Foreword

Two ideas lie gleaming on the jeweler's velvet. The first is the calculus, the second, the algorithm. The calculus and the rich body of mathematical analysis to which it gave rise made modern science possible; but it has been the algorithm that has made possible the modern world.

—David Berlinski, *The Advent of the Algorithm*

First there was the concept of integers, then there were symbols for integers: |, ||, |||, ||||, ~~||||~~ (what might be called a sticks and stones representation); I, II, III, IV, V (Roman numerals); 1, 2, 3, 4, 5 (Arabic numerals), etc. Then there were other concepts with symbols for them and algorithms (sometimes) for manipulating the new symbols. Then came collections of mathematical knowledge (tables of mathematical computations, theorems of general results). Soon after algorithms came devices that provided assistance for carrying out computations. Then mathematical knowledge was organized and structured into several related concepts (and symbols): logic, algebra, analysis, topology, algebraic geometry, number theory, combinatorics, etc. This organization and abstraction lead to new algorithms and new fields like universal algebra. But always our symbol systems reflected and influenced our thinking, our concepts, and our algorithms.

In the latter half of the twentieth century a powerful new computational device, the electronic computer, was stirred into the mix. These devices stimulated work on new algorithms, new ways of representing algorithms (more symbol systems) as programs written in various languages devised for use with the new computing devices, even validating (at least for the present) the formal notation of algorithm provided earlier by the logicians (Church's thesis and related concepts). New ways of representing mathematical knowledge in electronic form followed: data bases, electronic documents with new ways of representing mathematical concepts/symbols (TEX, for example) and more recently MathML that promises even more powerful and general ways of electronically storing, manipulating, and displaying mathematical knowledge. There are many other advances along these lines that could be mentioned. But the over arching question is: Where will all this lead?

Will we one day have a giant interacting web of mathematical machines capable of executing all known mathematical algorithms, of finding, retrieving, and organizing mathematical knowledge, of using heuristic procedures to answer mathematical questions when purely algorithmic methods do not exist or are unknown, of verifying mathematical proofs, of generating new conjectures and in some cases proving or finding counterexamples of conjectures (both those generated by machines and those generated by humans)?

In the last century giant strides have been made toward a grand vision of computational mathematics, in the broadest sense of this phrase. An important aspect of this progress has been the research on the application of computers to (mostly) exact mathematical computation—a field that we now call *computer algebra*—a field that brings together Berlinski's role of the calculus and the algorithm in a far-reaching manner. Important new data structures and algorithms, incredibly fast computational processors, large memories, new software systems, and applications to a myriad of problems in a multitude of disciplines have been the building blocks used by the computer algebraists in this fascinating, productive, and ever-widening journey.

This volume, this *Computer Algebra Handbook*, is an impressive snapshot taken with a wide-angle lens of the state of computer algebra research and applications in the last decade of the twentieth century. As with all snapshots of dynamic objects it can reveal a moment in time that reflects only a part of the past and give some indication of the future. But this volume, with the wide-angle lens of two hundred authors, gives a picture of the field that should be valuable to all current researchers and to generations of future researchers. So, gentle reader, I recommend this volume and all its concepts, symbols, and algorithms to you.

Bob Caviness
Newark, Delaware
(October 2001)

Editorial Remarks

The German special interest group for computer algebra (in German: *Fachgruppe für Computeralgebra*) in 1993 published a report "Computeralgebra in Deutschland Bestandsaufnahme, Möglichkeiten, Perspektiven," (in English "Computer Algebra in Germany Status, Possibilities, Perspectives"). The book was written in the German language and contained contributions mostly from German persons working in computer algebra. In 1995 Grabmeier and Weispfenning, who had edited the German report, approached Kaltofen with the idea to produce a translated and expanded version of their book in the English language. The "Computer Algebra Handbook" is the product of that effort. The outline of the original Report was retained, but the Handbook became about twice as long. The pre-dominantly German authorship of the contributions in the Handbook remains as a legacy from the German Report.

The Handbook describes computer algebra in three main chapters: Chapter 2 surveys the methodology of our discipline, Chapter 3 applications and Chapter 4 software. Chapter 1 attempts a definition of what computer algebra is, and Chapter 5 rounds out the Handbook with lists of activities and publications. The Handbook has over two hundred contributing authors, whose efforts and patience we thankfully acknowledge. Consequently, the individual sections are quite diverse in style and focus. One goal of our editorial work was to preserve this diversity. A reader can obtain a homogeneous presentation of computer algebra from any of the excellent books on the subjects cited in Chapter 5 that have been written by small author teams.

The Handbook is to serve as a reference to computer algebra. The bibliography is accordingly large, with over 2,100 entries. However, it is simply impossible to account for all important work in our field of considerable breadth, quantity, and active research. Therefore, the omission of a reference to one's work or software should in no way be taken as a value judgement. We intend to maintain a current bibliography on the web and invite everyone to send us reference to their omitted or future work.

The Handbook contains descriptions of software that may no longer be actively used. We believe it an important archival task to maintain a record of such systems. In the fast-moving world of computer software, significant ideas may simply fall by the wayside as program source code is removed from the storage media. In the attached compact disk we attempt to stem the tide by providing data files that may contain such information. This is just a beginning, and we believe that publishers of journals and proceedings in the discipline ought to follow our example. Like in physics, the experiments with our software must be repeatable.

As the great Donald Knuth has told us, the three tasks: (i) publishing a paper, (ii) publishing a book, (iii) writing a large computer program, are of increasing level of difficulty. Perhaps before and after (iii) one can place the tasks of writing a book and a program with 200 authors. Grants from the National Science Foundation (USA) have supported Kaltofen in part during his work on this book, which is gratefully acknowledged. We thank our colleagues Friedrich W. Hehl,

Cologne for editing section 3.1 Applications in Physics, Wolfram Koepf, Kassel for editing section 3.6 Computer Algebra in Education and Werner M. Seiler, Mannheim for editing section 2.11 Symbolic Methods for Differential Equations. Our work was assisted by several individuals who we would like to name. Markus Hitz, now at North Georgia College and State University, translated Chapter 2 of the German Report into English, layed out the production directory structure that we followed, and performed several other tasks. Ilias Kotsireas, now at the University of Western Ontario, helped us in processing the email of the final updates. And last but not least, our sincere thanks go to Martin Peters, our editor at Springer Verlag, and his staff members Ruth Allewelt and Claudia Kehl, who gave their enthusiastic support and showed great patience with our decidedly slow progress.

Now, we wish the reader enjoyment with the contents of our book.

Johannes Grabmeier
Erich Kaltofen
Volker Weispfenning
(October 2001)

http://www.springer.com/978-3-540-65466-7

Computer Algebra Handbook
Foundations · Applications · Systems
Grabmeier, J.; Kaltofen, E.; Weispfenning, V. (Eds.)
2003, XX, 637 p. With online files/update., Hardcover
ISBN: 978-3-540-65466-7