A framework for evaluating computer algebra systems for mathematics teaching and learning

Alasdair McAndrew
School of Engineering and Science
Victoria University
PO Box 14428
Melbourne, Victoria 8001
Alasdair.McAndrew@vu.edu.au

Abstract

Computer algebra systems: software which can perform algebraic as well as numeric and graphical computations, have become central to mathematics teaching and learning over the past few decades. However, there is little consensus as to what constitutes the best system for the purpose, or by what standards such a system should be judged. Part of the difficulty is that there are many competing systems, from the commercial (and very expensive), to open source systems, as well as specialized systems and handheld systems. Deciding which system is "best" is not simply a matter of lining the systems up and deciding which has the greatest mathematical power, but rather which system best fits the needs of the students and the teachers. This paper investigates this problem, and describes a framework—in a sense a decision making process—to help mathematics educators make such a decision.

1 Introduction

The modern Computer Algebra System (CAS) can be traced back to 1953, when two masters theses independently investigated systematic differentiation by computer [10]. Since then many systems have been developed, some general purpose; others designed for a particular research interest, for example to perform computations with large polynomials efficiently [16]. Grabmeier, Kaltofen, and Weispfenning [13] give an excellent snapshot of the systems available at that time, some of which, such as MACSYMA and Derive, no longer exist, at least under those names, and others, such as the handheld TI-Nspire or Casio ClassPad, or the system Sage, did not exist at that time. The development and use of CAS's remains a highly active area of research and investigation. Most modern general purpose CAS's are able to manage almost all undergraduate mathematics: calculus, including integral transforms and differential equations; algebra such as the simplification of complex expressions; arithmetic with integers of arbitrary length, or with real numbers of arbitrary precision; two and three dimensional graphics with animations; knowledge of transcendental functions including those defined by differential equations or an integral, such as Bessel functions or the error function.

However, although a CAS might be seen as a magical black box which produces answers to any mathematical questions thrown at it, it must be used with caution. For example, a first year student, using the technique of integration by parts, will find that

$$\int x(1-x)^{99}\,dx = -\frac{1}{100}(1-x)^{100} + \frac{1}{101}(1-x)^{101} + C.$$

However, a CAS might identify the integrand as a polynomial, and expand it to produce an antiderivative with 100 terms containing huge integers.

It has long been affirmed that CAS's have had the potential to both revitalize and invigorate the teaching, learning, and practice of mathematics. However, they have been variously applied. Buchberger [5], claims that mathematics educators could be divided into two classes: the purists, who believe that the CAS would destroy mathematical learning and that the only true learning in mathematics was by having the student slog through exercises and proofs by hand, and the populists, who believe that CAS's could be used as black boxes and that all the mathematics accessible by a CAS need no longer be taught. Most teachers seem to work between these extremes, and instead apply what Buchberger [6], calls the "white box/black box principle":

> In the stage where area X is new to the students, the use of a symbolic software system realizing the algorithms of area X as black boxes would be a disaster... Students have to study the area thoroughly, i.e. they should study problems, basic concepts, theorems, proofs, algorithms based on the theorems, examples, hand calculations.
>
> In the stage where area X has been thoroughly studied, when hand calculations for simple examples become routine and hand calculations for complex examples become intractable, students should be allowed and encouraged to use the respective algorithms available in the symbolic software systems.

This approach, according to Buchberger, can be applied "recursively". There is a huge literature now on the use of the CAS for mathematics education at both tertiary and secondary levels. Much of this literature has shown how the use of a CAS can increase student engagement, and deepen the learning, of a particular mathematical topic. For example Broline et al. [4] show how a CAS can be used to introduce some advanced algebra topics; Naidoo and Naidoo [19] discuss elementary differential calculus, including rates of change and tangents; Powers, Allison, and Grassl [20] investigate combinatorics, including summations and binomial coefficients. In fact, for almost every area of undergraduate mathematics, there seems to be a paper discussing how a CAS can be used to facilitate its learning. More recent research has investigated the best ways on introducing a CAS into a mathematics curriculum, and of ways to best facilitate student learning using CAS. Abdullah [1] reported a study of secondary school students learning the algebraic topic "completing the square", and showed that the students found this topic difficult, boring and time-consuming by hand, and discussed the use of CAS for this topic. Driver [12] looked at the use of a handheld CAS-calculator at secondary school, and showed a statistically significant positive impact on students using them. Güyer [14] shows how carefully written add-on packages with an educational emphasis can guide the students through mathematical processes, Hillman [15] reports on the pilot use of CAS calculators at a secondary school, Kieran and Saldanha [17] study how a CAS can be used to help students master the algebra underlying the equivalence of dissimilar mathematical expressions,

Tokpah [21] provides a meta-analysis of this literature, and concluded that “Results indicated that students exposed to CAS were likely to perform better than their peers taught using non-CAS instruction if both groups were given a common mathematics exam” and showed that the difference in performance was statistically significant.

2 CAS Evaluation and Comparison

Since their inception, CAS’s have been compared according to their performance on various “test suites”. Such a suite would consist of a range of problems from different areas of mathematics such as linear algebra, calculus and analysis, solution of equations, simplification of expressions, numerics, graphics, and the CAS would be tested to see if it could give a correct answer to the problem, a correct answer but with some assistance from the user, or either an incorrect answer or no answer at all. Wester [22] compares various of the systems of the time by this approach. It is clear that this “black box” evaluation technique—problem in, solution out—is not a suitable evaluation for teaching and learning, where the emphasis is not so much on the mathematical capabilities of the system, but the usability of the system to engage the students and enhance their learning. âĄăChonacky and Winch [8, 7] attempt to gauge the usefulness of three commercial systems, and aim to “to present a framework that helps educators make their own critical comparison of Maple, Mathematica, and Matlab as candidate computational productivity tools for use in their instructional programs”. As well as the mathematical power of these systems, they discuss the interfaces, the use of palettes for entering mathematical expressions, price and accessibility, fitness for purpose, and use outside the classroom. They conclude that each system has roughly equivalent power, but differ in their approaches to mathematical problem solving, and so the choice of system âĂIJmust be made locallyâĂİ. They provide a handful of questions to consider when choosing a system, but don’t go so far as to provide a complete rubric, as for example is done by Wrench [23]. Jackson [16] provides a more comprehensive list of questions for software comparison, but his questions are not specifically geared to CAS. A more recent generic rubric for software evaluation is provided by Sage Learning Systems (2001) A particularly elegant and simple rubric is given by Lever-Duffy and McDonald [18, Chapter 14].

3 Models for Evaluation: ISO 9126 and ACTIONS

In this section we investigate two evaluation models. ISO 9126 (ISO, 2001) is an international standard for software evaluation, and considers several characteristics and sub-characteristics. The characteristics are: Functionality (how well the software performs its task); Reliability (fault tolerance and handling of errors); Usability (how easy the software is to learn, user interface); Efficiency (response time and resource management); Maintainability (fault diagnosis, modifiability, testability); Portability (environmental flexibility, ease of installation); and compliance with laws and regulations. A table listing all the characteristics and sub-characteristics is given by Chua and Dyson [9]. The ACTIONS model [2] was developed to evaluate technologies for distance learning; its name is the acronym for Access, Cost, Teaching & Learning, Interactivity and User-friendliness, Organizational issues; Novelty, Speed. Bates points out that because of the huge variability of learning styles, motivation and so on, "teaching and learning is a weak discriminator or selecting and using technologies". The ACTIONS model was developed for a specific purpose, and not just for software, but it is clear that it shares many similarities with ISO 9126. The following table shows the similarities and differences between the models:

ISO 9126	**ACTIONS**
Functionality	
Reliability	
Usability	Interactivity & user-friendliness
Efficiency	Speed
Maintainability	
Portability	Access & Cost
Legal compliance	
	Teaching & learning
	Organizational issues
	Novelty

Note that the equivalences are not quite exact: Portability (in ISO 9126) is not exactly equivalent to Access & Cost in ACTIONS, and Efficiency (in ISO 9126) includes more than merely speed. However, these are close enough for comparison. Bates also proposed another evaluation model similar to ACTIONS but for use with on-campus technology, called SECTIONS [3] in which SE stands for Students, Ease of use, and all the other abbreviations are the same.

ISO9126 has been used in particular to evaluate learning management systems [9, 11].

If we are to evaluate a CAS for use in teaching and learning, then we must consider its functionality: a CAS which does not handle the mathematics within a course would be useless for its purpose. Also, a CAS must support not only the mathematics, but the students who will be using it: a CAS which might be suitable for students studying engineering mathematics may be different for a CAS suitable for students studying teacher education, and these may again be different from a CAS for students studying mathematics as their major discipline.

3.1 A hybrid model

Clearly some sort of hybrid model is needed to evaluate a CAS for teaching and learning; this section investigates a variation of Bates' ACTIONS model. The following need to be considered before using a CAS in a mathematics course:

Functionality, or fitness for purpose. The CAS must be able to manage all mathematics topics on the course. If a particular topic isn't built-in to the software, it should be available as either an add-on (either from the company or a third party supplier), or can be easily programmed by teachers. This requires that the CAS must have a built-in programming language by which it can be extended with user-written functions.

Access. Students learn best when the software becomes part of how they engage with the mathematics. For that reason, a software which is only available in a university computer lab is likely to be less useful than software they can carry with them. In other words, students should be able to access the software when they want to, rather then when their university timetable dictates. The software should also run on multiple platforms with no loss of functionality, and should be portable between systems. For example, students working on PCs running MS Windows in a computer lab should be able to save their work and open it on an Apple computer running OS X.

Cost. This would be a concern for any but the best financially endowed institutions. Site licenses for commercial software can be prohibitively expensive, more so if the institution wants to "sell on" the software at cut rates to the students and their teachers.

Teaching & learning. How will the CAS actually support mathematics learning, and how will it support the learning needs of the students? For example, students with weak backgrounds may well be better served by a simpler, easier system than by a formidable commercial system.

Interactivity and user-friendliness. It is easy for a teacher to be so excited with the software that more time is spent learning how to use the software than actually learning mathematics. The use of the software then should be easy enough that the students become comfortable with it very quickly, and use it confidently to extend and enhance their learning of mathematics. One aspect which shouldn't be overlooked is that of documentation, which should be easily available from within the system, and cover all aspects of its use.

Organizational issues. How will the system fit into the philosophy and practice of the institution? Can it be shoe-horned into a current syllabus, or will there need to be substantial re-writing? And how will the pedagogy of CAS use fit in with the teaching of successor subjects?

Novelty and newness. The system should be the newest of its kind, and be able to impart some level of novelty to the students. A system which is merely used to give answers to questions is unlikely to have much pedagogical value.

Students. There are many different cohorts of students likely to be studying some mathematics, for example students of: engineering, the sciences, education, economics, business, and of course mathematicians in training (almost certainly the smallest group of all). The mathematical needs of these cohorts vary greatly, as will their mathematical needs after graduation. An engineer, for example, may need a system which will allow simulation of a complex system; a school teacher a system which can guide students through their learning. The two are unlikely to be the same.

Considering its initial letters, this model could be called the FACTIONS model for software evaluation.

Although the FACTIONS model has been developed to facilitate the evaluation and choice of computer algebra systems for teaching, this model can be used to evaluate and choose any discipline-specific software. Examples include molecular modelling software for chemists, computer aided drafting systems for engineers, anatomy software for health or medical education.

4 Two case studies

We shall first show how this model might be used for selecting a CAS for use in an engineering mathematics subject. Topics taught include multivariate calculus, complex numbers, linear algebra, power series, differential equations. Possible CAS's are Matlab (with its Symbolic package), CAS calculators, Maple. We shall give each one a score from 0–2, with 0 meaning does not satisfy requirements, 2 meaning satisfies all requirements, and 1 either satisfies only some requirements, or satisfies all requirements but with difficulty. Table 1 shows a possible scoring.

	The possible CAS's		
Characteristic	Matlab[a]	CAS calculator[b]	Maple[c]
Fitness[d]	2	2	2
Access[e]	2	1	0
Cost	2	1	0
Teaching/Learning	1	2[f]	1
Interactivity	2	2	2
Organization	2[g]	1	0
Novelty	2	1	2
Students	2	1	2
Total	15	11	9

[a] Already available
[b] Most students have these, but not all. Also, students own different brands.
[c] This would have to be purchased at an institutional level.
[d] All systems are equally capable for our purposes. .
[e] See notes a, b and c.
[f] Most students will already have had exposure to CAS calculators at their high schools; the other two systems will require some learning.
[g] Of all the systems, Matlab is used in later engineering studies.

Table 1: FACTIONS for engineering mathematics

Note that these scores are, of course, subjective: another scoring may well be different. But the purpose of this system is not to provide an objective outcome, but simply to aid in the decision making process. Applying the model as shown indicates that Matlab would be the best tool, followed by CAS calculators.

For the second example, we consider an advanced undergraduate, or possibly even postgraduate, course in cryptography. This course would include classical and computational number theory, finite fields, elliptic curves and their groups, symmetric and asymmetric cryptosystems,

	The possible CAS's				
Characteristic	Mathematica[a]	Maple[a]	GNU Maxima[b]	Axiom[b]	Sage[b]
Fitness	2	1[c]	2	2	2
Access[d]	0	0	2	2	2
Cost	0	0	2	2	2
Teaching/Learning[e]	2	2	2	2	2
Interactivity	2	2	2	1[f]	2
Organization	0	0	2	2	2
Novelty	2	2	2	2	2
Students[g]	2	2	2	2	2
Total	10	9	16	15	16

[a] Not currently available: would require considerable expense to deploy
[b] All open source, and so can be made available at little costs
[c] At the time of experimentation, Maple's support for finite fields was clumsy
[d] Access depends on cost
[e] Once individual software differences are ironed out, the teaching and learning aspects of each option are very similar
[f] Under any version of MS Windows (the standard institutional OS), Axiom has a very old-fashioned, clunky, DOS-based interface. There is a new web-based interface, which requires Axiom to be installed on a server.
[g] The cohort of students is likely to be fairly homogeneous.

Table 2: FACTIONS for a cryptography course

and applications including cryptographic hash functions and digital signatures. Clearly we need a modern, robust, CAS—a CAS calculator would be totally inadequate here—and we have the choice of Mathematica, Maple, GNU Maxima, Axiom, Sage. Table 2 shows a possible scoring.

We see that for the purpose of our intended course and student cohort, there is not much to choose between GNU Maxima and Sage (and indeed the author has used both with students very successfully). As before, note that this is not designed as a abstract measure of the power and functionality of any given system, but as a means for educators to decide which system may best fit the needs of their teaching, their students, and their institution.

5 Conclusion

Most previous CAS comparisons concentrate entirely on the power of the CAS as a black box for solving problems, or displaying graphics. And so the more problems the CAS can manage (with minimal help), the more powerful it is. But this does not necessarily mean that such a CAS is the best fit for a given course, or cohort of students. A simpler CAS with a friendly interface may be more suitable than a more powerful CAS. Then there are the matters of cost and deployment: how will students have access to the CAS? All of these completing interests must be considered before a decision can be made.

We have demonstrated a model for assessing the suitability of a CAS for teaching purposes, and shown how it is based on, but is different to, previous software metrics. We believe that a tool such as this could be very useful for teachers and administrators in helping choose the most suitable software to support their course, and student learning.

References

[1] L. M. Abdullah. “Procedural Knowledge in the Presence of a Computer Algebra System (CAS): Rating the Drawbacks Using a Multi-factorial Evaluation Approach”. In: *International Journal for Technology in Mathematics Education* 14.1 (2006), pp. 14–21.

[2] A. W. Bates. *Technology, e-Learning and Distance Education*. 2nd. Routledge, 2006.

[3] A. W. Bates and G. Poole. *Effective Teaching with Technology in Higher Education: Foundations for Success*. Jossey-Bass, San Francisco, 2003.

[4] D. Broline et al. “Examples of Using a Computer Algebra System in Teaching Algebra”. In: *Primus* 17.3 (2007), pp. 256–257.

[5] B. Buchberger. “Computer Algebra: The End of Mathematics?” In: *SIGSAM Bulletin* 36.1 (2002), pp. 16–19.

[6] B. Buchberger. “Should Students Learn Integration Rules?” In: *SIGSAM Bulletin* 24.1 (1990), pp. 10–17.

[7] N. Chonacky and D. Winch. “3Ms for Instruction, Part 2: Reviews of Maple, Mathematica, and Matlab”. In: *Computing in Science and Engineering* 7.4 (2005).

[8] N. Chonacky and D. Winch. “3Ms for Instruction: Reviews of Maple, Mathematica, and Matlab”. In: *Computing in Science and Engineering* 7.3 (2005).

[9] Bee Bee Chua and Laurel Evelyn Dyson. “Applying the ISO 9126 model to the evaluation of an elearning system”. In: *Proc. of ASCILITE*. 2004, pp. 5–8.

[10] Wolfram Decker and Christoph Lossen. *Computing in algebraic geometry: A Quick Start Using Singular*. Springer, 2006. Chap. Introductory remarks on computer algebra, pp. 1–9.

[11] Rachida Djouab and Moncef Bari. “An ISO 9126 Based Quality Model for the e-Learning Systems”. In: *International Journal of Information and Education Technology* 6.5 (2016), p. 370.

[12] David Driver. “The impact on student achievement when CAS technology is introduced”. In: *Teaching Mathematics and its Applications* 31.1 (2012), pp. 3–10.

[13] Johannes Grabmeier, Erich Kaltofen, and Volker B. Weispfenning. *Computer Algebra Handbook: Foundations, Applications, Systems*. Springer, 2003.

[14] T. Güyer. “Computer algebra systems as the mathematics teaching tool”. In: *World Applied Sciences Journal* 3.1 (2008), pp. 132–139.

[15] Wendy Hillman. “Calculating Change: How CAS Is Changing the Classroom”. In: *Teacher: The National Education Magazine* July 2005 (2005), p. 54.

[16] Gregg B. Jackson. “How to Evaluate Educational Software and Websites”. In: *Technologies Today—TechKnowLogia* (2000), pp. 57–58.

[17] Carolyn Kieran and Luis Saldanha. “Computer algebra systems (CAS) as a tool for coaxing the emergence of reasoning about equivalence of algebraic expressions”. In: *Proceedings of the 29th Conference of the International Group for the Psychology of Mathematics Education, Melbourne, Australia*. Vol. 3. 2005, pp. 193–200.

[18] J. Lever-Duffy and J. McDonald. *Teaching and Learning with Technology*. 4th edition. Allyn and Bacon, 2010.

[19] K. Naidoo and R. Naidoo. “First year students understanding of elementary concepts in differential calculus in a computer laboratory teaching environment”. In: *Journal of College Teaching & Learning (TLC)* 4.9 (2011).

[20] Robert A. Powers, Dean E. Allison, and Richard M. Grassl. “A Study of the Use of a Handheld Computer Algebra System in Discrete Mathematics.” In: *International Journal for Technology in Mathematics Education* 12.3 (2005), pp. 103–114.

[21] Christopher L. Tokpah. “The Effects of Computer Algebra Systems on Students’ Achievement in Mathematics”. PhD thesis. Kent State University, 2008.

[22] Michael J. Wester. *Computer algebra systems: a practical guide*. John Wiley & Sons, Inc., 1999.

[23] Jason S. Wrench. “Educational software evaluation form: towards a new evaluation of educational software”. In: *The Source* 3.1 (2001), pp. 34–47.