Computational Mathematics with `SageMath`, by Paul Zimmermann, Alexandre Casamayou, Nathann Cohen, Guillaume Connan, Thierry Dumont, Laurent Fousse, François Maltey, Matthias Meulien, Marc Mezzarobba, Clément Pernet, Nicolas M. Thiéry, Erik Bray, John Cremona, Marcelo Forets, Alexandru Ghitza, and Hugh Thomas, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pennsylvania, 2018, xiv+464 pp., ISBN 978-1-611975-45-1, US$69.00

It is no exaggeration to say that in the past 40 years computer algebra systems (CAS) have changed the everyday practice of mathematics immensely. In general terms, CAS all deal with mathematical objects represented in symbolic form. Some, but not all, also provide interfaces to routines for numerical computation; some, but again not all, provide interfaces to graphical routines for visualization of mathematical objects. Almost all of them provide programming languages to allow users to extend the packages' functionality. All of them aim, in one way or another, to be "systems for doing mathematics by computer" as Stephen Wolfram's Mathematica package [15] boldly proclaims.

1. Historical development of CAS

The following discussion is meant to illustrate the range of such systems that have been created. It does not aim to be exhaustive, and the author apologizes in advance to anyone whose contributions have been omitted or overlooked.

The first computer algebra systems in effect grew out of artificial intelligence research starting in the 1960s, but connections with that side of computer science have lessened over time. First-generation systems were focused primarily on basic symbolic computation including expansion and simplification of expressions, symbolic differentiation and indefinite integration of functions, symbolic solutions of differential equations, and related operations. Among the earliest of the first-generation systems was Macsyma, by Project MAC at MIT, which also included well-known computer networking and artificial intelligence research groups. Macsyma became a commercial product in 1982, but the last version of MIT Macsyma was made available to academic and government users and this version developed into the open source Maxima system. Other well-known first-generation systems were REDUCE, developed by Anthony Hearn, and the Axiom system developed by Richard Jenks. All of these remain in use to this day.

The commercial potential of CAS came to the fore with the second generation starting in the 1980s. The muMATH system, developed by Albert Rich and David Stoutemyer of Soft Warehouse was the first such system designed to run on personal computers. Derive, a system developed by the same company, was a successor to muMATH. Maple [11], developed at the University of Waterloo, and Mathematica are among the best-known commercial CAS. The Symbolic Math Toolbox for MATLAB [12], incorporating the earlier MuPad system (developed at the University of

Paderborn in Germany) also falls within this group. These packages generally provide more extensive numerical and graphical capabilities addressing the needs of a wide range of users in research, industry, and education.

A number of other more specialized systems were also developed during roughly the same period as the second generation of general CAS. They usually grew out of the research efforts of mathematicians in areas where algorithmic and computational approaches were becoming more important. These systems generally do not attempt to be all things to all users and typically do not include extensive numerical facilities or graphics.

One such focus area was research in commutative algebra, algebraic geometry, and their applications. There, computations founded on Bruno Buchberger's algorithm for Gröbner bases, plus new techniques for multivariable polynomial resultants, implemented on the powerful computers becoming widely available through the 1980s, created something of a revolution. One of the first widely used systems was the original Macaulay program developed by David Bayer and Michael Stillman. Its successor, Macaulay2 [9], developed by Daniel Grayson and Stillman, followed. Other programs with related functionality include CoCoA [1], developed by a group at the University of Genoa in Italy, and Singular [5], developed by a group at the University of Kaiserslautern in Germany. The more recent Bertini system [2] takes an alternate numerical approach to algebraic geometry.

Other areas where the influence of computational methods have been especially strong include number theory, group theory, and parts of combinatorics. The PARI/GP system [13] is aimed primarily at applications in number theory and was originally developed by a team led by Henri Cohen. The GAP system [7] was originally developed at the Rheinisch-Westfällische Technische Hochschule in Aachen in Germany and focuses on applications in group theory and combinatorics.

Another extremely powerful package that incorporates features of both of the previous groups of systems is Magma [4], developed at the University of Sydney in Australia. This system includes features for computation in many sorts of algebraic structures. These include the polynomial rings and modules common in commutative algebra and algebraic geometry, groups, number and function fields, and many related structures.

2. CAS IN MATHEMATICAL RESEARCH

All of these CAS are potentially useful as tools for expert mathematicians. They can be used to automate tedious and lengthy calculations and amplify researchers' abilities to understand the range of behavior of structures. As a result, they have facilitated a concrete and experimental approach to doing mathematics in a number of areas.

In some cases we can even see a direct influence on the development of a research area resulting from the availability of, or even the features of, this sort of software. The Macaulay, Macaulay2, CoCoA, and Singular packages discussed above have had a huge impact on the fields of algebraic geometry and commutative algebra. Here is one example illustrating how even the way CAS have presented the results of computations can provide a language for describing phenomena that then leads to new theoretical results.

Let S be the polynomial ring $S = k[x_1, \ldots, x_n]$ (k a field) with the standard grading. That is, the x_i all have degree 1 and polynomials are considered as sums

of elements of the S_d, the vector subspaces of homogeneous polynomials of degree d (together with the zero polynomial). If M is a graded module over S, write M_d for its graded piece of degree d, and for convenience let $M(-s)$ be the associated module with shifted degrees, so that $M(-s)_d = M_{d-s}$. Since Hilbert's famous 1890 invariant theory paper [10], it has been known that the Hilbert function of M, namely the function

$$h_M(d) = \dim_k M_d,$$

can be computed once one has computed a minimal graded free resolution

$$0 \leftarrow M \leftarrow F_0 \leftarrow F_1 \leftarrow \cdots \leftarrow F_\ell,$$

where each F_i is a graded free S-module. Moreover, the length ℓ of the resolution is at most n, the number of variables. Indeed $h_M(d)$ is the alternating sum

$$h_M(d) = \sum_{i=0}^{\ell} (-1)^i \dim_k (F_i)_d.$$

The graded free modules F_i can be written as

$$F_i = \bigoplus_{j \in \mathbb{Z}} S(-j)^{\beta_{i,j}},$$

where the integers $\beta_{i,j}$ are known as the *graded Betti numbers* of the resolution. For example, if M is the quotient ring $M = k[x,y]/\langle x^2, xy, y^3 \rangle$, then its minimal free resolution can be written in the form

$$0 \leftarrow M \leftarrow S \xleftarrow{A} S(-2)^2 \oplus S(-3) \xleftarrow{B} S(-3) \oplus S(-4),$$

where the map A comes from the generators of the ideal

$$A = \begin{pmatrix} x^2 & xy & y^3 \end{pmatrix}$$

and the map B comes from the first syzygies on those generators

$$B = \begin{pmatrix} y & 0 \\ -x & y^2 \\ 0 & -x \end{pmatrix}.$$

The shifted gradings are used so that both maps are graded of degree 0. The graded Betti numbers of a resolution are of interest because they give finer numerical invariants than the Hilbert function. For instance, it is not hard to see that $M' = k[x,y]/\langle x^2, y^2 \rangle$ has the same Hilbert function as M above, but different graded Betti numbers.

Since the original Macaulay system mentioned above, it has been possible to compute graded free resolutions by computer (the packages Macaulay2, CoCoA, Singular, and several others mentioned above can all be used for this). Moreover, the original Macaulay package introduced a very useful convention for representing the graded Betti numbers that the authors dubbed the "Betti table". Instead of representing the β_{ij} as entries of a matrix with i being the row number and j being the column number, the Macaulay Betti table used the convention that $\beta_{i,i+j}$ is put in row i column j of the table (this has the effect of reducing the number of rows needed). Thus, for instance, the Betti table of the resolution given above would be

presented as follows.

$$
\begin{array}{rccc}
 & 0 & 1 & 2 \\
\text{total:} & 1 & 3 & 2 \\
0: & 1 & . & . \\
1: & . & 2 & 1 \\
2: & . & 1 & 1 \\
\end{array}
$$

The *total* row gives the total ranks of the free modules in the resolution, and the rows below give the graded Betti numbers as described above.

Apparently motivated at least in part by computing many of these diagrams and also in part by an older conjecture about multiplicities of such modules due to J. Herzog, C. Huneke, and H. Srinivasan, in 2006 M. Boij and J. Söderberg made two rather amazing conjectures in [3] about the cone generated by the Betti tables of resolutions of modules of codimension c with the shortest possible length $\ell = c$ (the Cohen–Macaulay modules) in the corresponding vector space over $\mathbb{Q}$. First, the extremal rays of this cone should come from the Betti diagrams of the so-called *pure resolutions*—resolutions with only one nonzero entry in each column. Second, the Betti diagrams of all Cohen–Macaulay modules are obtained as positive $\mathbb{Q}$-linear combinations of the diagrams of pure resolutions in a very specific and unique way.

The Boij–Söderberg conjectures immediately captured the imagination of many researchers in this area, perhaps partly due to the fact that the Macaulay Betti diagram was such a natural and easily manipulated object to describe resolutions. Those conjectures were eventually proved by Eisenbud and Schreyer in their 2009 article [6]; the results were extended by them to non-Cohen–Macaulay modules, and they provided a duality theory between Betti tables and cohomology tables of vector bundles over projective space. A whole, very active, area of *Boij–Söderberg theory* has been the result.

3. CAS in mathematics education

Almost from the start of the wide availability of CAS, some mathematics educators have sought to incorporate the use of these systems into mathematics teaching. One interesting and quite successful model for the incorporation of a specific CAS into the teaching of an advanced mathematical subject is the book *A* **Singular** *Introduction to Commutative Algebra* by G.-M. Greuel and G. Pfister [8]. The authors of this book are two of the original developers of the Singular CAS mentioned above. There has also been interest in getting CAS into mathematics courses even at the secondary-school level. The motivations for this are clear enough: Since these tools can be so powerful, it is natural to want to provide students who are learning mathematics with some of that power so that they too can investigate examples that are more interesting and realistic than those found in traditional textbook problems. Moreover, familiarity with these tools and skill at using them well will undoubtedly be valuable if students go on in mathematics or in fields where mathematics is used in a serious way.

However, it is not entirely clear that the mathematical community has figured out, even now, how to incorporate the full power of these tools, or even the power of graphing calculators (some of which now incorporate rudimentary CAS too) into our teaching in a totally beneficial way. In a subject like mathematics, just learning to use a tool should not be the end of the story. To give just one small example, consider the role of curve sketching using information from the first and second

derivatives of a function of one real variable in traditional Calculus 1 classes (a topic very much on the reviewer's mind when writing this review). It is very easy to poke fun at this topic as a useless, outdated exercise, given the fact that a graphing calculator can provide acceptable rough plots and a general-purpose CAS such as Maple, Mathematica, or `SageMath` can draw publication-quality graphs—far better graphs, in fact, than most of us can draw. However, unless serious care is taken to ensure students are getting good intuition in other ways, some understanding of the concepts of critical points and concavity can be lost if students never have to draw a curve based on the signs of the first and second derivatives by hand. The graphs themselves are not ultimately the point in the educational setting; the conceptual understanding of calculus that is derived by drawing the graphs is the point.

4. `SageMath` AND THE BOOK UNDER REVIEW

The `SageMath` system [14] that forms the subject of the book under review is one of the newest CAS, debuting in 2005. Its genesis came from the frustration that its creator William Stein and others felt with the commercial systems such as Maple, Mathematica, MATLAB, and the academic, but still expensive, Magma system. The proprietary nature of basic components of those systems made it problematic for researchers to use them in some cases, since there was essentially no way to check exactly what the lower-level functions were doing. At the same time, the relatively high cost of those systems was a barrier to wider use, especially in education.

Accordingly, from its inception, `SageMath` has been free, open-source software with functionality comparable to Maple and Mathematica. It also rivals Magma for algebraic computation, and it has many features useful for computations in number theory. There is a large community of active users who contribute code and constantly expand the scope of the system. (This is also true of several other packages mentioned above.) The overall design of `SageMath` is also quite different from that of its general-purpose competitors in that `SageMath` effectively serves as a front end to many other established specialty systems, such as GAP, Maxima, Singular, and the R statistical package. These are distributed with the `SageMath` source.

The book under review aims to be much more than simply a reference manual for the `SageMath` system. Part I is the section closest to a manual in that it deals with available user interfaces, the syntax of commands for basic operations from algebra and calculus, `SageMath` data structures and programming (based on the relatively user-friendly Python programming language), graphics, and the object-oriented paradigm that underlies the whole `SageMath` system. However, as the authors say in the preface, "[t]his book provides another approach, by giving a global and synthetic point of view, while insisting on the underlying mathematics, the classes of problems we can solve, and the corresponding algorithms" (p. xiii). Almost all of the examples deal with truly interesting mathematics.

Part II, titled "Algebra and Symbolic Computation", deals with finite fields and number theory, polynomials in one and several variables, and computational methods including Gröbner bases, linear algebra, plus solutions of differential equations and recurrences. Part III deals with numerical computation and covers floating point number systems, numerical solution of equations, numerical linear algebra, and numerical integration. The final Part IV offers brief introductions to the use

of `SageMath` in several areas within combinatorics—enumeration problems, graph theory, and integer programming.

The authors clearly envision, and indeed aim to promote, extensive use of this CAS in school and undergraduate mathematics, saying that "[s]tudents will be able to replace *pen and paper* by *keyboard and screen* while keeping the same intellectual challenge of understanding mathematics" (p. 17). Whether this was successful would also depend on what students were asked to do with `SageMath` and how they were expected to do it.

This book would be an excellent resource for instructors using `SageMath` and a good supplement to other textbooks focused exclusively on the mathematics. However, it would not really be suitable as the sole textbook for a course in any of the areas it discusses since much of the book is addressed to someone who already knows something about the mathematical theory involved and some discussions are quite brief. Nevertheless, there are insightful and thought-provoking comments sprinkled liberally throughout, such as, "We see here a general phenomenon of computer algebra: the best *data structure* to describe a complicated mathematical object (a real number, a sequence, a formal power series, a function, a set) is often an equation defining the object. ... Attempting to find a closed-form solution to this equation is not necessarily of interest ... " (p. 334). This book undoubtedly belongs on the bookshelves of serious users of the `SageMath` system, and it should prove inspiring for teachers looking to use it in their classes.

References

[1] J. Abbott, A. M. Bigatti, and L. Robbiano, *CoCoA: A system for doing Computations in Commutative Algebra*, available at `http://cocoa.dima.unige.it`

[2] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler, *Bertini: Software for Numerical Algebraic Geometry*, available at `bertini.nd.edu` with permanent doi: dx.doi.org/10.7274/R0H41PB5.

[3] M. Boij and J. Söderberg, *Graded Betti numbers of Cohen-Macaulay modules and the multiplicity conjecture*, J. Lond. Math. Soc. (2) **78** (2008), no. 1, 85–106, DOI 10.1112/jlms/jdn013. MR2427053

[4] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265, DOI 10.1006/jsco.1996.0125. Computational algebra and number theory (London, 1993). MR1484478

[5] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, SINGULAR 4-1-2—*A computer algebra system for polynomial computations*, 2019, available at `http://www.singular.uni-kl.de`.

[6] D. Eisenbud and F.-O. Schreyer, *Betti numbers of graded modules and cohomology of vector bundles*, J. Amer. Math. Soc. **22** (2009), no. 3, 859–888, DOI 10.1090/S0894-0347-08-00620-6. MR2505303

[7] The GAP Group, *GAP—Groups, Algorithms, and Programming, Version 4.10.2*; 2019, available at `https://www.gap-system.org`.

[8] G.-M. Greuel and G. Pfister, *A **Singular** introduction to commutative algebra*, Second, extended edition, Springer, Berlin, 2008. With contributions by Olaf Bachmann, Christoph Lossen and Hans Schönemann; With 1 CD-ROM (Windows, Macintosh and UNIX). MR2363237

[9] D. Grayson and M. Stillman, *Macaulay2, a software system for research in algebraic geometry*, available at `http://www.math.uiuc.edu/Macaulay2/`.

[10] D. Hilbert, *Ueber die Theorie der algebraischen Formen* (German), Math. Ann. **36** (1890), no. 4, 473–534, DOI 10.1007/BF01208503. MR1510634

[11] Maple 2019. Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario.

[12] MATLAB and Symbolic Mathematics Toolbox Release 2019b, The MathWorks, Inc., Natick, Massachusetts, United States.

[13] The PARI Group, PARI/GP version `2.11.0`, Univ. Bordeaux, 2018, available at `http://pari.math.u-bordeaux.fr/`.

[14] W. A. Stein et al., *SageMath Mathematics Software*, The SageMath Development Team, 2019, available at `https://www.sagemath.org`.

[15] Wolfram Research, Inc., Mathematica, Version 12.0, 2019, Champaign, IL.

JOHN B. LITTLE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
COLLEGE OF THE HOLY CROSS
Email address: `jlittle@holycross.edu`