 3 Analyzing Cyclic CC's with Gröbner-type Theory

We can represent  $A = \mathbb{F}[x]/\langle x^n - 1 \rangle$  as a product of fields in the following way. Let  $x^n - 1 = \pi_1 \cdots \pi_r$  be the decomposition of  $x^n - 1$  into pairwise different normalized prime factors. This decomposition is unique up to permutation of the  $\pi_i$ . Due to the Chinese Remainder Theorem we get the following isomorphism of rings:

$$\begin{aligned} \rho: \quad A &\rightarrow \mathbb{F}[x]/\langle \pi_1 \rangle \times \cdots \times \mathbb{F}[x]/\langle \pi_r \rangle \\ a &\mapsto [a \bmod \pi_1, \dots, a \bmod \pi_r] \end{aligned} \tag{*}$$

The element  $\varepsilon^{(\ell)} := \rho^{-1}([\dots, 0, 1, 0, \dots])$ , where the 1 is at the  $\ell$ -th position, is called the  $\ell$ th primitive idempotent element of  $A$ . Note that  $\sigma \in \text{Aut}_{\mathbb{F}}(A)$  permutes  $\varepsilon^{(1)}, \dots, \varepsilon^{(r)}$ . For a polynomial  $h \in A[z; \sigma]$  we call  $h^{(\ell)} := \varepsilon^{(\ell)} *_{\sigma} h$  the  $\ell$ -th component of  $h$ .

The order of the fields as chosen in (\*) and thus the order of the components induces a ‘term order’ on the elements of  $A[z; \sigma]$ . Since  $\mathcal{C}$  is a  $\sigma$ -cyclic CC if and only if  $\mathfrak{p}(\mathcal{C})$  is a left ideal in  $A[z; \sigma]$ , we can ask how to find a ‘nice’ generator set of a left ideal in  $A[z; \sigma]$ . It is standard to show that  $A[z; \sigma]$  is left Noetherian i.e., each left ideal is finitely generated. The ‘term order’ together with a Buchberger-type algorithm can be used to generate a ‘reduced’ set of generators with ‘nice’ properties. For details see Gluesing-Luerssen and Schmale (2004). The following list gives several results obtained by further developing this approach. See Gluesing-Luerssen

and Schmale (2004, 2006), Gluesing-Luerssen and Langfeld (2006a, 2006b) for more results and Lally (2009), Lally and Fitzpatrick (2001), Giorgetti et al. (2005) for two examples, where Gröbner-type techniques are used to analyze quasicyclic codes.

- If  $\mathcal{C}$  is a  $\sigma$ -cyclic CC, then  $\mathfrak{p}(\mathcal{C})$  is a left *principal* ideal. The converse is only true under additional assumptions.
- If  $\mathcal{C}$  is a  $\sigma$ -cyclic CC, then there exists a ‘reduced’ generator polynomial  $g \in A[z; \sigma]$  for  $\mathfrak{p}(\mathcal{C})$  i.e., the left ideal generated by  $g$  is  $\mathfrak{p}(\mathcal{C})$ . This  $g$  is generated via a Buchberger-type reduction process and it is unique up to left-multiplication with units in  $A[z; \sigma]$  (Gluesing-Luerssen and Schmale 2004). One can easily retrieve all algebraic parameters of  $\mathcal{C}$  from  $g$ .
- The dual  $\mathcal{C}^\perp$  of a  $\sigma$ -cyclic CC  $\mathcal{C}$  is  $\hat{\sigma}$ -cyclic for a suitable  $\hat{\sigma} \in \text{Aut}_{\mathbb{F}}(A)$ .
- A *minimal*  $\sigma$ -cyclic CC  $\mathcal{C}$  is defined to be a  $\sigma$ -cyclic CC that has no non-trivial  $\sigma$ -cyclic sub-CC. Each  $\sigma$ -cyclic CC  $\mathcal{C}$  can be decomposed into a direct sum  $\mathcal{C} = \bigoplus_{i=1}^s \mathcal{C}_i$ , where the  $\mathcal{C}_i$  are minimal  $\sigma$ -cyclic CC’s.
- $\mathcal{C}$  is a minimal  $\sigma$ -cyclic CC if and only if its generator polynomial  $g$  satisfies  $g = g^{(\ell)}$  for some  $1 \leq \ell \leq r$ .
- For a minimal  $\sigma$ -cyclic CC  $\mathcal{C}$  with  $g = g^{(\ell)}$  and  $k = \deg_x \pi_\ell$ , the complexity of  $\mathcal{C}$  is  $\delta = k \cdot d$  for some  $d \in \mathbb{N}$ . Moreover, we have the following equivalence: For any  $d \in \mathbb{N}_0$  there exists a minimal  $\sigma$ -cyclic  $(n, k, kd)$ -CC with generator polynomial  $g$  satisfying  $g = g^{(\ell)}$  if and only if  $\sigma(\varepsilon^{(\ell)}) \neq \varepsilon^{(\ell)}$ .
- Within the class of cyclic CC’s, *Reed-Solomon* (RS) and *BCH convolutional codes* can be defined. They contain (near) optimal codes w.r.t. distance and performance (Gluesing-Luerssen and Schmale 2006). In particular, one can construct cyclic one-dimensional MDS convolutional codes with a RS structure (Gluesing-Luerssen and Langfeld 2006a).

**Acknowledgement** The authors acknowledge support from the Austrian Academy of Sciences (Semester on Gröbner bases, Linz, Austria, 2006).

## References

- H. Gluesing-Luerssen and B. Langfeld, *A class of one-dimensional MDS convolutional codes*, J. Algebra Appl. **5** (2006a), no. 4, 505–520.
- H. Gluesing-Luerssen and B. Langfeld, *On the algebraic parameters of convolutional codes with cyclic structure*, J. Algebra Appl. **5** (2006b), no. 1, 53–76.
- H. Gluesing-Luerssen and W. Schmale, *On cyclic convolutional codes*, Acta Appl. Math. **82** (2004), no. 2, 183–237.
- H. Gluesing-Luerssen and W. Schmale, *On doubly-cyclic convolutional codes*, AAECC **17** (2006), no. 2, 151–170.
- M. Giorgetti, M. Rossi, and M. Sala, *On the Gröbner basis of a family of quasi-cyclic LDPC codes*, Bull. Iran. Math. Soc. **31** (2005), no. 2, 13–32.
- W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*, Cambridge University Press, Cambridge, 2003.
- K. Lally, *Canonical representation of quasicyclic codes using Gröbner basis theory*, this volume, 2009, pp. 351–355.

- K. Lally and P. Fitzpatrick, *Algebraic structure of quasicyclic codes*, Discrete Appl. Math. **111** (2001), nos. 1–2, 157–175.
- R. J. McEliece, *The algebraic theory of convolutional codes*, Handbook of Coding Theory, vol. **1**, Elsevier, Amsterdam, 1998, pp. 1065–1138.
- P. Piret, *Structure and constructions of cyclic convolutional codes*, IEEE Trans. on Inf. Th. **22** (1976), no. 2, 147–155.
- C. Roos, *On the structure of convolutional and cyclic convolutional codes*, IEEE Trans. on Inf. Th. **25** (1979), no. 6, 676–683.
- J. Rosenthal and R. Smarandache, *Maximum distance separable convolutional codes*, AAECC **10** (1999), no. 1, 15–32.
- G. Solomon and H. C. A. van Tilborg, *A connection between block and convolutional codes*, SIAM J. Appl. Math. **37** (1979), no. 2, 358–369.

# On the Non-linearity of Boolean Functions

Ilaria Simonetti

**Abstract** We compute the non-linearity of Boolean functions with Gröbner bases.

## 1 Introduction

Any function from  $(\mathbb{F}_2)^n$  to  $\mathbb{F}_2$  is called a Boolean function (Bf). Boolean functions are important in symmetric cryptography, since they are used in the confusion layer of ciphers. An affine Bf does not provide an effective confusion. To overcome this, we need functions which are as far as possible from being an affine function. The effectiveness of these functions is measured by a parameter called “non-linearity” (Carlet 2009). Usually, to compute the non-linearity of a Bf  $f$ , we have to compute the discrete Fourier transform  $\hat{f}_\chi$  of the function  $f_\chi(x) = (-1)^{f(x)}$ . Then the non-linearity of  $f$  is  $N(f) = 2^{n-1} - \frac{1}{2} \max_{a \in (\mathbb{F}_2)^n} |\hat{f}_\chi(a)|$ .

In this paper, we compute the non-linearity of Bf’s with Gröbner bases (Sala and Simonetti 2007).

## 2 Preliminaries and Notation

Let  $\mathbb{F}_2$  be the field with 2 elements. Let  $n \geq 1$  be an integer. From now on,  $n$  and an ordering on vectors in  $(\mathbb{F}_2)^n = \{v_1, \dots, v_{2^n}\}$  are understood.

We denote by  $\mathcal{B}_n$  the set of all Bf’s. It is well-known that  $f$  can be expressed as a polynomial in  $\mathbb{F}_2[X] = \mathbb{F}_2[x_1, \dots, x_n]$ , as follows

$$f = \sum_{S \subset \{1, \dots, n\}} b_S X_S, \quad \text{where } X_S = x_{i_1} \cdots x_{i_{|S|}}, S = \{i_1, \dots, i_{|S|}\}.$$

**Definition 1** Let  $f, g \in \mathcal{B}_n$ . The *distance*  $d(f, g)$  between  $f$  and  $g$  is the number of  $v \in (\mathbb{F}_2)^n$  such that  $f(v) \neq g(v)$ .

We denote by  $\mathcal{A}_n$  the set of all affine Bf’s:  $\mathcal{A}_n = \{a_0 + \sum_{i=1}^n a_i x_i \mid a_i \in \mathbb{F}_2\}$ , where  $a_i = a_{\{i\}}$  and  $a_\emptyset = a_\emptyset$ .

---

I. Simonetti

Department of Mathematics, University of Milano, Milan, Italy

e-mail: [simonet@mat.unimi.it](mailto:simonet@mat.unimi.it)

Let  $A$  and  $B$  be the following variable sets:  $A = \{a_i\}_{0 \leq i \leq n}$ ,  $B = \{b_S\}_{S \subset \{1, \dots, n\}}$ . We denote by  $\mathbf{g}_n \in \mathbb{F}_2[A, X] \subset \mathbb{F}_2[A, B, X]$ ,  $\mathbf{f}_n \in \mathbb{F}_2[B, X] \subset \mathbb{F}_2[A, B, X]$  the following polynomials:

$$\mathbf{g}_n = a_0 + \sum_{i=1}^n a_i x_i, \quad \mathbf{f}_n = \sum_{S \subset \{1, \dots, n\}} b_S X_S.$$

**Definition 2** (Carlet 2009) Let  $f \in \mathcal{B}_n$ . The *non-linearity*  $N(f)$  of  $f$  is the minimum of the distances between  $f$  and any affine function:

$$N(f) = \min_{\alpha \in \mathcal{A}_n} d(f, \alpha).$$

An upper bound on the non-linearity for a Boolean function  $f$  is (Carlet 2009):

$$N(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1}.$$

This upper bound can only be met if  $n$  is even (*bent functions*).

We consider a map from  $\mathcal{B}_n$  to  $(\mathbb{F}_2)^{2^n}$ , which sends a Bf  $f$  into a vector  $\underline{f} = (f(v_1), \dots, f(v_{2^n}))$ , obtained by evaluating  $f$ .

We denote by  $S_{\mathcal{A}_n}(f)$  the set

$$S_{\mathcal{A}_n}(f) = \{\underline{f} + \underline{g} \mid g \in \mathcal{A}_n\} \subset (\mathbb{F}_2)^{2^n}.$$

The following lemma is obvious:

**Lemma 1** *Let  $f, g$  be two Boolean functions. Then*

$$d(f, g) = d(\underline{f}, \underline{g}) = w(\underline{f} + \underline{g}),$$

where  $d$  ( $w$ ) is the Hamming distance (weight) in  $(\mathbb{F}_2)^{2^n}$ .

Therefore, computing the non-linearity of  $f \in \mathcal{B}_n$  is the same as finding the minimum weight of vectors in set  $S_{\mathcal{A}_n}(f)$ .

We can extend the evaluation to polynomials  $\mathbf{g}_n$  and  $\mathbf{f}_n$  as follows:

$$\underline{\mathbf{g}_n} = (\mathbf{g}_n(A, v_1), \dots, \mathbf{g}_n(A, v_{2^n})) \in (\mathbb{F}_2[A])^{2^n},$$

$$\underline{\mathbf{f}_n} = (\mathbf{f}_n(B, v_1), \dots, \mathbf{f}_n(B, v_{2^n})) \in (\mathbb{F}_2[B])^{2^n}.$$

For the remainder of this section, we recall some definitions and results about the weight of vectors in  $(\mathbb{F}_2)^n$ , taken from Guerrini et al. (2006).

Let  $1 \leq t \leq m$  be integers. We denote by  $E[Y]$  the following set of polynomials in  $\mathbb{F}_2[Y] = \mathbb{F}_2[y_1, \dots, y_m]$ :  $E[Y] = \{y_1^2 + y_1, \dots, y_m^2 + y_m\}$ .

**Definition 3** For  $\mathbf{m} \in \mathbb{F}_2[Y]$ ,  $\mathbf{m}$  is a *square-free t-monomial* if:

$$\mathbf{m} = y_{h_1} \cdots y_{h_t}, \quad \text{where } h_1, \dots, h_t \in \{1, \dots, m\} \text{ and } h_l \neq h_j, \forall l \neq j,$$

i.e. a monomial in  $\mathbb{F}_2[Y]$  such that  $\deg_{y_{h_i}}(\mathbf{m}) = 1$  for any  $1 \leq i \leq t$ . We denote by  $\mathcal{M}_{m,t}$  the set of all square-free  $t$ -monomials in  $\mathbb{F}_2[Y]$ .

Let  $I_{m,t} \subset \mathbb{F}_2[Y]$  be the following ideal

$$I_{m,t} = \langle \{\sigma_t, \dots, \sigma_m\} \cup E[Y] \rangle,$$

where  $\sigma_i$  are the elementary symmetric functions in variables  $Y$ .

For any  $1 \leq i \leq m$ , let  $P_i = \{v \in (\mathbb{F}_2)^m \mid w(v) = i\}$  and  $Q_i = \bigcup_{0 \leq j \leq i} P_j$ .

**Theorem 1** (Guerrini et al. 2006) *The vanishing ideal  $\mathcal{I}(Q_t)$  of  $Q_t$  is*

$$\mathcal{I}(Q_t) = I_{m,t+1},$$

and its reduced Gröbner basis  $G$  (w.r.t. any ordering) is

$$G = E[Y] \cup \mathcal{M}_{m,t}, \quad \text{for } t \geq 2,$$

$$G = \{y_1, \dots, y_m\}, \quad \text{for } t = 1.$$

### 3 Computing the Non-linearity

In this section we show how to use Theorem 1 to compute the non-linearity for a Boolean function  $f$ . We define an ideal where  $\mathbf{g}_n$  plays the role of a generic affine function. A point in its variety corresponds to an affine function with distance from  $f$  at most  $t - 1$ .

**Definition 4** Let  $f \in \mathcal{B}_n$ . We denote by  $J_t^n(f)$  the ideal in  $\mathbb{F}_2[A]$ :

$$\begin{aligned} J_t^n(f) &= \langle \{\mathbf{m}(\mathbf{g}_n(A, v_1) + f(v_1), \dots, \mathbf{g}_n(A, v_{2^n}) + f(v_{2^n})) \mid \mathbf{m} \in \mathcal{M}_{2^n,t} \cup E[A]\} \rangle \\ &= \langle \{\mathbf{m}(\underline{\mathbf{g}_n} + \underline{f}) \mid \mathbf{m} \in \mathcal{M}_{2^n,t}\} \cup E[A] \rangle. \end{aligned}$$

*Remark 1* Since  $E[A] \subset J_t^n(f)$ ,  $J_t^n(f)$  is zero-dimensional and radical.

**Lemma 2** Let  $f \in \mathcal{B}_n$ . Let  $t \in \mathbb{N}$  such that  $1 \leq t \leq 2^n$ . Then the following statements are equivalent:

1.  $\mathcal{V}(J_t^n(f)) \neq \emptyset$
2.  $\exists u \in S_{\mathcal{A}_n}(f)$  such that  $w(u) \leq t - 1$
3.  $\exists \alpha \in \mathcal{A}_n$  such that  $d(f, \alpha) \leq t - 1$ .

From Lemma 2 we immediately have the following theorem.

**Theorem 2** Let  $f \in \mathcal{B}_n$ . The non-linearity  $N(f)$  is the minimum  $t$  such that  $\mathcal{V}(J_{t+1}^n(f)) \neq \emptyset$ .

From this theorem we can derive an algorithm to compute the non-linearity for a function  $f \in \mathcal{B}_n$ , by computing any Gröbner basis of  $J_t^n(f)$ .

```

 $j = 1$ 
While  $\mathcal{V}(J_j^n(f)) = \emptyset$  do
     $j := j + 1;$ 
Output  $j - 1$ 

```

*Remark 2* If  $f$  is not affine, we can start our check from  $J_2^n(f)$ .

*Example 1* Let  $f : (\mathbb{F}_2)^3 \rightarrow \mathbb{F}_2$  be the Boolean function:

$$f(x_1, x_2, x_3) = x_1 x_2 + x_1 x_3 + x_2 + 1.$$

We want to compute  $N(f)$  and clearly  $f$  is not affine. We compute vector  $\underline{f}$  and we take a generic affine function  $\underline{g}_3$ , so that:  $\underline{f} = (1, 1, 0, 1, 1, 0, 0, 0)$ ,  $\underline{g}_3 = (\overline{a}_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2, a_0 + a_3, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3)$ . So  $\underline{f} + \underline{g}_3 = (a_0 + 1, a_0 + a_1 + 1, a_0 + a_2, a_0 + a_1 + a_2 + 1, a_0 + a_3 + 1, a_0 + a_1 + a_3, a_0 + a_2 + a_3, a_0 + a_1 + a_2 + a_3) = (p_1, p_2, \dots, p_8)$ . Ideal  $J_2^3(f)$  is the ideal generated by

$$J_2^3(f) = \langle \{p_1 p_2, p_1 p_3, \dots, p_7 p_8\} \cup \{a_0^2 + a_0, a_1^2 + a_1, a_2^2 + a_2, a_3^2 + a_3\} \rangle.$$

We compute any Gröbner basis of this ideal and we obtain that it is trivial, so  $\mathcal{V}(J_2^3(f)) = \emptyset$  and  $N(f) > 1$ . Now we have to compute a Gröbner basis for  $J_3^3(f)$ . We obtain, using degrevlex ordering with  $a_3 < a_2 < a_1 < a_0$ , that  $G(J_3^3(f)) = \{a_2 + a_3 + 1, a_3^2 + a_3, a_1 a_3 + a_0 + 1, a_0 a_3 + a_0 + a_3 + 1, a_1^2 + a_1, a_0 a_1 + a_0 + a_1 + 1, a_0^2 + a_0\}$ . So,  $N(f) = 2$  by Theorem 2. By inspecting  $G(J_3^3(f))$ , we also obtain all affine functions having distance 2 from  $f$ :

$$\alpha_1 = 1 + x_1 + x_2, \quad \alpha_2 = 1 + x_2, \quad \alpha_3 = 1 + x_3, \quad \alpha_4 = x_1 + x_3.$$

*Remark 3* Our method for computing the non-linearity is equivalent to decoding  $t$  errors with respect to the Reed–Muller code  $\text{RM}(1, m)$ , with  $t$  increasing until there is at least a solution.

**Acknowledgements** The author would like to thank her supervisor M. Sala. For their comments and suggestions, the author heartily thanks L. Budaghyan, E. Guerrini, E. Orsini, L. Perret, C. Traverso.

This work has been partially supported by STMicroelectronics contract “Complexity issues in algebraic Coding Theory and Cryptography”.

This work has been presented at Workshop D1 (Linz, 2006) and we acknowledge support from the Austrian Academy of Science.

## References

- C. Carlet, *Boolean methods and models*, ch. Boolean Functions for Cryptography and Error Correcting Codes, Cambridge University Press, 2009, to appear.
- E. Guerrini, M. Orsini, and M. Sala, *Computing the distance distribution of systematic non-linear codes*, BCRI preprint, [www.bcri.ucc.ie](http://www.bcri.ucc.ie) 50, UCC, Cork, Ireland, 2006.
- M. Sala and I. Simonetti, *An algebraic description of Boolean functions*, Proc. of WCC 2007 (2007), 343–349.

# Quasigroups as Boolean Functions, Their Equation Systems and Gröbner Bases

D. Gligoroski, V. Dimitrova and S. Markovski

**Abstract** In this short note we represent quasigroups of order  $2^n$  as vector valued Boolean functions  $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ . The representation of finite quasigroups as vector valued Boolean functions allows us systems of quasigroup equations to be solved by using Gröbner bases.

**Keywords** Quasigroups · Boolean functions · Classifications

## 1 Introduction

The quasigroup structures, their properties and especially their large number, enable them to be applied in many theories, like experimental designs, telecommunications, cryptography, coding theory and many more.

In this short note we will give a novel representation of quasigroups of order  $2^n$  as vector valued Boolean functions  $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ . We use this representation to classify the finite quasigroups according to the degree of polynomials involved in their representation. We will give the summary of a classification on linear and nonlinear quasigroups of order 4. The techniques presented here, in general, can be applied on quasigroups of any finite order  $2^n$ . For  $n \geq 3$  the classification can be more profound, one can consider linear quasigroups, quadratic quasigroups, third degree quasigroups, and so on.

We start by a brief introduction to the notion of quasigroup. Then we introduce the novel representation of quasigroups as Boolean functions. Next, we give a list of linear and nonlinear quasigroups of order 4. Finally, we show how Gröbner bases techniques (Mora 2009) can be naturally applied to solving equation systems in

---

D. Gligoroski

Centre for Quantifiable Quality of Service in Communication Systems—Q2S, Norwegian University of Science and Technology, Trondheim, Norway  
e-mail: [danielog@q2s.ntnu.no](mailto:danielog@q2s.ntnu.no)

V. Dimitrova · S. Markovski

Institute of Informatics, Faculty of Natural Sciences and Mathematics, Ss Cyril and Methodius University, Skopje, Macedonia

V. Dimitrova

e-mail: [vesnap@ii.edu.mk](mailto:vesnap@ii.edu.mk)

S. Markovski

e-mail: [smile@ii.edu.mk](mailto:smile@ii.edu.mk)

quasigroup modulo our representation. We are aware of alternative techniques conceptually different from a brute-force check.

## 2 Quasigroups as Vector Valued Boolean Functions

A quasigroup is a groupoid  $(Q, *)$ , i.e., a set  $Q$  endowed with a binary operation  $* : Q^2 \rightarrow Q$ , such that the equations  $x * a = b$ ,  $a * y = b$  have unique solutions  $x, y$ , for each given  $a, b \in Q$ . The uniqueness of the solutions of the above equations implies that the cancellation laws  $x * y = x * z \implies y = z$ ,  $y * x = z * x \implies y = z$  are satisfied in  $(Q, *)$  and vice versa.

Equivalent combinatorial structure to quasigroups are the Latin squares. To any finite quasigroup  $(Q, *)$ , given by its multiplication table, a Latin square consisting of the matrix formed by the main body of the table can be associated, since each row and column of the matrix is a permutation of  $Q$ .

### 2.1 Lexicographic Ordering of Finite Quasigroups

We need an ordering of the set of quasigroups of given order, and we use the lexicographic ordering as follows. Given the set of all quasigroups of order  $n$ , we represent each quasigroup as one string of  $n^2$  letters that is concatenation of the rows of its corresponding Latin square. Then the ordering of the quasigroups is given by the lex ordering of their representations.

*Example 1* There are 576 quasigroups of order 4. For quasigroups shown below, the corresponding indexes in the lex ordering are: 1, 168, 576.

*	0 1 2 3	*	0 1 2 3	*	0 1 2 3
0	0 1 2 3	0	1 0 2 3	0	3 2 1 0
1	1 2 3 0	1	3 2 1 0	1	2 1 0 3
2	2 3 0 1	2	2 3 0 1	2	1 0 3 2
3	3 0 1 2	3	0 1 3 2	3	0 3 2 1

### 2.2 Vector Valued Boolean Functions

Quasigroups have many equivalent representations (see for example Colbourn and Dinitz 1996; White paper 2009). However, so far we have not found in the literature their representation and exploitation as vector valued Boolean functions.

The idea is straightforward and we present it for quasigroups of order  $2^n$ . Let  $\mathbb{F}_2 = \{0, 1\}$  be the two-element field.

- (1) A Boolean function of  $n$  variables is a function  $f : (\mathbb{F}_2)^n = \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ .
- (2) A vector valued Boolean function is a map  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ , where  $m \geq 2$ .
- (3) Every Boolean function can be uniquely given in its algebraic normal form, i.e., as a polynomial in  $n$  variables over the field  $\mathbb{F}_2$  that has degree  $\leq 1$  in each single variable:

$$f(x_1, x_2, \dots, x_n) = \sum_{I \subseteq \{1, \dots, n\}} a_I x^I \quad (1)$$

where the monomial  $x^I$  is the product  $x^I = \prod_{i \in I} x_i$ ,  $x^\emptyset = 1$  and  $a_I \in \{0, 1\}$ .

Now, using the definitions (1) and (2) and the property (3), we can represent every quasigroup  $(Q, *)$  of order  $2^n$  by a vector valued Boolean function  $f : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ . Namely, we (may) suppose that the elements  $x$  of the quasigroup are given as binary vectors  $x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ . Then for each  $x, y \in Q$  we have that

$$x * y \equiv f(x_1, \dots, x_{2n}) = (f_1(x_1, \dots, x_{2n}), \dots, f_n(x_1, \dots, x_{2n}))$$

where  $x = (x_1, x_2, \dots, x_n)$ ,  $y = (x_{n+1}, x_{n+2}, \dots, x_{2n})$  and  $f_i : \{0, 1\}^{2n} \rightarrow \{0, 1\}$  are the corresponding components of  $f$ .

*Example 2* Let us take the second quasigroup given in Example 1. This quasigroup can be represented by the following vector valued Boolean function, i.e., as a pair of polynomials in  $\mathbb{F}_2[X_1, X_2, X_3, X_4]$ :

$$f(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3, 1 + x_1 + x_3 + x_1 x_3 + x_2 x_3 + x_4).$$

### 2.3 Classification of Quasigroups

There are several classifications of quasigroups. By the algebraic properties of the quasigroups two main classifications are obtained: classes of isotopic and classes of isomorphic quasigroups (known only for quasigroups of order  $n \leq 11$  Dénes and Keedwell 1974; McKay 2009). Also, there are some other classifications such as by random walk on torus or by graphical presentation of sequences obtained by quasigroup transformations (Markovski et al. 2005; Dimitrova 2005).

The following novel classification of quasigroups follows immediately from their representation as vector valued Boolean functions. Let  $(Q, *)$  be a quasigroup of order  $2^n$  and let

$$f(x_1, \dots, x_{2n}) = (f_1(x_1, \dots, x_{2n}), \dots, f_n(x_1, \dots, x_{2n}))$$

be its corresponding representation as vector valued Boolean function.

1. If all functions  $f_i$  for  $i = 1, 2, \dots, n$  are linear polynomials, then this quasigroup is called *linear quasigroup*.

2. Otherwise, if there exist function  $f_i$  for some  $i = 1, 2, \dots, n$  which is not linear, this quasigroup is called *nonlinear quasigroup*.

Considering the class of quasigroups of order 4, it can be checked that there are 144 linear and 432 non-linear quasigroups, i.e., there are three times more non-linear quasigroups of order 4. By several experiments we have made on quasigroups of order  $2^n$ , it can be concluded that the quotient  $\#(\text{linear quasigroups})/\#(\text{non-linear quasigroups})$  is going to 0 when  $n$  is going to infinity.

### 3 Systems of Quasigroup Equations and Gröbner Bases

Let us take again the quasigroup with the lexicographic index 168 (from Example 2) and let us consider the following system of quasigroup equations:

$$\begin{cases} (y_4 * (y_2 * ((y_2 * ((y_2 * (((y_3 * (y_2 * y_4)) * y_4) * y_4)) * y_1)) * y_1))) = 3 \\ (y_1 * (y_3 * (y_2 * (y_1 * (y_3 * (y_2 * ((y_1 * (y_2 * y_2)) * y_2))))))) = 0 \\ ((y_4 * (y_3 * (y_1 * (y_4 * (((y_1 * (y_2 * y_4)) * y_3) * y_3)))) * y_1)) = 2 \\ (y_2 * (y_1 * ((y_2 * ((y_4 * (y_4 * (y_2 * (y_3 * y_2)))) * y_3)) * y_2))) = 3 \end{cases} \quad (2)$$

where  $y_1, y_2, y_3, y_4 \in \{0, 1, 2, 3\}$ .

Neither we can get rid of the parentheses because the quasigroup is non-commutative nor we can change the order of the parentheses, because the quasigroup is non-associative. So, it seems that if we stay with the original  $*$  representation of the quasigroup—we would have to apply exhaustive search in the set of all  $4^4$  possibilities. This may be still feasible for this small example, but for some bigger systems that exhaustive search approach soon would become infeasible. However, if we represent the variables as  $y_1 = (x_1, x_2)$ ,  $y_2 = (x_3, x_4)$ ,  $y_3 = (x_5, x_6)$ ,  $y_4 = (x_7, x_8)$  and if we use the Boolean representation of the quasigroup, we will obtain the system of quasigroup (3).

If we try to solve the system (3) by Gröbner bases in  $\mathbb{F}_2$  it will take a fraction of a second to solve it.

We have tested this approach in numerous cases for solving systems of quasigroup equations with up to 40 variables (equivalent to 80 binary variables) and Gröbner bases approach was able to solve it every time. Better results were obtained by using the software package PolyBoRi (Brickenstein and Dreyer 2007), when systems of quasigroups equations of up to 50 variables could be solved.

$$\begin{aligned}
& 1 + x_2 + x_1x_3 + x_4 + x_3x_4 + x_1x_3x_4 + x_5 + x_1x_3x_5 + x_1x_4x_5 + x_1x_6 + x_7 \\
& \quad + x_1x_7 + x_3x_7 + x_1x_3x_7 + x_4x_7 + x_3x_4x_7 + x_1x_3x_4x_7 + x_1x_3x_5x_7 \\
& \quad + x_1x_4x_5x_7 + x_6x_7 + x_3x_6x_7 + x_4x_6x_7 + x_8 + x_7x_8 + x_1x_7x_8 \\
& \quad + x_1x_3x_7x_8 + x_1x_4x_7x_8 = 1
\end{aligned}$$

$$\begin{aligned}
& 1 + x_2 + x_1x_2 + x_3 + x_1x_3 + x_2x_3 + x_2x_4 + x_1x_3x_4 + x_1x_5 + x_6 + x_1x_6 \\
& \quad + x_1x_3x_6 + x_1x_4x_6 + x_7 + x_2x_7 + x_3x_7 + x_4x_7 + x_3x_4x_7 + x_1x_3x_4x_7 \\
& \quad + x_5x_7 + x_3x_5x_7 + x_4x_5x_7 + x_6x_7 + x_3x_6x_7 + x_1x_3x_6x_7 + x_4x_6x_7 \\
& \quad + x_1x_4x_6x_7 + x_8 + x_2x_8 + x_1x_3x_8 + x_4x_8 + x_3x_4x_8 + x_1x_3x_4x_8 + x_5x_8 \\
& \quad + x_1x_3x_5x_8 + x_1x_4x_5x_8 + x_1x_6x_8 + x_7x_8 + x_1x_7x_8 + x_3x_7x_8 \\
& \quad + x_4x_7x_8 + x_1x_4x_7x_8 + x_3x_4x_7x_8 + x_1x_3x_4x_7x_8 + x_1x_3x_5x_7x_8 + x_1x_4x_5x_7x_8 \\
& \quad + x_6x_7x_8 + x_3x_6x_7x_8 + x_4x_6x_7x_8 = 1
\end{aligned}$$

$$x_2 + x_4 + x_1x_4 + x_2x_4 + x_3x_4 = 0$$

$$1 + x_1 + x_2 + x_2x_3 + x_1x_3x_4 + x_2x_3x_4 = 0$$

$$\begin{aligned}
& x_2 + x_1x_2 + x_3 + x_1x_3 + x_2x_3 + x_4 + x_5 + x_1x_2x_5 + x_3x_5 + x_1x_3x_5 \\
& \quad + x_2x_3x_5 + x_1x_6 + x_1x_3x_6 + x_2x_3x_6 + x_4x_6 + x_1x_4x_6 + x_2x_4x_6 \\
& \quad + x_5x_6 + x_2x_5x_6 + x_3x_5x_6 + x_1x_3x_5x_6 + x_2x_3x_5x_6 + x_1x_4x_5x_6 + x_2x_4x_5x_6 \\
& \quad + x_7 + x_1x_7 + x_2x_7 + x_1x_3x_7 + x_2x_3x_7 + x_1x_4x_7 + x_2x_4x_7 + x_5x_7 + x_1x_5x_7 \\
& \quad + x_2x_5x_7 + x_3x_5x_7 + x_1x_3x_5x_7 + x_2x_3x_5x_7 + x_4x_5x_7 + x_1x_4x_5x_7 + x_2x_4x_5x_7 \\
& \quad + x_1x_6x_7 + x_2x_6x_7 + x_3x_6x_7 + x_4x_6x_7 + x_5x_6x_7 + x_1x_5x_6x_7 + x_2x_5x_6x_7 \\
& \quad + x_3x_5x_6x_7 + x_4x_5x_6x_7 + x_1x_8 + x_2x_8 + x_5x_8 + x_1x_5x_8 + x_2x_5x_8 \\
& \quad + x_6x_8 + x_5x_6x_8 = 1
\end{aligned}$$

$$\begin{aligned}
& x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4 + x_2x_4 + x_1x_5 + x_2x_5 + x_1x_2x_5 + x_3x_5 + x_1x_3x_5 \\
& \quad + x_2x_3x_5 + x_1x_2x_3x_5 + x_1x_4x_5 + x_2x_4x_5 + x_1x_6 + x_1x_3x_6 + x_1x_2x_3x_6 \\
& \quad + x_1x_2x_4x_6 + x_1x_5x_6 + x_1x_2x_5x_6 + x_1x_2x_3x_5x_6 + x_1x_4x_5x_6 + x_1x_2x_4x_5x_6 \\
& \quad + x_1x_7 + x_2x_7 + x_1x_2x_7 + x_3x_7 + x_1x_3x_7 + x_1x_2x_3x_7 + x_4x_7 + x_1x_4x_7 \\
& \quad + x_1x_2x_4x_7 + x_5x_7 + x_1x_5x_7 + x_2x_5x_7 + x_1x_2x_5x_7 + x_3x_5x_7 + x_1x_2x_3x_5x_7 \\
& \quad + x_4x_5x_7 + x_1x_2x_4x_5x_7 + x_1x_6x_7 + x_1x_2x_6x_7 + x_1x_3x_6x_7 + x_1x_4x_6x_7 \\
& \quad + x_1x_2x_5x_6x_7 + x_1x_3x_5x_6x_7 + x_1x_4x_5x_6x_7 + x_8 + x_1x_8 + x_1x_2x_8 + x_5x_8 \\
& \quad + x_1x_2x_5x_8 + x_1x_6x_8 + x_1x_5x_6x_8 = 0
\end{aligned}$$

$$x_1 + x_2 + x_4 + x_5 + x_4x_6 + x_3x_4x_6 + x_4x_5x_6 = 1$$

$$\begin{aligned}
& 1 + x_2 + x_1x_4 + x_2x_4 + x_1x_5 + x_2x_5 + x_1x_4x_6 + x_2x_4x_6 + x_3x_4x_6 + x_1x_3x_4x_6 \\
& \quad + x_2x_3x_4x_6 + x_4x_5x_6 + x_1x_4x_5x_6 + x_2x_4x_5x_6 = 1
\end{aligned} \tag{3}$$

For other applications of Gröbner bases to Boolean functions, see Sala and Simonetti (2007), Simonetti (2009), Gligoroski et al. (2009).

**Acknowledgement** The authors acknowledge support from the Austrian Academy of Sciences during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- M. Brickenstein and A. Dreyer, *PolyBoRi: A framework for Gröbner basis computations with Boolean polynomials*, Elec. Proc. of MEGA 2007, 2007, <http://www.ricam.oeaw.ac.at/mega2007/electronic/26.pdf>.
- C. J. Colbourn and J. Dinitz, *CRC Handbook of Combinatorial Design*, CRC, Boca Raton, 1996.
- J. Dénes and A. D. Keedwell, *Latin squares and their applications*, Academic, New York, 1974.
- V. Dimitrova, *Kvazigrupni transformacii i nivni primeni (Quasigroup transformations and their applications)*, Msc thesis, Ss. Cyril and Methodius University, Skopje, 2005.
- D. Gligoroski, S. Markovski, and S. J. Knapskog, *A new measure to estimate pseudo-randomness of Boolean functions and relations with Gröbner bases*, this volume, 2009, pp. 421–425.
- S. Markovski, D. Gligoroski, and J. Markovski, *Classification of quasigroups by random walk on torus*, J. Appl. Math. Comput. **19** (2005), nos. 1–2, 57–75.
- B. McKay, *Latin squares*, 2009, <http://cs.anu.edu.au/bdm/data/latin.html>.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- I. Simonetti, *On the non-linearity of Boolean functions*, this volume, 2009, pp. 409–413.
- M. Sala and I. Simonetti, *An algebraic description of boolean functions*, Proc. of WCC 2007 (2007), 343–349.
- White paper, *The Encyclopaedia of Design Theory*, 2009, <http://www.designtheory.org/library/encyc/topics/lsee.pdf>.

# A New Measure to Estimate Pseudo-Randomness of Boolean Functions and Relations with Gröbner Bases

Danilo Gligoroski, Smile Markovski and Svein Johan Knapskog

**Abstract** In this short note we will introduce a generic measure of the algebraic complexity of vector valued Boolean functions: Normalized Average Number of Terms (NANT). NANT can be considered as a tool that extracts those vector valued Boolean functions that are suitable for effective application of Gröbner bases. As an example, we use NANT to show clear differences between two popular cryptographic hash functions: SHA-1 and SHA-2. The obtained results show that SHA-1 is susceptible to attacks based on Gröbner bases, which lead us to believe that SHA-1 is much weaker than SHA-2 from a design point of view.

**Keywords** NANT · Hash · SHA-1 · SHA-2

## 1 Introduction

The complexity of Boolean functions has been a subject of cryptographic scrutiny for a long time and a lot of different types of Boolean functions and different measures have been introduced so far. An excellent review article covering recent developments in this field is the paper of Qu et al. (2001). The complexity of the Boolean functions is not strongly connected with their algebraic degree (see e.g. Simonetti 2009), and our measure proposed here does not depend on the degree of the function.

---

D. Gligoroski · S.J. Knapskog

Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, O.S. Bragstads plass 2E, 7491 Trondheim, Norway

D. Gligoroski

e-mail: [Danilo.Gligoroski@q2s.ntnu.no](mailto:Danilo.Gligoroski@q2s.ntnu.no)

S.J. Knapskog

e-mail: [Svein.J.Knapskog@q2s.ntnu.no](mailto:Svein.J.Knapskog@q2s.ntnu.no)

S. Markovski

Faculty of Natural Sciences and Mathematics, Institute of Informatics, “Ss Cyril and Methodius” University, P.O. Box 162, 1000 Skopje, Macedonia

e-mail: [smile@ii.edu.mk](mailto:smile@ii.edu.mk)

We can say that there is very close analogy between our introduced measure NANT— $\overline{L}_f(k)$  for a specific value  $k$  and the propagation criterion of degree  $l$  and order  $k$  ( $PC(l)$  of order  $k$ ) introduced by Preneel et al. (1991). Stronger mathematical relations between NANT and  $PC(l)$  of order  $k$  form an interesting research topic in itself.

## 2 Normalized Average Number of Terms—NANT

Let  $n \geq r \geq 1$  be integers and let  $f : \{0, 1\}^n \rightarrow \{0, 1\}^r$  be a vector valued Boolean function. The vector valued function  $f$  can be represented as an  $r$ -tuple of Boolean functions  $f = (f^{(1)}, f^{(2)}, \dots, f^{(r)})$ , where  $f^{(s)} : \{0, 1\}^n \rightarrow \{0, 1\}$  ( $s = 1, 2, \dots, r$ ), and the value of  $f^{(s)}(x_1, \dots, x_n)$  equals the value of the  $s$ -th component of  $f(x_1, \dots, x_n)$ . In finite field theory, Boolean functions are considered as maps  $g : (\mathbb{F}_2)^n \rightarrow \mathbb{F}_2$  and so the Boolean functions  $f^{(s)}(x_1, \dots, x_n)$  can be expressed in the Algebraic Normal Form (ANF) as polynomials in  $\mathbb{F}_2[x_1, \dots, x_n]$ , that is, as polynomials with  $n$  variables  $x_1, \dots, x_n$  of kind  $a_0 \oplus a_1 x_1 \oplus \dots \oplus a_n x_n \oplus a_{1,2} x_1 x_2 \oplus \dots \oplus a_{n-1,n} x_{n-1} x_n \oplus \dots \oplus a_{1,2,\dots,n} x_1 x_2 \dots x_n$ , where  $a_\lambda \in \{0, 1\}$ . Each ANF has up to  $2^n$  terms (i.e., monomials), depending on the values of the coefficients  $a_\lambda$ . Denote by  $L_{f^{(s)}}$  the number of terms in the ANF of the function  $f^{(s)}$  and define the number of terms of the vector valued function  $f$  by  $L_f = \sum_{s=1}^r L_{f^{(s)}}$ .

**Definition 1** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}^r$  be a vector valued Boolean function and  $k \in \{1, \dots, n\}$ . Let the naturally ordered sets  $\sigma_j = \{i_1, i_2, \dots, i_k\} \subset \{1, \dots, n\}$ ,  $1 \leq j \leq S$ , be chosen uniformly at random and let  $f_{\sigma_j}$  denote the restriction of  $f$  defined by  $f_{\sigma_j}(x_1, x_2, \dots, x_n) = f(0, \dots, 0, x_{i_1}, 0, \dots, 0, x_{i_2}, 0, \dots, 0, x_{i_k}, 0, \dots, 0)$ . We define the random variable  $\overline{L}_f(k)$ —the Normalized Average Number of Terms (NANT), by

$$\overline{L}_f(k) = \frac{1}{r} \cdot \frac{1}{2^{k-1}} \cdot \lim_{S \rightarrow \infty} \frac{1}{S} \sum_{j=1}^S L_{f_{\sigma_j}}.$$

Since the subsets  $\sigma_j$  are chosen uniformly at random, the average values of  $L_{f_{\sigma_j}^{(s)}}$  ( $s = 1, 2, \dots, r$ ) are  $2^{k-1}$  and the average value of  $L_{f_{\sigma_j}}$  is  $r2^{k-1}$ . Also,  $0 < L_{f_{\sigma_j}^{(s)}} \leq 2^k$ . So, the following theorem is true:

**Theorem 1** For any function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^r$  chosen uniformly at random from the set of all such functions, and for any  $k \in \{1, \dots, n\}$ , it is true that

$$0 < \overline{L}_f(k) \leq 2$$

and that the expected value is

$$EX(\overline{L}_f(k)) = 1.$$

We quote here a statement given by Faugère and Joux (2003) for effectiveness of computing Gröbner bases: “A crucial point in the cryptanalysis of HFE is the ability to distinguish a ‘random’ (or generic) algebraic system from an algebraic system coming from HFE.” Now, the role of our measure  $\overline{L}_f(k)$  is just the measurement of the non-randomness of Boolean functions. It is enough to compute the values of  $EX(\overline{L}_f(k))$  for several relatively small values of  $k$  ( $k = 3, 4, 5, 6, 7, 8, \dots$ ). Then, if  $EX(\overline{L}_f(k)) \ll 1$  or  $EX(\overline{L}_f(k)) \gg 1$ , we say that the crypt-system represented by  $f$  is weak, and a Gröbner bases attack on a weak crypt-system represented by  $f$  is likely successful. The price of computing  $EX(\overline{L}_f(k))$  is quite small compared to the possible values of  $n : n = 2^{256}, n = 2^{512}, \dots$

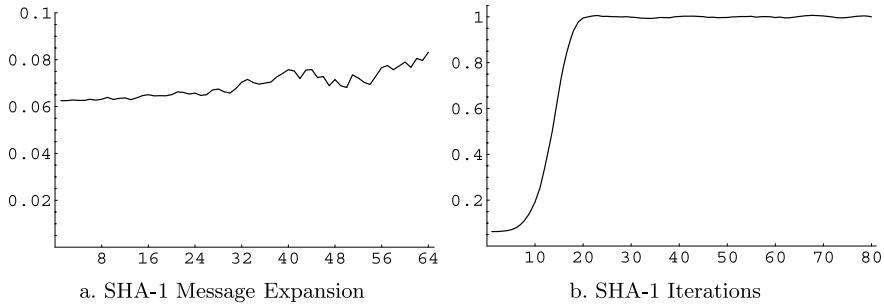
Considering a weak hash function represented by a vector valued Boolean function  $f = (f^{(1)}, f^{(2)}, \dots, f^{(r)})$ , one Gröbner bases attack for finding a collision can be the following. Let  $\sigma_j = \{i_1, i_2, \dots, i_k\} \subset \{1, \dots, n\}$  be given, for some enough small value of  $k$ . Denote  $\mathbf{x} = 0, \dots, 0, x_{i_1}, 0, \dots, 0, x_{i_2}, 0, \dots, 0, x_{i_k}, 0, \dots, 0$  and  $\mathbf{y} = 0, \dots, 0, y_{i_1}, 0, \dots, 0, y_{i_2}, 0, \dots, 0, y_{i_k}, 0, \dots, 0$ , where  $x_\lambda$  and  $y_\lambda$  are variables. Let  $I$  be the ideal in the ring  $\mathbb{F}_2[x_1, \dots, x_n, y_1, \dots, y_n]$  generated by the functions  $f_{\sigma_j}^{(1)}(\mathbf{x}) - f_{\sigma_j}^{(1)}(\mathbf{y}), f_{\sigma_j}^{(2)}(\mathbf{x}) - f_{\sigma_j}^{(2)}(\mathbf{y}), \dots, f_{\sigma_j}^{(r)}(\mathbf{x}) - f_{\sigma_j}^{(r)}(\mathbf{y}), x_1^2 - x_1, \dots, x_n^2 - x_n, y_1^2 - y_1, \dots, y_n^2 - y_n$ . The computation of the Gröbner bases  $G(I)$  for the ideal  $I$  can be effectively done, since  $k$  is chosen small enough. Finally, we have to find the vanishing set  $\mathcal{V}(G(I))$ , and if  $|\mathcal{V}(G(I))| > 1$ , a collision is found.

### 3 NANT and SHA-Family of Hash Functions

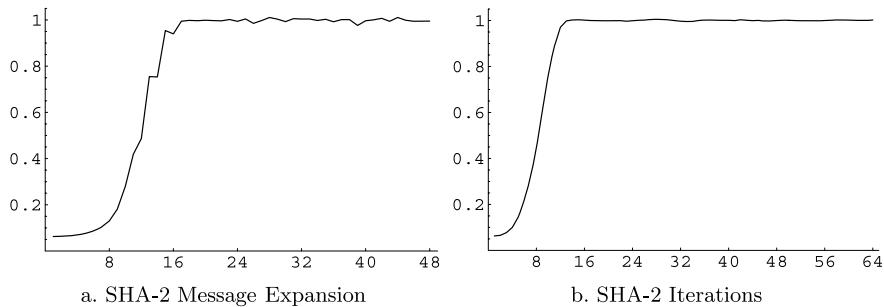
Having a completely linear message expansion part, SHA-1 reaches the level of complexity of a random nonlinear multivariate Boolean function over the field  $\mathbb{F}_2$  in about 20 steps in the iterative part of its compression function. Basically, all known methods for finding collisions on SHA-0 and SHA-1 exploit heavily that weak part of the design (Biham and Chen 2004; Biham et al. 2005; Wang et al. 2005a, 2005b).

Here we take  $n = 512$  and we consider functions  $f : \{0, 1\}^{512} \rightarrow \{0, 1\}^r$ , where  $r \in \{32, 160, 256\}$  depending on whether we measure the complexity in the message expansion part (32-bit variables), the iterative part of SHA-1 (the hash is 160 bits), or the iterative part of SHA-2 (the hash is 256 bits).

The reason why we chose to apply averaging in the definition of NANT was that we wanted to have a tool that will give us a quantitative measure how close to a random Boolean function some iteratively defined hash function is. We think that this measure is more suitable for hash functions than measures that are giving Yes/No answers in the analysis of iteratively defined hash functions. Moreover, having concretely defined compression functions (such as those of SHA-1 and SHA-2) with several hundreds of input bits it would be practically impossible to apply the definitions for the balanced Boolean function in particular for a high order of resiliency or for a high propagation degree (Preneel et al. 1991). Still, we want to stress that we do not consider the value of  $\overline{L}_f(k)$  as an ultimate measure that will unconditionally prove security claims for the hash functions. We see NANT as a tool to conjecture



**Fig. 1** Measuring the Complexity of SHA-1 in the message expansion part and in the iterative part of its compression function



**Fig. 2** Measuring the Complexity of SHA-2 in the message expansion part and in the iterative part of its compression function

how closely and how fast some iterated Boolean functions obtain a property that is true for a random Boolean function.

For small values of  $k$ , i.e.,  $k = 3, 4, \dots, 8$  the values  $\overline{L}_f(k)$  are easily computable and in Figs. 1 and 2 we give graphs for SHA-1 and SHA-2 for their message expansion part and for their iterative part for the value  $k = 5$ . Similar graphs can be obtained for other values of  $k$ .

In Fig. 1a. it can be seen that the message expansion part of SHA-1, being completely linear, never reaches the complexity of a random Boolean function. Further in Fig. 1b we can see that SHA-1 reaches the complexity of a random Boolean function after 20 steps in its iterative part.

The situation with SHA-2 is significantly different. From Fig. 2a we see that the message expansion part of SHA-2 is much better designed and it reaches the same complexity as a random Boolean function after 16 steps, which reflects afterwards in the iterative part of SHA-2 that achieves the complexity level of a random Boolean function after 13 steps (Fig. 2b).

By our observation of SHA-2 we deduce the following remark. The design of SHA-2 may be further improved by starting the computations in its iterative part (of

the compression function) only on the variables produced in the message expansion part having the complexity level of a random Boolean function.

**Acknowledgement** The authors acknowledge support from the Austrian Academy of Sciences during the Special Semester on Gröbner bases (Linz, Austria, 2006).

## References

- E. Biham and R. Chen, *Near-collision of SHA-0*, Proc. of CRYPTO 2004 (M. Franklin, ed.), LNCS, vol. **3152**, Springer, Berlin, 2004, p. 290.
- E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet, and W. Jalby, *Collisions of SHA-0 and reduced SHA-1*, Proc. of EUROCRYPT 2005, LNCS, vol. **3494**, Springer, Berlin, 2005, pp. 36–57.
- J. C. Faugère and A. Joux, *Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases*, LNCS, vol. **2729**, Springer, Berlin, 2003, pp. 44–60.
- B. Preneel, R. Govaerts, and J. Varitlevale, *Boolean functions satisfying higher order propagation criteria*, Proc. of EUROCRYPT1991, LNCS, vol. **547**, Springer, Berlin, 1991, pp. 141–152.
- C. Qu, J. Seberry, and T. Xia, *Boolean Functions in Cryptography*, 2001, <http://citeseer.ist.psu.edu/qu01boolean.html>.
- I. Simonetti, *On the non-linearity of Boolean functions*, this volume, 2009, pp. 409–413.
- X. Wang, Y. L. Yin, and H. Yu, *Finding collisions in the full SHA-1*, Proc. of CRYPTO 2005, LNCS, vol. **3621**, Springer, Berlin, 2005a, pp. 17–36.
- X. Wang, H. Yu, and Y. L. Yin, *Efficient collision search attacks on SHA-0*, Proc. of CRYPTO 2005, LNCS, vol. **3621**, Springer, Berlin, 2005b, pp. 1–16.

# Radical Computation for Small Characteristics

Ryutaroh Matsumoto

**Abstract** In applications to coding theory and cryptography, the characteristic of the coefficient field is often small or 2. We will briefly review an algorithm computing the radical of a polynomial ideal specialized for small characteristics.

## 1 Introduction

When we need ideal theoretic symbolic computations of polynomials, the characteristic of the coefficient field is often zero or a small positive number. In applications to coding theory and cryptography, the characteristic is often very small, especially 2. One of important problems is the computation of the radical of an ideal, where the radical of an ideal  $I$  is defined by  $\sqrt{I} = \{x \mid x^n \in I, \text{ for some positive integer } n\}$ . In addition to the radical being important by itself, the radical computation is often the first step for computing the primary decomposition (Gianni et al. 1988; Krick and Logar 1991) and the integral closure of an affine ring (de Jong 1998). There was some difficulty in radical computation of *positive*-dimensional ideals in small positive characteristic, as described below.

The frequently used method of radical computation includes the following steps (Gianni et al. 1988; Krick and Logar 1991): Given a *positive*-dimensional ideal  $I$  of a polynomial ring  $\mathbb{K}[X_1, \dots, X_n]$ , find the set  $\{X_{i_1}, \dots, X_{i_d}\}$  of variables such that  $I$  generates a zero-dimensional proper ideal of the polynomial ring over the coefficient field  $\mathbb{K}(X_{i_1}, \dots, X_{i_d})$ . The problem of radical computation of a general ideal over  $\mathbb{K}$  is reduced to that of a zero-dimensional ideal over  $\mathbb{K}(X_{i_1}, \dots, X_{i_d})$ .

For zero-dimensional ideals over finite fields, their radicals can be computed by well-known Seidenberg's (1974) method combined with the computation of square-free part of a univariate polynomial (Berlekamp 1970; Davenport 1981; Gianni and Trager 1996; Knuth 1997). However, Seidenberg's method does not work when the coefficient field is imperfect (Becker and Weispfenning 1993, Example 8.16), and in particular cannot be used over  $\mathbb{K}(X_{i_1}, \dots, X_{i_d})$ .

When the coefficient field has positive characteristic and the ideal has positive dimension, there seemed to be no algorithm that can be executed in reasonable time, before the algorithm by Kemper (2002). The difficulty in the zero-dimensional radical computation in positive characteristic was that we have to factorize univariate polynomials over a finitely generated field over a field of positive characteristic,

---

R. Matsumoto

Department of Communications and Integrated Systems, Tokyo Institute of Technology,  
152-8550 Tokyo, Japan  
e-mail: ryutaroh@matsumoto.org

which is computationally hard (see Kemper 2002 for a more detailed description). Kemper (2002) showed how to avoid this difficulty, which enabled the working implementation in the computer algebra system Singular (Greuel et al. 2007).

The author proposed an entirely different method for radical computation (Matsumoto 2001), which is suitable for small positive characteristic slightly before Kemper (2002). The purpose of this short note is to briefly review the author's method, and compare its computational time with the methods implemented in Singular (Greuel et al. 2007). The comparison shows that the computational time of the author's method is much smaller in some cases. Therefore, when the computation of radical does not end in reasonable time with a method based on Kemper's findings, it is worth trying the author's method.

## 2 Another Radical Computation Method for Positive Characteristic

Let  $I$  be an ideal of a polynomial ring  $R$  over a field  $\mathbb{K}$  of characteristic  $p$ . For any given  $a \in \mathbb{K}$ , we assume that we can efficiently compute  $b \in \mathbb{K}$  such that  $b^p = a$ . This condition implies that  $\mathbb{K}$  is perfect, and it is satisfied if  $\mathbb{K}$  is a finite field of characteristic  $p$ . Let  $\varphi$  be the map from  $R$  to itself  $\varphi(f) = f^p$ . Since  $\varphi(f+g) = (f+g)^p = f^p + g^p = \varphi(f) + \varphi(g)$ , the map  $\varphi$  is a ring homomorphism. Let  $I_0 = I$  and  $I_{i+1} = \varphi^{-1}(I_i)$ . Then  $I_i$  is an ideal of  $R$  and  $I_i \subseteq I_{i+1} \subseteq \sqrt{I}$  holds.  $I_i \subsetneq I_{i+1}$  if and only if  $I_i \neq \sqrt{I}$ . Since  $R$  is a Noetherian ring, there exists an integer  $j$  such that  $I_{j-1} \neq I_j = \sqrt{I}$ . The integer  $j$  is the number of computations of  $\varphi^{-1}$  required for the radical computation. We can upper bound  $j$  by  $\lceil n \log_p d \rceil$ , where  $n$  is the number of variables in  $R$  and  $d$  is the maximum of total degrees of generators of  $I$ .

The inverse image  $\varphi^{-1}$  cannot be computed by Buchberger's algorithm (Buchberger 1965, 2006; Mora 2009), because  $\varphi$  moves elements in  $\mathbb{K}$ , which means that  $\varphi$  is not a homomorphism of  $\mathbb{K}$ -algebras. Define maps

$$\varphi_v \left( \sum \alpha_{k,\ell} X_k^\ell \right) = \sum \alpha_{k,\ell} X_k^{\ell p}, \quad \varphi_c \left( \sum \alpha_{k,\ell} X_k^\ell \right) = \sum \alpha_{k,\ell}^p X_k^\ell.$$

Then we have  $\varphi = \varphi_c \circ \varphi_v = \varphi_v \circ \varphi_c$ , and  $\varphi^{-1}(I_i) = \varphi_c^{-1}(\varphi_v^{-1}(I_i))$ . It is well-known that the generators of  $\varphi_v^{-1}(I_i)$  can be computed by Buchberger's algorithm from those of  $I_i$ . We note that the computation of  $\varphi_v^{-1}$  becomes hard as  $p$  increases. By the assumption on  $\mathbb{K}$ ,  $\varphi_c^{-1}$  can be easily computed. In particular, when  $\mathbb{K}$  is the finite field  $\mathbb{F}_{p^m}$  with  $p^m$  elements,

$$\varphi_c^{-1} \left( \sum \alpha_{k,\ell} X_k^\ell \right) = \sum \alpha_{k,\ell}^{p^{m-1}} X_k^\ell.$$

## 3 Comparison of Computational Time and Discussion

In Table 1, we compare the computational time used by the author's method and the “radical” function provided in the computer algebra system Singular (Greuel et al.

2007). Examples are taken from Decker et al. (1999). Characteristic of the field is always 2. In Table 1, “Ex. #” indicates the example number in Decker et al. (1999) pp. 212–217, “Mat” means the time used by the author’s method, “rad” means that by the “radical” function, and “KL” means that by the “radical” function with option “KL”, which instructs the original method in Krick and Logar (1991) is used. The implementation of the author’s method is exactly the same as that listed in Matsumoto (2001). The tests were conducted on Singular version 3.0.3 released on May 2007 running on Linux 2.6.18. The computer had Intel Pentium 4 CPU with 3.2 GHz clock speed, 1 GB of memory, and 2 GB of disk space used for virtual memory. The Singular was allowed to use up to 0.5 GB of memory. This limit on memory was imposed by “ulimit -d” and “ulimit -v” available on the “bash” command interpreter on Linux. When the computation cannot be done within 0.5 GB of memory, “Out” is indicated in Table 1. The numbers in Table 1 show the time used by respective methods in the unit of 1/100 second.

With most examples, the computational times required by the author’s method and the methods provided by Singular do not differ much. However, with example 30, the author’s method can compute the radical with about 35 seconds, while the radical functions in Singular say “out of memory” after 25 minutes of computation. With such examples there is merit to use the author’s algorithm. With example 28, all three methods fail to compute the radical.

Kemper also compared his method and the author’s method in Kemper (2002). The comparison results in Table 1 generally agree to those in Kemper (2002).

**Table 1** Comparison of the time complexity

Ex. #	Mat	Rad	KL	Ex. #	Mat	Rad	KL
1	26	9	7	18	6	5	4
2	3	3	3	19	5	3	3
3	6	6	7	20	21	18	32
4	7	6	12	21	2766	1523	1583
5	2	3	3	22	1	4	3
6	20	11	8	23	12	3	17
7	2	2	1	24	20	14	9
8	8	8	8	25	5	3	1
9	5	4	2	26	2	4	3
10	2	2	1	27	6	3	2
11	3	4	4	28	Out	Out	Out
12	152	24	9	29	32	7	35
13	15	19	17	30	3535	Out	Out
14	8	3	2	31	4	3	3
15	72	67	47	32	63	12	9
16	6	5	11	33	13	9	6
17	14	54	54	34	4	73	52

**Acknowledgements** The author would like to thank helpful comments by Prof. Teo Mora and Prof. Massimiliano Sala that improved the presentation of this note.

## References

- T. Becker and V. Weispfenning, *Gröbner bases*, Graduate Texts in Mathematics, vol. **141**, Springer, Berlin, 1993. A computational approach to commutative algebra, In cooperation with Heinz Kredel.
- E. R. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp. **24** (1970), 713–735.
- B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, Innsbruck, 1965.
- B. Buchberger, *Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal*, J. Symb. Comput. **41** (2006), nos. 3–4, 475–511.
- J. H. Davenport, *On the integration of algebraic functions*, LNCS, vol. **102**, Springer, Berlin, 1981.
- W. Decker, G. M. Greuel, and G. Pfister, *Primary decomposition: algorithms and comparisons*, Algorithmic algebra and number theory (Heidelberg, 1997), Springer, Berlin, 1999, pp. 187–220.
- T. de Jong, *An algorithm for computing the integral closure*, J. Symbolic Comput. **26** (1998), no. 3, 273–277.
- G.-M. Greuel, G. Pfister, and H. Schönemann, *Singular 3.0. a computer algebra system for polynomial computations*, <http://www.singular.uni-kl.de>, 2007, Centre for Computer Algebra, University of Kaiserslautern.
- P. Gianni and B. Trager, *Square-free algorithms in positive characteristic*, AAECC **7** (1996), no. 1, 1–14.
- P. Gianni, B. Trager, and G. Zacharias, *Gröbner bases and primary decomposition of polynomial ideals*, J. Symbolic Comput. **6** (1988), no. 2–3, 149–167.
- G. Kemper, *The calculation of radical ideals in positive characteristic*, J. Symbolic Comput. **34** (2002), no. 3, 229–238.
- D. E. Knuth, *The art of computer programming*, vol. 2: *Seminumerical algorithms*, Third ed., Addison–Wesley, Reading, 1997.
- T. Krick and A. Logar, *An algorithm for the computation of the radical of an ideal in the ring of polynomials*, Proc. of AAECC 1991, LNCS, vol. **539**, Springer, Berlin, 1991, pp. 195–205.
- R. Matsumoto, *Computing the radical of an ideal in positive characteristic*, J. Symbolic Comput. **32** (2001), no. 3, 263–271.
- T. Mora, *Gröbner technology*, this volume, 2009, pp. 11–25.
- A. Seidenberg, *Constructions in algebra*, Trans. Amer. Math. Soc. **197** (1974), 273–313.