AIM - A PARABLE IN DISSEMINATION

Joseph KYLE
School of Mathematics and Statistics,
University of Birmingham,
Birmingham, B15 2TT, UK
e-mail: j.kyle@bham.ac.uk

Christopher J SANGWIN
School of Mathematics and Statistics,
University of Birmingham,
Birmingham, B15 2TT, UK
e-mail: c.j.sangwin@bham.ac.uk

ABSTRACT

AIM is an initiative in interactive mathematics, which exploits the power of the computer algebra package Maple V in an extremely flexible way that can be applied to a variety of curricula.
The authors present the results of a project concerned with replacing parts of core first year materials at the University of Birmingham, UK.
Most software for computer-based assessment has limited use in mathematics. Common problems are:
· Poor display of mathematical expressions. (despite MathML and plug-ins like IBM TechExplorer),
· Restricted choice of question types,
· Failure to recognise mathematically equivalent solutions,
· Difficulty of assigning partial credit,
· Inability to test students' creativity (eg give an example of a function which satisfies XXX but does not satisfy YYY)
Effective integration of computer algebra has made it possible to address these issues. The ability to monitor students' progress in more detail has allowed us to provide individual students with tailored advice on suitable additional learning opportunities (e.g. the use of appropriate learning packages) and to efficiently mount support activities (e.g. targeted small group sessions). This has enhanced and made more focussed support for our students.
Pilot studies have been very encouraging. Students find the software easy to use (97% agree/strongly agree), like the immediate feedback (100% agree/strongly agree), and find it helpful (87% agree/strongly agree).
Our paper:
outlines the genesis and nature of AIM,
reports and elaborates on the above results,
offers an indication of the range of applicability of this shareware - from widening participation to honing advanced specialist skills.
Of particular interest is a parallel study, which explores the factors, which determine whether an innovation is likely to be easily transferable. We look to distil principles of value to innovators in the learning and teaching of mathematics.

1. Introduction - What is AIM?

AIM is a system for computer-aided assessment (CAA) in mathematics and related disciplines. It has been tries and tested in both summative and formative assessment, with the emphasis leaning towards the latter. The acronym, introduced by the original developers at the University of Gent in Belgium, stands for **Alice Interactive Mathematics**. From there it has rapidly been embraced by academics around the world and is now undergoing further development in the UK. The original Belgian site is to be found at:

http://allserv.rug.ac.be/~nvdbergh/aim/docs/

Further information and examples for English speaking users is at:

http://www.mat.bham.ac.uk/aim/

and the most recent revisions, and documentation and downloads are available from:

http://aim.shef.ac.uk

Each of these sites offers ample opportunities for visitors to log on as a "guest" and interact with AIM in a number of mathematical situations.

2. What is distinctive about AIM?

Certainly there is no shortage of CAA systems that have been developed for mathematics. See for instance the articles appearing in the on-line periodical:

http://ltsn.mathstore.ac.uk/articles/maths-caa-series/index.htm

and the archives of MATHS-CAA@JISCMAIL.AC.UK at:

http://www.jiscmail.ac.uk/lists/maths-caa.html

We feel therefore that it is worth outlining what are seen to be the distinctive features of AIM.

AIM Exploits Computer Algebra In the case of AIM, the underlying computer algebra package is Maple. The full power of the mathematics programmed into Maple is therefore available to be called upon in both the authoring and the checking procedures.. It also generates a very high degree of flexibility for authors and participants. For example, if a student gives a correct answer in a form different from that supplied, AIM can still determine that it is correct.

- If a student solves a system of equations incorrectly, then AIM can substitute the incorrect answers back in to the equations, and show student that they do not work out.
- If a student integrates an expression incorrectly, then AIM can differentiate the incorrect answer and show the student that it is not same as the original function.
- More detailed feedback is available for certain common errors. For example, one can set a standard integration question asking students to integrate $\sin^2(x)$. Without any explicit action by the question setter, AIM will examine the form of the integrand and recognise that student be tempted to answer $\sin^3(x)/3$, or possibly $\sin^3(x)/(3\cos(x))$ etc. AIM can automatically generate explanations to cover these common errors.
- More generally, questions can be set up to give immediate and detailed feedback depending on mathematical features of the answer offered. It has to be admitted that this is easier for some (types of) questions than others but then this is probably the case for all CAA systems.
- Deeper understanding can be tested by asking the student to give an example of a mathematical object - say a function, a matrix, or a vector - with some specified properties. Through this kind of exercise one can begin to test, via CAA, the so-called higher mathematical attributes. See (Sangwin, 2002) for a further discussion of this issue. Clearly

a CAA system can only deal with questions of this type if it has enough mathematical "intelligence", such as Maple or a similar engine, behind the scenes.

Freedom of Ownership The source code for AIM is and always has been freely available within the academic community. From its inception in Belgium, a growing community of active mathematicians in Belgium, Canada, Australia, the US, the UK and elsewhere has adopted the common aim to develop a freely available resource for those with an interest in using technology to enhance learning and teaching in mathematics. Being owned by these academics, AIM is highly responsive to the interests of the discipline and its students. It is also inexpensive to set up; one copy of Maple running on one server, allows a department to mount a number of AIM sessions on the web. There is no further cost and the system makes no demands upon the student other than they access the web via a conventional browser.

Freedom of Expression AIM can perfectly well present questions in any of the usual CAA formats - multiple-choice, multiple response, numerical/text input. But AIM is at its most powerful when students are required to enter their answers as free text using Maple syntax. For example, an answer of $\mathbf{sin^3(x)/(3cos(x))}$ would be entered as **sin(x)^3/(3*cos(x**)). There is little doubt that questions of this type have a greater pedagogical value than, say, the commonly used multiple-choice vehicles. On the other hand, critics of AIM point to this need for syntax as a disadvantage of AIM. Advocates would point out two things:

1) For students with little or no facility in CAA environments, we would (as in any CAA system) restrict ourselves to multiple-choice and other "easy" formats.
2) There is a substantial help system that has been designed to support students as they negotiate the syntax. In particular,
   a) Answers that cannot be parsed are discounted without penalty. Students can ask AIM to parse their answers without marking them and to check that they are interpreted as intended.
   b) If a student enters an answer with mismatched brackets, then AIM can indicate graphically which brackets match against which other brackets, and which bracket is causing a problem.
   c) If a student forgets the syntax for multiplication (eg **5x** in place of **5*x**) then AIM will generally indicate the omission and report back the student's answer with the suggestions highlighted.
   d) Similar feedback is given for a variety of other common errors, such as "**t^-2**" in place of "**t^ (- 2)**", cosx in place of cos(x), and so on.

Of course the "help" in AIM is, as any other system, largely heuristic and based upon matters of judgement. Users will vary in their reactions to items. Nonetheless, as it evolves, students generally find it increasingly helpful. (See Section 4 below.)

Some see that the currently popular graphical interfaces for building up and checking mathematical expressions as the "solution" to the "syntax problem". However, many users of AIM see genuine educational benefits in asking students to come to terms with the syntax. For such educators the argument is that all mathematics graduates should be able to enter reasonably complex mathematical expressions into some kind of computer system and should have some awareness of how a system such as Maple operates. Some would go further and argue that interaction at a programming level really test whether a mathematical concept has been mastered. Dubinsky (2000) in another context has argued a similar case in the use of technology in testing understanding in areas such as group theory and other "advanced" areas of pure mathematics. AIM is a very good vehicle for encouraging these skills. It has more detailed and user-friendly help files than Maple itself and is flexible enough to allow authors some control over the level of rigour in syntax expected of students.

3. Authoring in AIM

Questions are authored using a simple web interface to edit a plain text file containing lines of code.

```
1. t> Give an example of a differentiable function
2. t> <i>f(x)</i> which has a turning point at <i>x=1</i>
3. v> 2
4. ap> <i>f(x):=</i>
5. s> [(ans)->`aim/Testzero`(subs(x=1,diff(ans,x))),x^2+2*x+3]
6. end>
```

Each line begins with an AIM "flag". For example, line 1 begins with the flag "**t>**" which instructs Maple to display this line as **text**. The body of the question is written in a simple mark up language, which may include standard HTML commands. The six lines above will generate the following:

Give an example of a differentiable function *f(x)* which has a turning point at *x = 1*.

(followed by an answer entry box)

Full details on authoring in AIM may be found in Klai et al (2000) or at the web sites listed in Section 1, but one more example will be useful in illustrating the ease with which AIM can introduce powerful randomisation and feedback facilities.

```
1. h> p_ :=rand(1..3)();
2. h> q_ :=p_+rand(1..2)();
3. t> Give an example of a cubic polynomial <I>p(x)</I>
4. t> with the following properties
5. t> <OL> <LI> <I>p(0)=1</I>
6. t> <LI> <I>p(x)=0</i> at <I>x=$p_</I> and <I>x=$q_</I>.</OL>
7. v> 4
8. ap> <I>p(x):=</I>
```

The lines above generate the following question:

Give an example of a cubic polynomial *p(x)* with the following properties

1. ***p(0) = 1*,**

2. ***p(x) = 0* at *x = a* and *x = b***

At each presentation of the question, the parameters ***a*** and ***b*** are randomly selected as indicated. The flag "**h>**" represents an instruction to hide from the student the randomisation process. The flag "**v>**" sets the "value" or number of marks for the question and "**ap>**" generates an "answer prompt" for the student.[1] Conventional HTML commands have been incorporated in lines 3, 5, and 6 to generate an ordered list and italic display.

However, a few more lines of code can allow immediate tailored feedback to student. Lines 9 to 24 provide the marking regime and detailed feedback. Here we see how each of the four conditions is checked. If the student's answer fails then feedback is given and marks deducted.

---

[1] Note the line numbers are not part of the code. They appear here simply to facilitate reference.

Line 23 ends this procedure and provides a correct answer, in terms of the random variables, which will be substituted in before being displayed the system.

```
9. s> [ proc(ans) local marks_; marks_:=1;\
10. if not `aim/Testzero`(subs(x=0,ans)-1) then\
11. printf("Your polynomial fails to satisfy <I>p(0)=1</I>.<BR>");\
12. marks_:=marks_-0.25; fi;\
13. if not `aim/Testzero`(subs(x=p_,ans)) then\
14. printf("Your polynomial fails to satisfy <I>p(x)=0</i> at <i>x=%g </i>.<BR>",p_);\
15. marks_:=marks_-0.25; fi;\
16. if not `aim/Testzero`(subs(x=q_,ans)) then\
17. printf("Your polynomial fails to satisfy <I>p(x)=0</i> at <i>x=%g </i>.<BR>",q_);\
18. marks_:=marks_-0.25; fi;\
19. if not `aim/Testzero`(degree(ans,x)-3) then\
20. printf("Your polynomial is not a cubic.<BR>");\
21. marks_:=marks_-0.25; fi;\
22. marks_;\
23. end ,(1-x/p_)*(1-x/q_)*(1-x)]
24. end>
```

The powerful randomisation features built into Maple allow authors a great deal of flexibility. For example, the next few lines generate a question testing whether students can integrate a monic polynomial of degree 5 with many of the coefficients randomly varying within a pre-selected range (here integers from -9 to 9).

```
1. h>p_:=x^5+randpoly(x,degree=4,coeffs=rand(-9..9));
2. t> Evaluate the following integral:
3. p> Int(p_,x)
4. s> [proc(ans) 'aim/Test'(diff(ans,x),p_) end,int(p_,x)]
```

For a new user the lines 9 to 24 above may seem a little daunting, but a novice may start authoring at the level of simply editing existing freely available questions such as the four lines immediately above. With a little more confidence, new questions can be written *ab initio*, using the language and notation students have already seen, and linking if desired to a course or departmental web site.

4. What students think?

Over the last few years, colleagues have moved on from installing and running AIM to a process of evaluation. A major study (involving around 180 students) at the University of Birmingham is assessing through questionnaires and student focus groups, student reaction to AIM. The results of this study are still being compiled and analysed and will be reported in detail later. However, the following quotes from students indicate that they understand and appreciate the underlying purpose.

"A: Question 1 was certainly asking us stuff that we had to think about.

Interviewer: In what way?"

"A: You didn't give us an equation and then say "solve it". You have got to really think about what it means. You have to get a solution and then you think, OK that's the answer. Doing a question like that you think, argh, right, that is the shape of the graph."

Generally the students demonstrated a mature understanding of the rationale behind these questions. There was widespread agreement that such questions

"B ... test your understanding of the subject, rather than your ability to turn a handle."

At the same institution, AIM was used for part of the assessment of an advanced FORTAN course. When asked students (around 40) found the software easy to use (97% agreed/strongly agreed), liked the immediate feedback (100% agreed/strongly agreed), and found it helpful (87% agreed/strongly agreed).

At the University of Sheffield AIM was used with over 200 first year students. An analysis (Strickland, 2002) of their opinions revealed *inter alia* that they most liked:

- The fact that you could try again if answers incorrect (this was extremely popular).[2]
- Instant feedback
- The ability to do questions from home, in their own time, without any pressure.

Least popular were:

- Difficulties with syntax
- Lack of "method marks"[3]

Finally, in Belgium we find that in a class of around 45, 69% preferred assignments conducted and marked on the web by AIM compared to 31% who preferred paper assignments. Furthermore,

- 94% used their home computer (of those 66% used home computer exclusively)
- 25% used library computers
- 15% used computer university computer labs.

Overall 88% rated AIM as good or very good.[4]

5. Dissemination - what colleagues think?

This is perhaps the most interesting intriguing feature arising out of the genesis of AIM. Given its relative youth and the lack of organised sponsorship or marketing, AIM has spread remarkably easily among the academic community. There is already an active email discussion list serving the world-wide community of users. This contrasts with a number of other perfectly good and better-promoted systems. Through their connection with the LTSN Maths, Stats and OR Network in the UK the authors are aware of a number of worthy developments and initiatives in this area. But there is a noticeable inertia against these systems spreading to new users. None of these comments is intended to imply that these packages are intrinsically poor or defective. However the observation did prompt a discussion of exactly what features might assist in the dissemination of initiatives in learning and teaching. When asked, users of AIM quote the following reasons for adopting AIM.

- The ability to develop an "intelligent" system which exploits the power of computer algebra to address the mathematical needs of the course
- That it is an open source project developed and owned by the academic community
- The high degree of flexibility in authoring and the freedom to customise questions
- That it is deliverable over a standard browser with no additional plug-ins required.

---

[2] AIM allows lecturers to control whether students may try again. It also allows the imposition of a penalty (say 10% or 15% of the available marks) for second and subsequent attempts.

[3] This has been addressed to some extent by authors who have split questions and allowed say 50% for the subsequent correct manipulation of a wrong procedure or function.

[4] The complete results of the Belgian survey are presented in Table 1

These features (and others mentioned in earlier sections) are not necessarily unique to AIM. AIM is not the only system that meets these criteria. However taken as a whole they explain why AIM has been so readily adopted by international mathematicians.

6. Conclusions

As the debate about the relevance of CAA in the assessment of higher mathematical skills unfolds, it is clear that the needs of the discipline will not be met by many of the existing vehicles, relying as they do upon methods such as multiple-choice questions. At the same time encouraging reluctant practitioners to embrace new technological initiatives will be hampered by even the slightest obstacles be they financial, technological or pedagogical. In this paper, the authors wish not so much to advocate AIM (history shows that it needs no special pleading) but to open a discussion on the principles that will underpin easy transfer of initiatives in learning and teaching mathematics. Within the context of CAA, the four points in Section 5 have been distilled out of our observations. Too often initiatives thrive locally, fed perhaps by the charisma or commitment of their authors, but do not travel well. If we can identify the design features that will assist transfer and dissemination of such initiatives, then the whole mathematics community will benefit.

REFERENCES

Dubinsky, E., 2000 "Towards a theory of learning advanced mathematical concepts", Proceedings of the 9th International Conference on Mathematical Education, (to appear)

Klai, S., Kolokolnikov, T., Van den Bergh, N., 2000 "Using Maple and the Web to grade mathematics tests", *International Workshop on Advanced Learning Technologies, Palmerston North, New Zealand.*

Sangwin, C. J., 2002a "New opportunities for encouraging higher level mathematical learning by creative use of emerging computer aided assessment" (to appear)

Strickland, N., 2002 "Alice Interactive Mathematics", MSOR Connections, Vol 2, No 1, 27 - 30

There was sufficient instruction/resources provided to start using AIM:	72	25	3	0	0
AIM was easy for me to learn to use:	63	38	0	0	0
The opportunity to re-attempt questions in case of a wrong answer helped me to better understand the material:	56	34	6	3	0
The feedback from AIM was helpful:	19	44	22	6	9
The penalty system (deduction of 15% for every wrong answer) was fair:	38	16	19	25	3
The questions asked in AIM were capable of testing material that required a deeper understanding of the course:	34	41	9	13	3
The time available to complete the assignments was adequate:	69	22	6	3	0
Give your overall rating for AIM the way you experienced it, on the scale from +2 (good) to -2 (bad):o	47	28	22	3	0
Give your overall rating for the potential that AIM has for use in future math courses.	44	44	6	6	0

Table 1 - Views on AIM from Belgium (courtesy of http://allserv.rug.ac.be/~nvdbergh/aim/docs/)