中国科学院研究生教学丛书

算术代数

ALGORITHMIC ALGEBRA

Bhubaneswar Mishra

科学出版社 Springer-Verlag

影印版

中国科学院研究生教学丛书

算 术 代 数

ALGORITHMIC ALGEBRA

（影印版）

著者 Bhubaneswar Mishra

科学出版社
Springer-Verlag

2001

内 容 简 介

本书是中国科学院推荐的研究生原版教材之一，是近年来出版的计算机代数方面的权威著作. 书中全面介绍了近 20 年来该领域的主要成果，包括 Gröbner 基、Wu-Ritt 特征基、系统式法、实代数几何等。这些成果是计算机与代数几何交叉研究所产生的新成果，不仅对代数的发展有很大的影响，也对代数学算法在机器人、计算机视觉等方面的应用提供了基础。本书可作为数学系及计算机系相关专业研究生的教材。

ISBN: 7-03-008907-3/O·1293
图字：01-2000-2999

科学出版社 出版
北京东黄城根北街 16 号
邮政编码：100717
源海印刷厂 印刷
科学出版社发行 各地新华书店经销
*
2001 年 2 月第 一 版 开本：710×1000 B5
2001 年 2 月第一次印刷 印张：27
印数：1—3 000 字数：510 000
定价：54.00 元
（如有印装质量问题，我社负责调换〈杨中〉）

《中国科学院研究生教学丛书》序

在21世纪曙光初露，中国科技、教育面临重大改革和蓬勃发展之际，《中国科学院研究生教学丛书》——这套凝聚了中国科学院新老科学家、研究生导师们多年心血的研究生教材面世了。相信这套丛书的出版，会在一定程度上缓解研究生教材不足的困难，对提高研究生教育质量起着积极的推动作用。

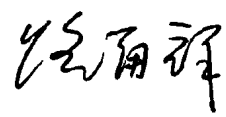
21世纪将是科学技术日新月异，迅猛发展的新世纪，科学技术将成为经济发展的最重要的资源和不竭的动力，成为经济和社会发展的首要推动力量。世界各国之间综合国力的竞争，实质上是科技实力的竞争。而一个国家科技实力的决定因素是它所拥有的科技人才的数量和质量。我国要想在21世纪顺利地实施“科教兴国”和“可持续发展”战略，实现邓小平同志规划的第三步战略目标——把我国建设成中等发达国家，关键在于培养造就一支数量宏大、素质优良、结构合理、有能力参与国际竞争与合作的科技大军。这是摆在我国高等教育面前的一项十分繁重而光荣的战略任务。

中国科学院作为我国自然科学与高新技术的综合研究与发展中心，在建院之初就明确了出成果出人才并举的办院宗旨，长期坚持走科研与教育相结合的道路，发挥了高级科技专家多、科研条件好、科研水平高的优势，结合科研工作，积极培养研究生；在出成果的同时，为国家培养了数以万计的研究生。当前，中国科学院正在按照江泽民同志关于中国科学院要努力建设好“三个基地”的指示，在建设具有国际先进水平的科学研究基地和促进高新技术产

业发展基地的同时，加强研究生教育，努力建设好高级人才培养基地，在肩负起发展我国科学技术及促进高新技术产业发展重任的同时，为国家源源不断地培养输送大批高级科技人才。

质量是研究生教育的生命，全面提高研究生培养质量是当前我国研究生教育的首要任务。研究生教材建设是提高研究生培养质量的一项重要的基础性工作。由于各种原因，目前我国研究生教材的建设滞后于研究生教育的发展。为了改变这种情况，中国科学院组织了一批在科学前沿工作，同时又具有相当教学经验的科学家撰写研究生教材，并以专项资金资助优秀的研究生教材的出版。希望通过数年努力，出版一套面向21世纪科技发展、体现中国科学院特色的高水平的研究生教学丛书。本丛书内容力求具有科学性、系统性和基础性，同时也兼顾前沿性，使阅读者不仅能获得相关学科的比较系统的科学基础知识，也能被引导进入当代科学研究的前沿。这套研究生教学丛书，不仅适合于在校研究生学习使用，也可以作为高校教师和专业研究人员工作和学习的参考书。

“桃李不言，下自成蹊。”我相信，通过中国科学院一批科学家的辛勤耕耘，《中国科学院研究生教学丛书》将成为我国研究生教育园地的一丛鲜花，也将似润物春雨，滋养莘莘学子的心田，把他们引向科学的殿堂，不仅为科学院，也为全国研究生教育的发展作出重要贡献。



To my parents

Purna Chandra & Baidehi Mishra

Preface

In the fall of 1987, I taught a graduate computer science course entitled "Symbolic Computational Algebra" at New York University. A rough set of class-notes grew out of this class and evolved into the following final form at an excruciatingly slow pace over the last five years. This book also benefited from the comments and experience of several people, some of whom used the notes in various computer science and mathematics courses at Carnegie-Mellon, Cornell, Princeton and UC Berkeley.

The book is meant for graduate students with a training in theoretical computer science, who would like to either do research in computational algebra or understand the algorithmic underpinnings of various commercial symbolic computational systems: *Mathematica*, *Maple* or *Axiom*, for instance. Also, it is hoped that other researchers in the robotics, solid modeling, computational geometry and automated theorem proving communities will find it useful as symbolic algebraic techniques have begun to play an important role in these areas.

The main four topics–Gröbner bases, characteristic sets, resultants and semialgebraic sets–were picked to reflect my original motivation. The choice of the topics was partly influenced by the syllabii proposed by the Research Institute for Symbolic Computation in Linz, Austria, and the discussions in Hearn's Report ("Future Directions for Research in Symbolic Computation").

The book is meant to be covered in a one-semester graduate course comprising about fifteen lectures. The book assumes very little background other than what most beginning computer science graduate students have. For these reasons, I have attempted to keep the book self-contained and largely focussed on the very basic materials.

Since 1987, there has been an explosion of new ideas and techniques in all the areas covered here (e.g., better complexity analysis of Gröbner basis algorithms, many new applications, effective Nullstellensatz, multivariate resultants, generalized characteristic polynomial, new stratification algorithms for semialgebraic sets, faster quantifier elimination algorithm for Tarski sentences, etc.). However, none of these new topics could be included here without distracting from my original intention. It is hoped that this book will prepare readers to be able to study these topics on their own.

Also, there have been several new textbooks in the area (by Akritas, Davenport, Siret and Tournier, and Mignotte) and there are a few more on the way (by Eisenbaud, Robbiano, Weispfenning and Becker, Yap, and Zippel). All these books and the current book emphasize different materials, involve different degrees of depth and address different readerships. An instructor, if he or she so desires, may choose to supplement the current book by some of these other books in order to bring in such topics as factorization, number-theoretic or group-theoretic algorithms, integration or differential algebra.

The author is grateful to many of his colleagues at NYU and elsewhere for their support, encouragement, help and advice. Namely, J. Canny, E.M. Clarke, B. Chazelle, M. Davis, H.M. Edwards, A. Frieze, J. Gutierrez, D. Kozen, R. Pollack, D. Scott, J. Spencer and C-K. Yap. I have also benefited from close research collaboration with my colleague C-K. Yap and my graduate students G. Gallo and P. Pedersen. Several students in my class have helped me in transcribing the original notes and in preparing some of the solutions to the exercises: P. Agarwal, G. Gallo, T. Johnson, N. Oliver, P. Pedersen, R. Sundar, M. Teichman and P. Tetali.

I also thank my editors at Springer for their patience and support. During the preparation of this book I had been supported by NSF and ONR and I am gratified by the interest of my program officers: Kamal Abdali and Ralph Wachter.

I would like to express my gratitude to Prof. Bill Wulf for his efforts to perform miracles on my behalf during many of my personal and professional crises. I would also like to thank my colleague Thomas Anantharaman for reminding me of the power of intuition and for his friendship. Thanks are due to Robin Mahapatra for his constant interest.

In the first draft of this manuscript, I had thanked my imaginary wife for keeping my hypothetical sons out of my nonexistent hair. In the interim five years, I have gained a wife Jane and two sons Sam and Tom, necessarily in that order–but, alas, no hair. To them, I owe my deepest gratitude for their understanding.

Last but not least, I thank Dick Aynes without whose unkind help this book would have gone to press some four years ago.

B. Mishra
mishra@nyu.edu.arpa

Contents

Chapter 1

Introduction

1.1 Prologue: Algebra and Algorithms

The birth and growth of both algebra and algorithms are strongly intertwined. The origins of both disciplines are usually traced back to Muhammed ibn-Mūsa al-Khwarizmi al-Quturbulli, who was a prominent figure in the court of Caliph Al-Mamun of the Abassid dynasty in Baghdad (813–833 A.D.). Al-Khwarizmi's contribution to Arabic and thus eventually to Western (i.e., modern) mathematics is manifold: his was one of the first efforts to synthesize Greek axiomatic mathematics with the Hindu algorithmic mathematics. The results were the popularization of Hindu numerals, decimal representation, computation with symbols, etc. His tome "al-Jabr wal-Muqabala," which was eventually translated into Latin by the Englishman Robert of Chester under the title "Dicit Algoritmi," gave rise to the terms algebra (a corruption of "al-Jabr") and algorithm (a corruption of the word "al-Khwarizmi").

However, the two subjects developed at a rather different rate, between two different communities. While the discipline of algorithms remained in its suspended infancy for years, the subject of algebra grew at a prodigious rate, and was soon to dominate most of mathematics.

The formulation of geometry in an algebraic setup was facilitated by the introduction of coordinate geometry by the French mathematician Descartes, and algebra caught the attention of the prominent mathematicians of the era. The late nineteenth century saw the function-theoretic and topological approach of Riemann, the more geometric approach of Brill and Noether, and the purely algebraic approach of Kronecker, Dedekind and Weber. The subject grew richer and deeper, with the work of many illustrious algebraists and algebraic geometers: Newton, Tschirnhausen, Euler, Jacobi, Sylvester, Riemann, Cayley, Kronecker, Dedekind, Noether, Cremona, Bertini, Segre, Castelnuovo, Enriques, Severi, Poincaré, Hurwitz, Macaulay, Hilbert, Weil, Zariski, Hodge, Artin, Chevally, Kodaira, van der

Waerden, Hironaka, Abhyankar, Serre, Grothendieck, Mumford, Griffiths and many others.

But soon algebra also lost its constructive foundation, so prominent in the work of Newton, Tschirnhausen, Kronecker and Sylvester, and thereby its role as a computational tool. For instance, under Bourbaki's influence, it became fashionable to bring into disrepute the beautiful and constructive *elimination theory*, developed over half a century by Sylvester, Kronecker, Mertens, König, Hurwitz and Macaulay. The revival of the field of constructive algebra is a rather recent phenomenon, and owes a good deal to the work of Tarski, Seidenberg, Ritt, Collins, Hironaka, Buchberger, Bishop, Richman and others. The views of a constructive algebraist are closest to the ones we will take in the book. These views were rather succinctly described by Hensel in the preface to Kronecker's lectures on number theory:

> [Kronecker] believed that one could, and that one must, in these parts of mathematics, frame each definition in such a way that one can test in a finite number of steps whether it applies to any given quantity. In the same way, a proof of the existence of a quantity can only be regarded as fully rigorous when it contains a method by which the quantity whose existence is to be proved can actually be found.

The views of constructive algebraists are far from the accepted dogmas of modern mathematics. As Harold M. Edwards [68] put it: "Kronecker's views are so antithetical to the prevailing views that the natural way for most modern mathematicians to describe them is to use the word 'heresy'."

Now turning to the science of algorithms, we see that although for many centuries there was much interest in mechanizing the computation process, in the absence of a practical computer, there was no incentive to study general-purpose algorithms. In the 1670's, Gottfried Leibnitz invented his so-called "Leibnitz Wheel," which could add, subtract, multiply and divide. On the subject of mechanization of computation, Leibnitz said ([192], pp. 180–181):

> And now that we may give final praise to the machine we may say that it will be desirable to all who are engaged in computations...managers of financial affairs, merchants, surveyors, geographers, navigators, astronomers....But limiting ourselves to scientific uses, the old geometric and astronomic tables could be corrected and new ones constructed....Also, the astronomers surely will not have to continue to exercise the patience which is required for computation....For it is unworthy of excellent men to lose hours like slaves in the labor of computation.

Leibnitz also sought a *characteristica generalis*, a symbolic language, to be used in the translation of mathematical methods and statements into *algorithms* and *formulas*. Many of Leibnitz's other ideas, namely, the binary number system, *calculus ratiocanator* or calculus of reason, and *lingua characteristica*, a universal language for mathematical discourse, were to

influence modern-day computers, computation and logical reasoning. The basic notions in *calculus ratiocanator* led to Boolean algebra, which, in turn, formed the foundations for logic design, as developed by C. Shannon.

However, the technology of the time was inadequate for devising a practical computer. The best computational device Leibnitz could foresee was a "learned committee" sitting around a table and saying:

> "*Lasst uns rechnen!*"

In the nineteenth century, Charles Babbage conceived (but never constructed) a powerful calculating machine, which he called an *analytical engine*. The proposed machine was to be an all-purpose automatic device, capable of handling problems in algebra and mathematical analysis; in fact, of its power, Babbage said that "it could do everything but compose country dances." [102]

Except for these developments and a few others of similar nature, the science of computation and algorithms remained mostly neglected in the last century. In this century, essentially two events breathed life into these subjects: One was the study concerning the foundations of mathematics, as established in "Hilbert's program," and this effort resulted in Gödel's incompleteness theorems, various computational models put forth by Church, Turing, Markov and Post, the interrelatedness of these models, the existence of a "universal" machine and the problem of computability (the *Entsheidungsproblem*). The other event was the advent of modern high-speed digital computers in the postwar period. During the Second World War, the feasibility of a large-scale computing machine was demonstrated by Colossus in the U.K. (under M.H.A. Newman) and the ENIAC in the U.S.A. (under von Neumann, Eckert and Mauchly). After the war, a large number of more and more powerful digital computers were developed, starting with the design of EDVAC in the U.S.A. and Pilot ACE and DEDUCE in the U.K.

Initially, the problems handled by these machines were purely numerical in nature, but soon it was realized that these computers could manipulate and compute with purely symbolic objects. It is amusing to observe that this had not escaped one of the earliest "computer scientists," Lady Ada Augusta, Countess Lovelace. She wrote [102], while describing the capabilities of Babbage's analytical engine,

> Many persons who are not conversant with mathematical studies imagine that because the business of [Babbage's analytical engine] is to give its results in numerical notation, the nature of its process must consequently be arithmetical rather than algebraic and analytical. This is an error. The engine can arrange and combine its numerical quantities exactly as if they were letters or any other general symbols; and, in fact, it might bring out its results in algebraic notation were provisions made accordingly.

The next major step was the creation of general-purpose programming languages in various forms: as instructions, introduced by Post; as productions, independently introduced by Chomsky and Backus; and as functions, as introduced by Church in λ-calculus. This was quickly followed by the development of more powerful list processing languages by Newell and Simon of Carnegie-Mellon University, and later the language LISP by McCarthy at M.I.T. The language LISP played a key role in the rapid development of the subjects of artificial intelligence (AI) and symbolic mathematical computation. In 1953, some of the very first symbolic computational systems were developed by Nolan of M.I.T. and Kahrimanian of Temple University.

In parallel, the science of design and complexity analysis of discrete combinatorial algorithms has grown at an unprecedented rate in the last three decades, influenced by the works of Dijkstra, Knuth, Scott, Floyd, Hoare, Minsky, Rabin, Cook, Hopcroft, Karp, Tarjan, Hartmanis, Stern, Davis, Schwartz, Pippenger, Blum, Aho, Ullman, Yao and others. Other areas such as computational geometry, computational number theory, etc. have emerged in recent times, and have enriched the subject of algorithms. The field of computational algebra and algebraic geometry is a relative newcomer, but holds the promise of adding a new dimension to the subject of algorithms.

After a millennium, it appears that the subjects of algorithms and algebra may finally converge and coexist in a fruitful symbiosis. We conclude this section with the following quote from Edwards [68]:

> I believe that Kronecker's best hope of survival comes from a different tendency in the mathematics of our day...., namely, the tendency, fostered by the advent of computers, toward algorithmic thinking.... One has to ask oneself which examples can be tested on a computer, a question which forces one to consider concrete algorithms and to try to make them efficient. Because of this and because algorithms have real-life applications of considerable importance, the development of algorithms has become a respectable topic in its own right.

1.2 Motivations

What happened to Hilbert's man in the street?
—Shreeram S. Abhyankar

There are essentially four groups of people, who have been instrumental in the rapid growth of the subject of "algorithmic algebra." Although, in some sense, all of the four groups are working toward a common goal, namely, that of developing an algorithmic (read, constructive) foundation for various problems in algebra, their motivations differ slightly from one another. The distinction is, however, somewhat artificial, and a considerable overlap among these communities is ultimately unavoidable.