Preface

Systems of polynomial equations are central to mathematics and its application to science and engineering. Their solution sets, called algebraic sets, are studied in algebraic geometry, a mathematical discipline of its own. Algebraic geometry has a rich history, being shaped by different schools. We quote from Hartshorne's introductory textbook (1977):

> "Algebraic geometry has developed in waves, each with its own language and point of view. The late nineteenth century saw the function-theoretic approach of Brill and Noether, and the purely algebraic approach of Kronecker, Dedekind, and Weber. The Italian school followed with Castelnuovo, Enriques, and Severi, culminating in the classification of algebraic surfaces. Then came the twentieth-century "American school" of Chow, Weil, and Zariski, which gave firm algebraic foundations to the Italian intuition. Most recently, Serre and Grothendieck initiated the French school, which has rewritten the foundations of algebraic geometry in terms of schemes and cohomology, and which has an impressive record of solving old problems with new techniques. Each of these schools has introduced new concepts and methods."

As a result of this historical process, modern algebraic geometry provides a multitude of theoretical and highly abstract techniques for the qualitative and quantitative study of algebraic sets, without actually studying their defining equations at the first place.

On the other hand, due to the development of powerful computers and effective computer algebra algorithms at the end of the twentieth century, it is nowadays possible to study explicit examples via their equations in many cases of interest. In this way, algebraic geometry becomes accessible to experiments. The experimental method, which has proven to be highly successful in number theory, now also adds to the toolbox of the algebraic geometer.

As in other areas of pure mathematics, computer algebra may help

- to discover unexpected mathematical evidence, leading to new conjectures or theorems, later proven by traditional means,
- to construct interesting objects and determine their structure (in particular, to find counterexamples to conjectures),
- to verify negative results such as the nonexistence of certain objects with prescribed invariants,
- to verify theorems whose proof is reduced to straightforward but tedious calculations,
- to solve enumerative problems, and
- to create data bases.

There is a growing number of research papers in algebraic geometry originating from explicit computations. The computational methods also play a significant role when it comes to applications of algebraic geometry to practical problems. And, they enter the classroom, allowing us to introduce students at an early stage to algebraic geometry, without developing too much of its abstract machinery.

What are these Notes About ?

These notes are intended to provide a quick start to computing in algebraic geometry. For each topic treated, we include a compact presentation of the background material from commutative algebra and algebraic geometry needed to understand that topic. Further, we discuss the relevant algorithms and explain how to use them in studying algebraic sets. And, we present many explicit computational examples which simultaneously introduce the computer algebra system `SINGULAR` and which may serve as samples for computations carried out by the reader. By revealing implementation details, we point out how to access alternative algorithms for specific tasks. When applying the algorithms to concrete research problems, the difference in their performance could mean to get a result, or to run out of time or memory.

In our presentation, we essentially omit proofs, giving references to standard textbooks instead. Also, at the end of each chapter, we give hints on further reading. These refer to basic definitions and proofs, and to more advanced material as well.

Our main reason for focusing on a single computer algebra system is that we want to keep the size of the notes within reasonable limits. `SINGULAR`, the system of our choice for this purpose, offers a large variety of tools for computations in commutative algebra, algebraic geometry, and singularity theory. We use `SINGULAR 3-0` whose many new features have not yet been described in other textbooks. In fact, some of these features have been implemented by the authors together with other members of the `SINGULAR` team to make the examples presented in these notes work.

The notes originate from an intense one week course given by the authors at Allahabad, India, January 5–11, 2003 and from other schools taught by the authors. The Allahabad course started with an introductory lecture on computer algebra, and it ended with an additional lecture on computing sheaf cohomology and Beilinson monads, given on demand of some of the experienced members of the audience. In between, the authors gave a series of lectures in the morning, and posted practical exercises for the afternoon. The junior and senior participants worked on the exercises in front of the computers, with advice being given by the authors. It is worth pointing out that these practical sessions often ended well after midnight.

Who may Benefit from the Text

The text may accompany students taking a beginner's course in algebraic geometry who might wish to further explore their new playground by experimenting with examples according to their growing knowledge. It can also help Master and PhD students, as well as more experienced researchers, add powerful computational methods to their personal toolbox. Further, it may appeal to users of systems other than `SINGULAR` who might occasionally need tools unavailable in their system of choice. In addition, we address students and researchers interested in implementing their own computational tools using `SINGULAR` as their basis. For these readers we explain, in particular, how to write and debug `SINGULAR` libraries. We believe that the use of this book is not restricted to students and researchers specializing in algebraic geometry itself, but will also prove useful to those in related disciplines.

The Structure of the Text

Although this text widely extends the written material presented at Allahabad, we maintained the original structure of the course, organizing the material as Lectures 0–9, Practical Sessions I–V, and an appendix.

We start in Lecture 0 by giving some remarks on the development of computer algebra. On our way, we present several computer algebra sessions featuring some of the systems which are relevant for researchers in algebraic geometry and related fields.

Lecture 1 is an introduction by historical remarks to the concept of Gröbner bases which is fundamental to computational algebraic geometry. In Lecture 2, we discuss basic computational problems arising from the geometry-algebra dictionary and their solution by means of Gröbner basis methods. Lectures 1 and 2 both already contain explicit `SINGULAR` examples which may serve as samples for those wishing to make their first computational experiments with `SINGULAR`. A thorough introduction to `SINGULAR` is given in Lecture 3. This introduction widely exceeds what is necessary for taking the first steps into `SINGULAR` as it should also serve as a reference for more experienced users. In the Allahabad course, Lecture 3 was divided into two parts, with Practical Session I being held right after the first part, and Practical Session II right after the second part.

Lectures 4 and 5 treat computations in homological algebra, covering basic constructions such as kernels, cokernels, Hom, Ext and Tor, and the more advanced concepts of flatness and Cohen-Macaulay rings. Such computations take center stage in Practical Session III.

In Lecture 6, we discuss some of the methods for exact and symbolic-numerical solving of systems of polynomial equations. These include decomposition techniques for algebraic sets. Primary decomposition is treated separately in Lecture 7 which also deals with normalization. Next follows Practical Session IV.

In Lecture 8, we return to the historical origin of Gröbner bases as presented in Lecture 1, giving an overview on recent algorithms for invariant theory.

Lecture 9 is dedicated to the local study of algebraic sets and, thus, to computations in local rings. For this, we extend the concept of Gröbner bases by introducing standard bases. Corresponding exercises can be found in Practical Session V.

In the appendix, we include the additional lecture on computing sheaf cohomology and Beilinson monads, and we give solutions to the exercises posted in Practical Sessions I–V.

The Level of the Text

Since we address students and researchers, the level of the text is necessarily uneven. Some familiarity with groups, rings, ideals, fields, and vector spaces, together with the information provided in these notes, will enable the reader to understand many of the computations in Lectures 1, 2, 3, 6, 7, and 8. Though we summarize some of the basic concepts of algebraic geometry to provide a common language for all readers, some familiarity with these concepts is needed to fully appreciate our geometric interpretation of the computations. The absolute beginner should, thus, read these notes in conjunction with other books such as Reid's "Undergraduate Algebraic Geometry" (1988) or the undergraduate text "Ideals, Varieties, and Algorithms" by Cox, Little, and O'Shea (1997).

Even in the introductory lectures, the unexperienced reader will find mathematical statements for which he is not prepared. In contrast, the computational recipes arising from the statements are often easy to understand. The reader who is willing to take the recipes for granted will have no problems in applying them to study algebraic sets.

For large parts of Lectures 4, 5, and 9, for the second section of Lecture 8, and for Appendix A, more background in commutative algebra and algebraic geometry is needed.

Exercises

The exercises are designed so as to make the beginner familiar with some basic features of computational algebraic geometry and `SINGULAR`. The serious reader should solve each exercise in front of the computer before turning to the authors' solution of that exercise in the appendix. Further, we highly recommend to check the textbooks by Cox, Little, and O' Shea (1997, 1998), Greuel and Pfister (2002), and Decker and Schreyer (2006) for further exercises admitting a `SINGULAR` solution.

Basic Conventions

If not otherwise mentioned, each **ring** considered in these notes is commutative, and it has a multiplicative identity 1. Ring homomorphisms take 1 to 1. If R is a subring of a ring S, and if I is an ideal of R, we write $I \cdot S$ or simply IS for the ideal generated by I in S. In the context of free resolutions, we often write $N \longleftarrow M$ for a homomorphism $M \longrightarrow N$ since this fits well with how `SINGULAR` displays numerical information on free resolutions.

We work over a field K, referring to the elements of K as **scalars**. We usually write $K[\boldsymbol{x}] = K[x_1, \ldots, x_n]$ for the polynomial ring in n variables over K. A **monomial** in $K[\boldsymbol{x}]$ is a product $\boldsymbol{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$. A **term** in $K[\boldsymbol{x}]$ is a scalar times a monomial.

Timings and `SINGULAR` Output

Occasionally, we print the CPU time used by a `SINGULAR` computation. All timings are given in full seconds, taken on a Pentium IV 2.4 GHZ processor.

In documenting `SINGULAR` sessions, we print `[...]` to indicate that part of the `SINGULAR` output is omitted. Without printing `[...]`, we omit the output displayed by `SINGULAR` when loading a library. This output gives information on the library loaded and on related libraries which are automatically loaded as well.

Website

We maintain a website for this book at

`http://www.singular.uni-kl.de/BOOK_DL/`

We are grateful for comments and corrections which will be posted at the website if appropriate. Also, selected pieces of code written for the book can be downloaded from the website.

Acknowledgment

The Allahabad school was a joint activity of the Harish-Chandra Research Institute at Allahabad and Bhaskaracharya Pratishthana at Pune. We greatly appreciated the facilities provided by the Harish-Chandra Research Institute, and we are grateful to C. S. Inamdar, the Custodian of Bhaskaracharya

Pratishthana. The school was partially supported by the National Board for Higher Mathematics, India. We would like to thank all participants of the Allahabad school and the other schools we taught over the years for their enthusiasm and the fun we had together. Special thanks go to S.D. Adhikari, S.A. Katre, R.A. Rao, D.N. Sheth and J.K. Verma for inviting us to India and for plenty of feedback and help. We are particularly grateful to M. Joshi for guiding us to some of the Allahabad sights, including the confluence of the two rivers, and for patiently answering so many questions on Indian culture. Further, we very much enjoyed a wonderful concert with classical Indian music, featuring, among others, our colleagues M.V. Manohar and A.R. Shastri.

We would like to thank G. Pfister, H. Schönemann, and F.-O. Schreyer for many helpful discussions, G. Lecerf for sharing his ideas on polynomial factorization and for writing the library `absfact.lib`, and A. Frühbis-Krüger, M. Kunte, V. Levandovskyy, T. Markwig, T. von Oertzen, and J. Webber for reading parts of the manuscript and making valuable suggestions. Last but not least, the first author is grateful to the MSRI at Berkeley for its hospitality while working on the final version of the book.

Saarbrücken, Kaiserslautern, October 2005

Wolfram Decker
Christoph Lossen