

Universitext

Maria Welleda Baldoni • Ciro Ciliberto  
Giulia Maria Piacentini Cattaneo

# **Elementary Number Theory, Cryptography and Codes**

 Springer

Maria Welleda Baldoni  
Ciro Ciliberto  
Giulia Maria Piacentini Cattaneo  
Università di Roma - Tor Vergata  
Dipartimento di Matematica  
Via della Ricerca Scientifica, 1  
00133 Roma  
Italy  
baldoni@mat.uniroma2.it  
cilibert@mat.uniroma2.it  
piacentini@mat.uniroma2.it

*Translator:*  
Daniele A. Gewurz  
Università di Roma "La Sapienza"  
Dipartimento di Matematica  
Ple Aldo Moro, 2  
00185 Roma  
Italy  
gewurz@mat.uniroma1.it

Translation from the Italian language edition:  
*Aritmetica, crittografia e codici* by W.M. Baldoni, C. Ciliberto, and G.M. Piacentini Cattaneo  
Copyright © 2006 Springer-Verlag Italia  
Springer is a part of Springer Science+Business Media  
All Rights Reserved

ISBN 978-3-540-69199-0

e-ISBN 978-3-540-69200-3

Library of Congress Control Number: 2008938959

Mathematics Subject Classification (2000): 11G05, 14G50, 94B05

© 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.

Cover figure from *Balla, Giacomo* © VG Bild-Kunst, Bonn 2008

Cover design: WMX Design GmbH, Heidelberg

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

---

## Introduction

Mathematics, possibly due to its intrinsic abstraction, is considered to be a merely intellectual subject, and therefore extremely remote from everyday human activities. Surprisingly, this idea is sometimes found not only among laymen, but among working mathematicians as well. So much so that mathematicians often talk about *pure mathematics* as opposed to *applied mathematics* and sometimes attribute to the former a questionable birthright.

On the other hand, it has been remarked that those two categories do not exist but, just as we have good and bad literature, or painting, or music, so we have *good* or *bad* mathematics: the former is applicable, even if at first sight this is not apparent, in any number of fields, while the latter is worthless, even within mathematics itself. However, one must recognise the truth in the interesting sentence with which two of our colleagues, experts about applications, begin the preface to the book [47]: *In theory there is no difference between theory and practice. In practice there is.*

We believe that this difference cannot be ascribed to the intrinsic nature of mathematical theories, but to the stance of each single mathematician who creates or uses these theories. For instance, until recently the branch of mathematics regarded as the closest to applications was undoubtedly mathematical analysis and especially the theory of differential equations. The branches of mathematics supposed to be farthest from applications were algebra and number theory. So much so that a mathematician of the calibre of G. H. Hardy claimed in his book [25] the supremacy of number theory, which was to be considered the true *queen* of mathematics, precisely due to its distance from the petty concerns of everyday life. This made mathematics, in his words, “gentle and clean”. A strange opinion indeed, since the first developments of algebra and number theory among the Arabs and the European merchants in the Middle Ages find their motivation exactly in very concrete problems arising in business and accountancy.

Hardy’s opinion, dating back to the 1940s, was based upon a prejudice, then largely shared among scientists. It is quite peculiar that Hardy did not know, or pretended not to know, that A. Turing, whom he knew very well, had

used that very mathematics he considered so detached to break the Enigma code, working for English secret services, dealing a deadly blow to German espionage (cf. [28]). However, the role played by algebra and number theory in military and industrial cryptography is well known from time immemorial. Perhaps Hardy incorrectly believed that the mathematical tools then used in cryptography, though sometimes quite complex, were nevertheless essentially elementary, not more than combinatorial tricks requiring a measure of extemporaneous talent to be devised or cracked, but leading to no solid, important, and enduring theories.

The advances in computer science in the last sixty years have made cryptography a fundamental part of all aspects of contemporary life. More precisely, cryptography studies transmission of data, coded in such a way that authorised receivers only may decode them, and be sure about their provenience, integrity and authenticity. The development of new, non-classical cryptographic techniques, like public-key cryptography, have promoted and enhanced the applications of this branch of the so-called *discrete mathematics*, which studies, for instance, the enumeration of symbols and objects, the construction of complex structures starting with simpler ones, and so on. Algebra and number theory are essential tools for this branch of mathematics, which is in a natural way suitable for the workings of computers, whose language is intrinsically *discrete* rather than *continuous*, and is essential in the construction of all security systems for data transmission. So, even if we are not completely aware of it, each time we use credit cards, on-line bank accounts or e-mail, we are actually fully using algebra and numbers. But there is more: the same techniques have been applied since the 1940s to the transmission of data on channels where interference is present. This is the subject of the theory of error-correcting codes which, though unwittingly, we use daily in countless ways: for instance when we listen to music recorded on a CD or when surfing the Web.

This textbook originated from the teaching experience of the authors at the University of Rome “Tor Vergata” where, in the past years, they taught this subject to Mathematics, Computer Science, Electronic Engineering and Information Technology students, as well as for the “Scuola di Insegnamento a Distanza”, and at several different levels. They gave courses with a strong algebraic or geometric content, but keeping in mind the algorithmic and constructive aspects of the theories and the applications we have been mentioning.

The point of view of this textbook is to be *friendly* and *elementary*. Let us try to explain what we mean by these terms.

By *friendly* we mean our attempt to always give motivations of the theoretical results we show to the reader, by means of examples we consider to be simple, meaningful, sometimes entertaining, and useful for the applications. Indeed, starting from the examples, we have expounded the general methods of resolution of problems that only apparently look different in form, setting and language. With this in mind, we have aimed to a simple and colloquial

style, while never losing sight of the formal rigour required in a mathematical treatise.

By *elementary* we mean that we assume our readers to have a quite limited background in basic mathematical knowledge. As a rule of the thumb, a student having followed a good first semester in Mathematics, Physics, Computer Science or Engineering may confidently venture through this book. However, we have tried to make the treatment as self-contained as possible regarding the elements of algebra and number theory needed in cryptography and coding theory applications. *Elementary*, however, does not mean *easy*: we introduced quite advanced concepts, but did so gradually and always trying to accompany the reader, without assuming previous advanced knowledge.

The starting point of this book is the well-known set of integer numbers and their *arithmetic*, that is the study of the operations of addition e multiplication. Chapter 1 aims to make the reader familiar with integer numbers. Here mathematical induction and recursion are covered, giving applications to several concrete problems, such as the analysis of dynamics of populations with assigned reproduction rules, the computation of numbers of moves in several games, and so on. The next topics are divisions, the greatest common divisor and how to compute it using the well-known Euclidean algorithm, the resolution of Diophantine equations, and numeral systems in different bases. These basic notions are first presented in an elementary way and then a more general theoretical approach is given, by introducing the concept of Euclidean ring. The last part of the chapter is devoted to continued fractions.

One of the goals of Chapter 1 is to show how, in order to solve concrete problems using mathematical methods, the first step is to build a *mathematical model* that allows a translation into one or more mathematical problems. The next step is the determination of suitable *algorithms*, that is procedures consisting of a finite sequence of *elementary operations* yielding the solution to the mathematical problems describing the initial question. In Chapter 2 we discuss the fundamental concept of *computational complexity* of an algorithm, which basically counts the elementary operations an algorithm consists of, thus evaluating the time needed to execute it. The importance of this concept is manifest: among the algorithms we have to distinguish the feasible ones, that is those executable in a sufficiently short time, and the unfeasible ones, due to the time needed for their execution being too long independently of the computing device used. The algorithms of the first kind are the *polynomial* ones, while among those of the second kind there are, for instance, the *exponential* ones. We proceed then to calculate the complexity of some fundamental algorithms used to perform elementary operations with integer numbers.

In Chapter 3 we introduce the concept of congruence, which allows the passage from the infinite set of integer numbers to the finite set of residue classes. This passage from infinite to finite enables us to implement the elementary operations on integers in computer programming: a computer, in fact, can work on a finite number of data only.

Chapter 4 is devoted to the fundamental problem of factoring integer numbers. So we discuss prime numbers, which are the building blocks of the structure of integer numbers, in the sense that each integer number may be represented as a product of prime numbers: this is the so-called *factorisation* of an integer number. Factoring an integer number is an apparently harmless problem from a theoretical viewpoint: the factorisation exists, it is essentially unique, and it can be found by the famous *sieve of Eratosthenes*. We show, however, the unfeasibility of this exponential algorithm. For instance, in 1979 it has been proved that the number  $2^{44497} - 1$ , having 13395 decimal digits, is prime: by using the sieve of Eratosthenes, it would take a computer executing one million multiplications per second about  $10^{6684}$  years to get this result! The modern public-key cryptography, covered in Chapter 7, basically relies on the difficulty of factoring an integer number. In Chapter 4 elements of the general theory of factorial rings can also be found, in particular as regards its application to polynomials.

In Chapter 5 finite fields are introduced; they are a generalisation of the rings of residue classes of integers modulo a prime number. Finite fields are fundamental for the applications to cryptography and codes. Here we present their main properties, expounded with several examples. We give an application of finite fields to the resolution of polynomial Diophantine equations. In particular, we prove the law of quadratic reciprocity, the key to solving second degree congruences.

In Chapter 6 most of the theory presented so far is applied to the search for *primality tests*, that is algorithms to determine whether a number is prime or not, and for factorisation methods more sophisticated than the sieve of Eratosthenes; even if they are in general exponential algorithms, just like Eratosthenes', in special situations they may become much more efficient. In particular, we present some primality tests of probabilistic type: they are able to discover in a very short time whether a number has a high probability of being a prime number. Moreover, we give the proof of a recent polynomial primality test due to M. Agrawal, N. Kayal and N. Saxena; its publication has aroused a wide interest among the experts.

Chapter 7 describes the applications to cryptography. Firstly, we describe several classical cryptographic methods, and discuss the general laying out of a cryptographic system and the problem of cryptanalysis, which studies the techniques to break such a system. We introduce next the revolutionary concept of public-key cryptography, on which the transmission of the bulk of confidential information, distinctive of our modern society, relies. We discuss several public-key ciphers, main among them the well-known *RSA* system, whose security relies on the computational difficulty of factoring large numbers, and some of its variants making it possible, for instance, the electronic authentication of signatures. Recently new frontiers for cryptography, especially regarding security, have been opened by the interaction of classical algebra and arithmetic with ideas and concepts originating from algebraic geometry, and especially the study of a class of plane curves known as *elliptic*

*curves*. At the end of the chapter an introduction to these important developments is given.

Chapter 8 presents an introduction to coding theory, already mentioned above. This is a recent branch of mathematics in which sophisticated combinatorial, algebraic and geometric techniques converge, in order to study the mathematical aspects of the problem of transmitting data through noisy channels. In other words, coding theory studies techniques to send data through a channel when we give for granted that some errors will happen during transmission. These techniques enable us to correct the errors that might arise, as well as to quickly encode and decode the data we intend to send.

In Chapter 9 we give a quick glance at the new frontiers offered by *quantum cryptography*, which relies on ideas originating in *quantum mechanics*. This branch of physics makes the creation of a *quantum computer* at least conceivable; if such a computer were actually built, it could execute in polynomial time computations a usual computer would need an exponential time to perform. This would make all present cryptographic systems vulnerable, seriously endangering civil, military, financial security systems. This might result in the collapse of our civilisation, largely based on such systems. On the other hand, by its very nature, the concept of a quantum computer allows the design of absolutely unassailable *quantum cryptographic systems*, even by a quantum computer; furthermore, such systems have the astonishing property of being able to detect if eavesdroppers attempt, even unsuccessfully, to hear in on a restricted communication.

Each chapter is followed by an appendix containing:

- a list of exercises on the theory presented there, with several levels of difficulty; in some of them proofs of supplementary theorems or alternative proofs of theorems already proved in the text are given;
- a list of exercises from a computational viewpoint;
- suggestions for programming exercises.

The most difficult exercises are marked by an asterisk. At the end of the book many of the exercises are solved, especially the hardest theoretical ones.

Some sections of the text may be omitted in a first reading. They are set in a smaller type, and so are the appendices.

We wrote this book having in mind students of Mathematics, Physics, Computer Science, Engineering, as well as researchers who are looking for an introduction, without entering in too many details, to the themes we have quickly described above.

In particular, the book can be useful as a complementary text for first and second year students in Mathematics, Physics or Computer Science taking a course in Algebra or Discrete Mathematics. In Chapters 1, 3, and 4 they will find a concrete approach, with many examples and exercises, to some basic algebraic theories. Chapters 5 and 6, though more advanced, are in our opinion within the reach of a reader of this category.



The text is particularly suitable for a second or third year course giving an introduction to cryptography or to codes. Students of such a course will probably already have been exposed to the contents of Chapters 1, 3, and 4; so teachers can limit themselves to quick references to them, suggesting to the students only to solve some exercises. They can then devote more time to the material from Chapter 5 on, and particularly to Chapter 7, giving more or less space to Chapters 8 and 9.

The bibliography lists texts suggested for further studies in cryptography and codes, useful for more advanced courses.

A first version of this book, titled “Note di matematica discreta”, was published in 2002 by Aracne; we are very grateful to the publishers for their permission for the publication of this book. This edition is widely expanded and modified: the material is presented differently, several new sections and in-depth analysis have been added, a wider selection of solved exercises is offered.

Lastly, we thank Dr Alberto Calabri for supervising the layout of the book and the editing of the text, especially as regards the exercise sections.

Rome,  
August 2008

*M. Welleda Baldoni*  
*Ciro Ciliberto*  
*Giulia Maria Piacentini Cattaneo*

---

## Contents

<b>1</b>	<b>A round-up on numbers</b> . . . . .	<b>1</b>
1.1	Mathematical induction . . . . .	1
1.2	The concept of recursion . . . . .	5
1.2.1	Fibonacci numbers . . . . .	6
1.2.2	Further examples of population dynamics . . . . .	11
1.2.3	The tower of Hanoi: a non-homogeneous linear case . . . . .	13
1.3	The Euclidean algorithm . . . . .	14
1.3.1	Division . . . . .	14
1.3.2	The greatest common divisor . . . . .	16
1.3.3	Bézout's identity . . . . .	17
1.3.4	Linear Diophantine equations . . . . .	20
1.3.5	Euclidean rings . . . . .	21
1.3.6	Polynomials . . . . .	23
1.4	Counting in different bases . . . . .	30
1.4.1	Positional notation of numbers . . . . .	30
1.4.2	Base 2 . . . . .	32
1.4.3	The four operations in base 2 . . . . .	33
1.4.4	Integer numbers in an arbitrary base . . . . .	39
1.4.5	Representation of real numbers in an arbitrary base . . . . .	40
1.5	Continued fractions . . . . .	43
1.5.1	Finite simple continued fractions and rational numbers . . . . .	44
1.5.2	Infinite simple continued fractions and irrational numbers . . . . .	48
1.5.3	Periodic continued fractions . . . . .	56
1.5.4	A geometrical model for continued fractions . . . . .	57
1.5.5	The approximation of irrational numbers by convergents . . . . .	58
1.5.6	Continued fractions and Diophantine equations . . . . .	61
	Appendix to Chapter 1 . . . . .	62
A1	Theoretical exercises . . . . .	62
B1	Computational exercises . . . . .	73
C1	Programming exercises . . . . .	84

<b>2</b>	<b>Computational complexity</b> .....	87
2.1	The idea of computational complexity .....	87
2.2	The symbol $\mathcal{O}$ .....	89
2.3	Polynomial time, exponential time .....	92
2.4	Complexity of elementary operations .....	95
2.5	Algorithms and complexity .....	97
2.5.1	Complexity of the Euclidean algorithm .....	98
2.5.2	From binary to decimal representation: complexity .....	101
2.5.3	Complexity of operations on polynomials .....	101
2.5.4	A more efficient multiplication algorithm .....	103
2.5.5	The Ruffini–Horner method .....	105
	Appendix to Chapter 2 .....	107
A2	Theoretical exercises .....	107
B2	Computational exercises .....	109
C2	Programming exercises .....	113
<b>3</b>	<b>From infinite to finite</b> .....	115
3.1	Congruence: fundamental properties .....	115
3.2	Elementary applications of congruence .....	120
3.2.1	Casting out nines .....	120
3.2.2	Tests of divisibility .....	121
3.3	Linear congruences .....	122
3.3.1	Powers modulo $n$ .....	126
3.4	The Chinese remainder theorem .....	128
3.5	Examples .....	133
3.5.1	Perpetual calendar .....	133
3.5.2	Round-robin tournaments .....	136
	Appendix to Chapter 3 .....	136
A3	Theoretical exercises .....	136
B3	Computational exercises .....	140
C3	Programming exercises .....	147
<b>4</b>	<b>Finite is not enough: factoring integers</b> .....	149
4.1	Prime numbers .....	149
4.1.1	The Fundamental Theorem of Arithmetic .....	150
4.1.2	The distribution of prime numbers .....	152
4.1.3	The sieve of Eratosthenes .....	157
4.2	Prime numbers and congruences .....	160
4.2.1	How to compute Euler function .....	160
4.2.2	Fermat’s little theorem .....	162
4.2.3	Wilson’s theorem .....	165
4.3	Representation of rational numbers in an arbitrary base .....	166
4.4	Fermat primes, Mersenne primes and perfect numbers .....	168
4.4.1	Factorisation of integers of the form $b^n \pm 1$ .....	168
4.4.2	Fermat primes .....	170

4.4.3	Mersenne primes	172
4.4.4	Perfect numbers	173
4.5	Factorisation in an integral domain	173
4.5.1	Prime and irreducible elements in a ring	174
4.5.2	Factorial domains	175
4.5.3	Noetherian rings	177
4.5.4	Factorisation of polynomials over a field	179
4.5.5	Factorisation of polynomials over a factorial ring	182
4.5.6	Polynomials with rational or integer coefficients	188
4.6	Lagrange interpolation and its applications	191
4.7	Kronecker's factorisation method	195
	Appendix to Chapter 4	198
A4	Theoretical exercises	198
B4	Computational exercises	204
C4	Programming exercises	211
<b>5</b>	<b>Finite fields and polynomial congruences</b>	<b>213</b>
5.1	Some field theory	213
5.1.1	Field extensions	213
5.1.2	Algebraic extensions	214
5.1.3	Splitting field of a polynomial	217
5.1.4	Roots of unity	218
5.1.5	Algebraic closure	219
5.1.6	Finite fields and their subfields	220
5.1.7	Automorphisms of finite fields	222
5.1.8	Irreducible polynomials over $\mathbb{Z}_p$	222
5.1.9	The field $\mathbb{F}_4$ of order four	224
5.1.10	The field $\mathbb{F}_8$ of order eight	225
5.1.11	The field $\mathbb{F}_{16}$ of order sixteen	226
5.1.12	The field $\mathbb{F}_9$ of order nine	226
5.1.13	About the generators of a finite field	227
5.1.14	Complexity of operations in a finite field	228
5.2	Non-linear polynomial congruences	229
5.2.1	Degree two congruences	234
5.2.2	Quadratic residues	236
5.2.3	Legendre symbol and its properties	238
5.2.4	The law of quadratic reciprocity	243
5.2.5	The Jacobi symbol	245
5.2.6	An algorithm to compute square roots	248
	Appendix to Chapter 5	251
A5	Theoretical exercises	251
B5	Computational exercises	255
C5	Programming exercises	260

<b>6</b>	<b>Primality and factorisation tests</b> .....	261
6.1	Pseudoprime numbers and probabilistic tests .....	261
6.1.1	Pseudoprime numbers .....	261
6.1.2	Probabilistic tests and deterministic tests .....	263
6.1.3	A first probabilistic primality test .....	263
6.1.4	Carmichael numbers .....	264
6.1.5	Euler pseudoprimes .....	265
6.1.6	The Solovay–Strassen probabilistic primality test .....	268
6.1.7	Strong pseudoprimes .....	268
6.1.8	The Miller–Rabin probabilistic primality test .....	272
6.2	Primitive roots .....	273
6.2.1	Primitive roots and index .....	278
6.2.2	More about the Miller–Rabin test .....	279
6.3	A polynomial deterministic primality test .....	281
6.4	Factorisation methods .....	290
6.4.1	Fermat factorisation method .....	291
6.4.2	Generalisation of Fermat factorisation method .....	292
6.4.3	The method of factor bases .....	294
6.4.4	Factorisation and continued fractions .....	299
6.4.5	The quadratic sieve algorithm .....	300
6.4.6	The $\rho$ method .....	309
6.4.7	Variation of $\rho$ method .....	311
	Appendix to Chapter 6 .....	313
A6	Theoretical exercises .....	313
B6	Computational exercises .....	315
C6	Programming exercises .....	317
<b>7</b>	<b>Secrets... and lies</b> .....	319
7.1	The classic ciphers .....	319
7.1.1	The earliest secret messages in history .....	319
7.2	The analysis of the ciphertext .....	325
7.2.1	Enciphering machines .....	329
7.3	Mathematical setting of a cryptosystem .....	330
7.4	Some classic ciphers based on modular arithmetic .....	334
7.4.1	Affine ciphers .....	336
7.4.2	Matrix or Hill ciphers .....	340
7.5	The basic idea of public key cryptography .....	341
7.5.1	An algorithm to compute discrete logarithms .....	344
7.6	The knapsack problem and its applications to cryptography ...	345
7.6.1	Public key cipher based on the knapsack problem, or Merkle–Hellman cipher .....	348
7.7	The <i>RSA</i> system .....	349
7.7.1	Accessing the <i>RSA</i> system .....	351
7.7.2	Sending a message enciphered with the <i>RSA</i> system ...	352
7.7.3	Deciphering a message enciphered with the <i>RSA</i> system .....	354

7.7.4	Why did it work? . . . . .	356
7.7.5	Authentication of signatures with the <i>RSA</i> system . . . . .	360
7.7.6	A remark about the security of <i>RSA</i> system . . . . .	362
7.8	Variants of <i>RSA</i> system and beyond . . . . .	363
7.8.1	Exchanging private keys . . . . .	363
7.8.2	ElGamal cryptosystem . . . . .	364
7.8.3	Zero-knowledge proof: persuading that a result is known without revealing its content nor its proof . . . . .	365
7.8.4	Historical note . . . . .	366
7.9	Cryptography and elliptic curves . . . . .	366
7.9.1	Cryptography in a group . . . . .	367
7.9.2	Algebraic curves in a numerical affine plane . . . . .	368
7.9.3	Lines and rational curves . . . . .	369
7.9.4	Hyperelliptic curves . . . . .	370
7.9.5	Elliptic curves . . . . .	372
7.9.6	Group law on elliptic curves . . . . .	374
7.9.7	Elliptic curves over $\mathbb{R}$ , $\mathbb{C}$ and $\mathbb{Q}$ . . . . .	380
7.9.8	Elliptic curves over finite fields . . . . .	381
7.9.9	Elliptic curves and cryptography . . . . .	384
7.9.10	Pollard's $p - 1$ factorisation method . . . . .	385
	Appendix to Chapter 7 . . . . .	386
A7	Theoretical exercises . . . . .	386
B7	Computational exercises . . . . .	390
C7	Programming exercises . . . . .	401
<b>8</b>	<b>Transmitting without... fear of errors</b> . . . . .	<b>405</b>
8.1	Birthday greetings . . . . .	406
8.2	Taking photos in space or tossing coins, we end up at codes . . . . .	407
8.3	Error-correcting codes . . . . .	410
8.4	Bounds on the invariants . . . . .	413
8.5	Linear codes . . . . .	419
8.6	Cyclic codes . . . . .	425
8.7	Goppa codes . . . . .	429
	Appendix to Chapter 8 . . . . .	436
A8	Theoretical exercises . . . . .	436
B8	Computational exercises . . . . .	439
C8	Programming exercises . . . . .	443
<b>9</b>	<b>The future is already here: quantum cryptography</b> . . . . .	<b>445</b>
9.1	A first foray into the quantum world: Young's experiment . . . . .	446
9.2	Quantum computers . . . . .	449
9.3	Vernam's cipher . . . . .	451
9.4	A short glossary of quantum mechanics . . . . .	454
9.5	Quantum cryptography . . . . .	460
	Appendix to Chapter 9 . . . . .	467

XVI Contents

A9	Theoretical exercises .....	467
B9	Computational exercises .....	468
C9	Programming exercises .....	469
<b>Solution to selected exercises</b> .....		471
	Exercises of Chapter 1 .....	471
	Exercises of Chapter 2 .....	482
	Exercises of Chapter 3 .....	483
	Exercises of Chapter 4 .....	487
	Exercises of Chapter 5 .....	492
	Exercises of Chapter 6 .....	496
	Exercises of Chapter 7 .....	498
	Exercises of Chapter 8 .....	501
	Exercises of Chapter 9 .....	504
<b>References</b> .....		507
<b>Index</b> .....		511