Universitext

Springer-Verlag Berlin Heidelberg GmbH

Sebastià Xambó-Descamps

Block Error-Correcting Codes

A Computational Primer

Springer

Sebastià Xambó-Descamps
Universitat Politecnica de Catalunya
C. Pau Gargallo 5
08028 Barcelona
Spain
e-mail: sebastia.xambo@upc.es

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

ISBN 978-3-540-00395-3 ISBN 978-3-642-18997-5 (eBook)
DOI 10.1007/978-3-642-18997-5

Mathematics Subject Classification (2000): 94Bxx, 68P30, 11T71

http://www.springer.de

Originally published by Springer-Verlag Berlin Heidelberg New York in 2003

Cover design: *design & production,* Heidelberg
Typesetting by the author using a TEX macro package
Printed on acid-free paper 40/3142ck-543210

Preface

> There is a unique way to teach: to lead the other person through the same experience with which you learned.
>
> Óscar Villarroya, cognitive scientist.[1]

In this book, the mathematical aspects in our presentation of the basic theory of block error-correcting codes go together, in mutual reinforcement, with computational discussions, implementations and examples of all the relevant concepts, functions and algorithms. We hope that this approach will facilitate the reading and be serviceable to mathematicians, computer scientists and engineers interested in block error-correcting codes.[2]

In its digital form, which is a **pdf** document with hyperlinks, the examples included in the book can be run with just a mouse click. Moreover, the examples can be modified by users and saved to their local facilities for later work. For the convenience of readers, the site

http://www.wiris.com/cc/

has been set up not only to allow a free downloading of the digital version, but also, and primarily, in order to provide direct and free access to the examples, and to the services provided to run them.

The program that handles the computations in the examples, together with the interface that handles the editing (mathematics and text), here will be called **WIRIS/*cc***. More specifically, **WIRIS** stands for the interface, the (remote) computational engine and the associated language, and ***cc*** stands for the extension of **WIRIS** that takes care of the computational aspects that are more specific of error-correcting codes.

WIRIS/*cc* is an important ingredient in our presentation, but we do not presuppose any knowledge of it. As it is very easy to learn and use, it is

[1] Cognitive Research Center at the Universitat Autònoma de Barcelona, interviewed by Lluís Amiguet in "la contra", LA VANGUARDIA, 22/08/2002.

[2] A forerunner of this approach was outlined in [31].

introduced gradually, according to the needs of our presentation. A succint description of the functions used can be found in the Index-Glossary at the end. The Appendix is a summary of the main features of the system.

This book represents the author's response to the problem of teaching a one-semester course in coding theory under the circumstances that will be explained in a moment. Hopefully it will be useful as well for other teachers faced with similar challenges.

The course has been taught at the Facultat de Matemàtiques i Estadística (FME) of the Universitat Politècnica de Catalunya (UPC) in the last few years. Of the sixty sessions of fifty minutes each, nearly half are devoted to problem discussions and problem solving. The students are junior or senior mathematics majors (third and fourth year) and some of them are pursuing a double degree in mathematics and telecommunications engineering.

The nature of the subject, the curriculum at the FME and the context at the UPC advise that, in addition to sound and substantial mathematical concepts and results, a reasonable weight should be given to algorithms and the effective programming of them. In other words, learning should span a wide spectrum ranging from the theoretical framework to aspects that may be labeled as 'experimental' and 'practical'.

All these various boundary conditions, especially the stringent time constraints and the huge size of the subject, are to be pondered very carefully in the design of the course. Given the prerequisites that can be assumed (say linear and polynomial algebra, some knowledge of finite fields, and basic facts about probability theory), it is reasonable to aim at a good understanding of some of the basic algebraic techniques for constructing block error-correcting codes, the corresponding coding and decoding algorithms and a good experimental and practical knowledge of their working. To give a more concrete idea, this amounts to much of the material on block error correcting codes included in, say, chapters 3-6 in [25] (or chapters 4 to 8 in [20]), supplemented with a few topics that are not covered in these books, plus the corresponding algorithmics and programming in the sense explained above.

Of course, this would be impossible unless a few efficiency principles are obeyed. The basic one, some sort of Ockam's razor, is that in the theoretical presentation mathematics is introduced only when needed. A similar principle is applied to all the computational aspects.

The choice of topics is aimed at a meaningful and substantial climax in each section and chapter, and in the book as a whole, rather than at an obviously futile attempt at completeness (in any form), which would be senseless anyway due to the immense size of the fields involved.

WIRIS/*cc* makes it possible to drop many aspects of the current presentations as definitely unnecessary, like the compilations of different sorts of tables (as for example those needed for the hand computations in finite fields).

As a consequence, more time is left to deal with conceptual matters.

It is perhaps also interesting to point out that the examples provided at

http://www.wiris.com/cc/

can be a key tool for the organization of laboratory sessions, both for individual work assignments and for group work during class hours. The system, together with the basic communications facilities that today can be assumed in most universities, makes it easier for students to submit their homework in digital form, and for teachers to test whether the computational implementations run properly.

The basic pedagogical assumptions underlying the whole approach are that the study of the algorithms leads to a better understanding of the mathematics involved, and that the discipline of programming them in an effective way promotes a better understanding of the algorithms. It could be argued that the algorithms and programs are not necessary to solve problems, at least not in principle, but it can also be argued that taking them into account reinforces learning, because it introduces an experimental component in the mathematical practice, and because it better prepares learners for future applications.

Of course, we do not actually know whether these new tools and methods will meet the high expectations of their capacity to strengthen the students' understanding and proficiency. But part of the beauty of it all, at least for the author, stems precisely from the excitement of trying to find out, by experimenting together with our students, how far those tools and methods can advance the teaching and learning of mathematics, algorithms and programs.

> The growing need for mathematicians and computer scientists in industry will lead to an increase in courses in the area of discrete mathematics. One of the most suitable and fascinating is, indeed, coding theory.
>
> J. H. van Lint, [25], p. ix.

Acknowledgements

The character of this book, and especially the tools provided at

http://www.wiris.com/cc/,

would hardly be conceivable without the **WIRIS** system, which would not have been produced without the sustained, industrious and unflappable collaboration of Daniel Marquès and Ramon Eixarch, first in the Project OMEGA at the FME and afterwards, since July 1999, as partners in the firm **Maths for More**.

On the Springer-Verlag side, the production of this book would never have been completed without the help, patience and know-how of Clemens Heine. Several other people at Springer have also been supportive in different phases of the project. My gratitude to all, including the anonymous people that provided English corrections in the final stage.

It is also a pleasant duty to thank the FME and the Departament de Matemàtica Aplicada II (MA2) of the UPC for all the support and encouragement while in charge of the Coding Theory course, and especially the students and colleagues for the everyday give-and-take that certainly has influenced a lot the final form of this text. Joan Bruna, for example, suggested an improvement in the implementation of the Meggitt decoder that has been included in Section 3.4.

Part of the material in this text, and especially the software aspects, were presented and discussed in the EAGER school organized by Mina Teicher and Boris Kunyavski at Eilat (Israel, January 12-16, 2003). I am grateful to the organizers for this opportunity, and to all the participants for their eager inquiring about all aspects of the course. In particular, I have to thank Shmulik Kaplan for suggesting an improvement of the alternant decoder implementation presented in Section 4.3.

Grateful thanks are due to Thomas Hintermann for sharing his enlightening views on several aspects of writing, and especially on English writing. They have surely contributed to improve this work, but of course only the author is to be found responsible for the mistakes and blunders that have gone through undetected.

And thanks to my wife, Elionor Sedó. Without her enduring support and love it would not have been possible to dedicate this work to her on the occasion of our thirtieth wedding anniversary, for it would hardly have been finished.

The author
L'Escala
20/1/03

Contents