

COMPUTER ALGEBRA BASED ASSESSMENT OF MATHEMATICS ONLINE

Laura Naismith and Christopher J Sangwin

Computer Algebra Based Assessment of Mathematics Online

Laura Naismith
The Centre for Educational Technology and Distance Learning
(CETADL)
The University of Birmingham
Birmingham B15 2TT
L.Naismith@bham.ac.uk

Christopher J Sangwin
LTSN Maths Stats and OR Network
School of Mathematics and Statistics
The University of Birmingham
Birmingham B15 2TT
C.J.Sangwin@bham.ac.uk

Abstract

In this paper, we investigate computer algebra based assessment of mathematics online, with a focus on undergraduate students. Introducing a computer algebra system to assist in marking allows a paradigm shift from teacher-provided answers to student-provided answers. The Computer Algebra Based Learning and Evaluation (CABLE) system is presented as an open source infrastructure for mathematical learning objects. Features of the CABLE system, including the modular design, database structure, learning object specification and learning object contextualisation are described and areas for future work are identified.

Why should we use computer algebra for assessment of mathematics?

Computer algebra systems (CAS) are well-established research tools for performing symbolic manipulation of mathematical expressions. For example, they may symbolically expand an expression such as $(x+1)^2$ to $x^2 + 2x + 1$ and then perform the reverse factorisation process. Fundamental to any CAS is the ability to correctly assess whether two expressions are algebraically equivalent, even if written in different forms. In a teaching and learning context, one expression might be a student's answer and the other a teacher's answer.

Incorporating a CAS into a computer aided assessment (CAA) system allows students' free text responses to be tested objectively. In contrast, multiple

choice questions (MCQs) and other traditional 'objective' questions require students to select from a list of teacher-provided answers. It is entirely possible, and in many circumstances likely, that students manipulate the provided answers, and do not attempt to work out their own answers. Indeed, there is some evidence, for example (Hassmén 1994), that MCQs contain inherent gender bias. By introducing CAS to evaluate students' answers, we may make a paradigm shift from teacher-provided answers to student-provided answers in CAA systems for mathematics.

How can we use computer algebra for assessment?

Checking the algebraic equivalence of two expressions is not restricted to comparing a student's particular response with a correct answer. The CAS can also check the student's answer for equivalence with an incorrect answer derived from a common mistake. Tailored feedback can then be given to explain the student's particular misconception. In traditional objective tests, such as those of (Nichols 2002), each incorrect response is derived from a possible misconception or common mistake. The misconceptions are thus "up front" for a student to select from, while with CAS, they can remain in the background and generate formative feedback if and when a student responds with an incorrect answer.

The CAS may also be used to test for properties of a student's answer. In particular, students can be asked to create an instance of an object that satisfies specific mathematical properties, a type of question believed to promote higher order skills, as in (Dahlberg 1996). Examples of this type of question include 'Give an example of a differentiable function $f(x)$ with a critical point at $x=1$ ' and 'Give an example of a quadratic $p(x)$ with roots at 3 and 7'. Computer algebra facilitates marking by checking the student's answer against each required condition separately, a procedure that could be extremely laborious if done by hand, and providing partial credit and tailored feedback as appropriate. Such questions are discussed in, for example (Sangwin 2003 and 2004).

The CAS also allows for the instantiation of questions with random parameters. When setting a conventional test, it is common for teachers to begin with the answer and manipulate it mathematically to generate a question. Such manipulations may not be trivial, especially when teaching mathematics to undergraduate students, but are precisely the job for which a CAS is designed. By using random parameters, each student can receive a unique set of questions, which may assist in reducing plagiarism and impersonation. Assuming a suitably developed infrastructure within the CAA system, students may request additional distinct, but equivalent, problem sets for further practice if they feel unsure of their competence, thus giving them more control over their learning. As discussed in (Sangwin 2004), the use of random parameters can also present valuable opportunities for group work, wherein collaborative discussions can focus on *methods* rather than on specific *answers*.

Such approaches to CAA are not speculative, and have been used for a number of years with implementations such as AIM, reported in, for example (Klai 2000) and (Strickland 2002). Other approaches are possible, for example (Moore 2001). It is expected that the use of CAS will become increasingly common for mathematical CAA in the near future.

Features of the CABLE system

Based in part on previous work utilising CAS for mathematical CAA, CABLE (Computer Algebra Based Learning and Evaluation) provides an open source, online infrastructure for authoring, testing, storing, presenting and marking lightweight and flexible *mathematical learning objects* together with a relational database structure for their storage and use. This section addresses the design and partial implementation of this system.

It is with great care that the design, of both the database structure and learning objects, has taken place without referencing implementation by a specific CAS or virtual learning environment. It is hoped that others will adopt similar structures, and indeed develop these, as initial work towards the production of interoperability standards for computer algebra marked mathematical learning objects progresses.

CABLE currently combines the Axiom CAS and the LogiCampus VLE. Both components are available open source and are free from licensing fees.

Modular design

Adopting a modular design approach allows the functions that require computer algebra support to be separated from those that manage student and course data. This approach is intended to facilitate future integration with an alternate CAS and/or VLE and also the migration to question and test interoperability and mathematical display (e.g. MathML) standards.

The database structure

The CABLE database holds a number of *objects*. The mathematical learning object, referred to as a *question*, is actually significantly more than a single question, together with a marking scheme. In particular, since randomisation is possible, a student receives not a question but a *question instance*. The system must therefore track which instance is given to a particular student. Furthermore, a student may have more than one *attempt* at a particular instance and may indeed answer more than one instance to gain repeated practice. There must also be some mechanism for collecting questions together into quizzes, or learning materials. It is therefore clear that a relational database structure is necessary to coordinate the system.

The learning objects

The learning object consists of the following four major components.

1. Computations using CAS

Each question contains a number of possible variants, and this component contains a short section of computer algebra system code that is executed at question instantiation to create a list of *local variables*. The values of these variables are available to subsequent parts of the question.

2. Question stem

This creates a string which is displayed to the student containing the statement of the question itself. This may contain the values of local variables defined above. Currently this is authored and stored using simple LaTeX, with syntax for string substitutions.

3. Marking procedure

This is a procedure in the computer algebra system, which evaluates the student's answer. The prototype procedure takes two expressions, one assumed to be the student's and one the teacher's. These are subtracted and the resulting expression simplified as far as is possible. A zero result indicates algebraic equivalence between the student's and teacher's answers. The marking procedure returns marks and feedback, which can subsequently be displayed.

4. Worked solution template

There is also the facility to produce a worked solution. This is analogous to the question stem, and may use the local variables. The worked solution does not depend on a student's answer(s).

Question instance contexts

Each question is presented to a student in what we describe as a *context*. The context essentially consists of a list of global variables upon which the learning object may rely. The list of global variables has default values which may be altered in a number of places, providing a context for the question. For example, the number of marks, details of any penalties for incorrect attempts and even the URL of help files or notes can form part of the context. A cascading system of options will allow maximum flexibility, while imposing consistency over whole collections of questions. This allows one question object to be reused easily in different contexts.

The options can also influence the way a student's answer is evaluated through *answer tests*. Answer tests are predefined CAS procedures that can be employed within the learning object's marking procedure. As a specific example, an accuracy test could take two options as arguments: whether floating point approximations of coefficients are allowed, and if so, the level of accuracy required. These options may be changed to accommodate different groups of students, thereby promoting question reuse.

Future work

The CABLE system is currently in a state of partial implementation. We intended to explore the design issues further over the next few months within a working prototype system.

A key area for further investigation is the ability to assess individual steps in a single learning object, as described in (Beevers 1995). This can facilitate the awarding of partial credit, identified in (Strickland 2002) as a key issue for undergraduate students.

Acknowledgements

This research is funded through the LTSN Maths, Stats and OR Network and a grant by Microsoft UK Limited to The Centre for Educational Technology and Distance Learning (CETADL) at The University of Birmingham, United Kingdom.

References

- Beevers, C. E., McGuire, G. R., Stirling, G. and Wild, D. G.: 1995, Mathematical ability assessed by computer, *Computers & Education* **25**(3), 123-132.
- Dahlberg, R. P. and Housman, R. P.: 1997, Facilitating learning events through example generation, *Educational Studies in Mathematics* **33**, 283-299.
- Hassmén, P. and Hunt, D. P.: 1994, Human self-assessment in multiple choice, *Journal of Educational Measurement* **31**(2), 149-160.
- Klai, S., Kolokolnikov, T. and Van den Bergh, N.: 2000, Using Maple and the web to grade mathematics tests, *Proceedings of the International Workshop on Advanced Learning Technologies, Palmerston North, New Zealand, 4-6 December*.
- Moore, R. C. and Griffin, F.: 2001, Macq_TE_X: self-testing quizzes, using PDF, Preprint.
- Nichols, D. and Greenhow, M.: 2002, Using question mark perception v3 for testing mathematics, *MSOR Connections* **2**(3), 19-22. <http://ltsn.mathstore.ac.uk/newsletter/aug2002/pdf/perception3.pdf> (viewed May 2004).
- Sangwin, C. J.: 2003, [Assessing higher mathematical skills using computer algebra marking through AIM](#), *Proceedings of the Engineering Mathematics and Applications Conference (EMAC03, Sydney, Australia)*, pp. 229-234.
- Sangwin, C. J.: 2004, [Assessing mathematics automatically using computer algebra and the internet](#), *Teaching Mathematics and its Applications* **23**(1), 1-14.

Strickland, N.: 2002, Alice interactive mathematics, *MSOR Connections* 2(1), 27-30. <http://tsn.mathstore.ac.uk/newsletter/feb2002/pdf/aim.pdf> (viewed December 2002).