Maths-CAA Series
July 2004

A Conservative Approach to Computerised Marking of Mathematical Assignments

Contents

The Challenge  2
The Solution  3
Randomised problem sheets  4
Immediate feedback  5
Saving marking time  7
The Implementation  8
The technology  8
Authoring questions  9
Summary  10
References  11
A Conservative Approach to Computerised Marking of Mathematical Assignments

Gustav W. Delius
Department of Mathematics, University of York, York, YO10 5DD
Email: gwd2@york.ac.uk

Abstract: Computerised marking of weekly mathematics assignments has several potential benefits: 1) randomised questions reduce plagiarism, 2) immediate feedback increases motivation, 3) staff-time spent on marking is reduced. A practical way of introducing computerised marking into departmental teaching, gradually, without disrupting current practice, is described in this article. The approach has been followed at the University of York during the last academic year. It makes use of the open-source software packages AiM and Moodle.

1. The Challenge

It is reassuring to see that university mathematics courses follow the same pattern almost everywhere in the world: students attend two or three lectures every week and are then given a weekly assignment consisting of a collection of problems that the students take home and work on in their own time. They are usually given one week to complete the assignment and hand in their solutions. These are then marked, either by the lecturer or by a graduate assistant, and handed back to the student about a week later. The marks may contribute a small amount to the student’s final grade. Usually there is a regular problems class during which the assigned problems, or similar ones, are discussed in smaller groups.

This system, which is so universally followed, is designed to encourage the student to apply the knowledge acquired in the lectures and to make it their own. While the system is very sound in principle, there are some flaws in practice. Three of the major problems are:

1. Some students do not attempt to solve the problems themselves but instead copy the solutions. Unfortunately, these are usually the weaker students who are most in need of the practice afforded by doing the problems.
2. Students have to wait about a week before finding out whether or not they have solved the problem correctly. When they finally find out that there was a mistake, there is no incentive for them to correct it. The “learning from one’s mistakes” is not taking place.
3. The marking of the assignments takes up a lot of staff time that would be better spent in direct student contact.

Luckily, all of these problems have an obvious solution:

1. Give every student a slightly different set of problems.
2. Give immediate feedback to the student.
3. Automate the marking.

Clearly, this requires computer-based solution. The challenge is to implement this without disrupting the usual system that, after all, serves quite well. The desire is to start to solve the problems while changing the existing system as little as possible, and to allow the change to take place gradually.

The most important point to keep in mind is that the kind of problems assigned to the students are those that the student needs to take home and spend some time on to carefully work out the solution. It is not appropriate to replace these by some quiz questions that the student answers while sitting at a computer. After all, the goal is not to test the students’ knowledge but to motivate them to deepen their understanding by working on challenging problems. What is wanted, then, is very different from traditional computer-aided assessment.

2. The Solution

This article describes how this challenge has been addressed in the Department of Mathematics at the University of York. The solution consists of providing the lecturer with the means to computerise some of the questions on the weekly problem sheets.

The technological heart of the system is AiM (www.aiminfo.net), a sophisticated CAA system that uses the power of the computer algebra system Maple (www.maplesoft.com) to randomise and to mark mathematics problems. AiM and its uses have been the topic of previous articles in this series [1,2] and elsewhere [3,4,5]. This article concentrates on describing how it is being employed in the weekly assignments at York.

Users interact with the system through Moodle (www.moodle.org), a particularly user-friendly and easy-to-use virtual learning environment. Most of the lecturers at York have been making course material available on the web for a long time. Moodle simply provides a convenient mechanism for doing this and for collecting everything in one place on the web. It also facilitates communication among the students and between students and lecturers.
Some details about the implementation of this system are described in Section 3. In the following three subsections the most important aspects of the York approach are summarised using the same division into 3 points employed in the introduction.

2.1 Randomised problem sheets

It has always been the practice at York to hand out typeset problem sheets prepared using TeX or LaTeX. This practice will continue with only one subtle change: each student receives a slightly different version of the problem sheet. The randomisation is performed by AiM. This process can be explained with an example problem from vector calculus:

*Use Green’s Theorem to evaluate the line integral*

\[ I = \int_C x^3 y^2 \, dx + 2xy^3 \, dy \]

where \( C \) is the triangle with vertices (0,a), (2,0), (-2,0).

In the past the lecturer used the following TeX source to typeset the problem:

\[ \text{Use Green’s Theorem to evaluate the line integral} \]

\[ \int \left\{ \begin{array}{l} x^3 y^2 \, dx + 2xy^3 \, dy \\ \text{where } \{C\} \text{ is the triangle with vertices} \end{array} \right\} \]

\[ (0,a),(2,0),(-2,0) \]$

Now, in order to randomise this problem, the following AiM source is used:

\[ r := \text{rand}(1..3): \]

\[ P := r()^2 r()^2 r(): \]

\[ Q := r()^2 r()^2 r(): \]

\[ \text{xmax} := r(): \]

Notice that the AiM syntax is a combination of Maple syntax and TeX syntax. The lines after the *h>* flag are Maple commands, to choose the random parameters. The lines after the *t>* flag are TeX, with an important extension: things enclosed in @...@ are evaluated by Maple and then substituted in. When processed by AiM this source will produce the same kind of problem as the one above but with randomised integrand and contour.

The system combines the randomised problems produced by AiM with the conventional problems to produce the required number of personalised weekly problem sheets and sends them to the laser printer. The lecturer can then take these to the lecture where each student picks up a random copy. Each random problem sheet has a number in the corner to identify it.

It is desirable to encourage the students to discuss mathematics with each other. It has thus been a pity that working together on the assigned problems had to be discouraged in the past. With the randomised sheets the students can be told that they are allowed to discuss the solution method together as long as they apply it afterwards to their particular problem by themselves.

The differences between the problem sheets do not have to be large, just large enough to discourage students from copying the answer of a fellow student.

2.2 Immediate feedback

Once students have worked out the solution to a computerised problem they can go to any computer with an Internet connection, browse to the course homepage on the departmental Moodle website and log in. In order to get the same version of the problem sheet that they worked on at home they type in the identification number from their problem sheet. They are then presented with a webpage containing the questions and boxes into which they can type their answers.
2. If the problem does not have a unique answer then AiM can be told how to check that the student’s answer has the required properties. There is no limit to the sophistication in this marking procedure; the question author can employ the full power of Maple’s programming language.

The pedagogical benefit of the new system arises when students do not get the correct answer. They will then be encouraged to go away and find their mistake as when they have found it they can come back to the computer and type in the corrected result. If necessary they can repeat this several times. There will be a penalty, however, for each wrong attempt in order to encourage students to do their solutions carefully the first time. By default, this penalty is 10% of the total mark, but this can be changed for each problem individually.

Note that students are given credit only if they finally arrive at the correct answer. This is a drastic change from current practice. It is conventional to give a student partial credit for a solution even if the answer is incorrect. Unfortunately, this practice of giving partial credit for incorrect answers has led to a culture where some students just scribble something down in the hope of attracting marks. These students are not being motivated to do the work carefully because they are given the wrong signal that getting the answer right is not essential. The computerised assignments, on the other hand, give the student both the possibility and the incentive to persevere until the answer is correct. These assignments, thus, train the students in the important skill of finding their own mistakes. This is a crucial mathematical practice as, so often, personal experience demonstrates. The York experience with computerised assignments has shown two things:

1. the students do indeed persevere and by the time they get to the right answer they understand the material much better than after the first attempt.
2. students are motivated to spend a lot more time on their assignments.

This second effect has to be taken into account when designing the assignments.

2.3 Saving marking time

Conventional marking of the weekly assignments takes up a lot of time that would be better employed differently. It is clear that having some computerised problems on a problem sheet cuts down on marking time because the marker does not have to look at the solutions to these problems in detail. It is, however, important that the time saved is not lost again in the process of adding together the computer marks and the marks for the conventional problems.
This is avoided by letting Moodle collect the computer marks automatically. The marker simply types in the marks from the non-computerised problems and Moodle takes care of everything else. Both students and lecturers can always easily get an overview of the grades on the Moodle site.

Solutions to problem sheets are marked not only in order to give feedback to the students but also so that lecturers can gain an impression of how well the students are absorbing the material being delivered in lectures. Ideally, the observations made in marking feed back into the teaching during lectures and problem classes, where the points that have proven difficult in the assignments can be repeated or explained more fully. Again, computerised marking has several advantages. One is timing. The information about how students are doing is available as soon as they input their answers into the system, i.e. even before the due date. This enables relevant points to be stressed in the lectures before new subjects are introduced. The second advantage is convenience. The AiM system provides the lecturers with several tools to analyse the results. The lecturer can pull up an overview of how the class did on a particular problem or look at all the attempts an individual student made. The lecturer can even ask for a list of all the students who made similar mistakes. It is much better to do this in a problems class after having seen an overview of the students’ performances. Moreover, Moodle shows the results alongside the students’ photographs. Students in large classes are quite thrilled when they are approached in person with verbal feedback on their recent work.

Clearly, besides the time saved through the automated marking, the time spent in setting up and testing the computerised problems has to be considered. This is an important issue and will be discussed further in the next section.

3. The Implementation

The system was developed and implemented in the summer of 2003 with the help of a small grant from the University of York through its Fund for Innovation and Development in Teaching and Learning.

3.1 The technology

The software required for the computerised assessment system can run on any modern PC with an Internet connection under Windows or Linux. This computer must also have Maple version 7, 8 or 9 installed.

Students and lecturers access the system from their own computer through a web browser. They do not need any extra software installed on their computer.

As already mentioned, the York system combines the VLE Moodle with the CAA system AiM. One point worth stressing is that both Moodle and AiM are open-source software, which has allowed both to be modified in such a way that the users look like a single system. Most of the software development necessary to achieve this integration was done last summer by Alex Smith, then an undergraduate student in the Department.

The integration of the assessment system into a VLE has been crucial for its acceptance. The students immediately appreciated the convenience afforded by the VLE of having all their course materials and information available in one user-friendly place. This positive attitude towards the VLE then extended towards the assessment system that they saw as one piece of a package.

The system was rolled out across the whole department for all the undergraduate modules. Each module was given its page on the VLE and all students were enrolled automatically, using data exported from the university’s student record system. It was made departmental policy that all coursework marks for all modules were to be entered into this VLE. Every lecturer was then free to choose whether they wanted to use some computerised questions on their problem sheets or not.

3.2 Authoring questions

The experience at other institutions suggested that the main stumbling block to the introduction of the AiM assessment system into a department was that lecturers do not have the time that needs to be invested initially to author the questions for AiM. To overcome this problem a team of seven undergraduate students was assembled in the summer with the task of transferring some of the existing assignment questions to AiM. For each student the initial learning curve was steep, so between them the students spent in the order of 1000 hours to convert about 200 questions for 13 different undergraduate modules. If this exercise were repeated then a smaller team would be chosen to reduce the number of hours spent on this task.

What is really needed is a simple and convenient authoring tool that will allow lecturers themselves to computerise some of their simpler questions more quickly. A grant from JISC has been sought to develop such tools, and to make mathematics assessment systems interoperable with each other and with other components of a managed learning environment. If this is approved then these tools will become available in spring 2005.

One important lesson learned is that it is vital to test the marking routines very
carefully. In the beginning it occasionally happened that a student's answer was marked as incorrect when it was correct. This may also sometimes happen in hand-marking but it has a much more devastating effect when it happens in an automatic system because it may lead students to waste a lot of time chasing for mistakes in their solution when there are none. Understandably, this situation got students very annoyed. The solution is to have the questions tested thoroughly and independently by several marking assistants before their first use on students.

4. Summary

In this article some of the potential benefits of computerised assignments have been pointed out and a way of blending them seamlessly with the existing practice has been described. The following points summarise some of the aspects that were important for the success of the approach taken at York:

- The computerised problems are integrated into the usual printed problem sheets. This encourages the student to take the computerised problems just as seriously as the conventional ones.
- The students enter their solutions through Moodle, which they use for their other course materials anyway. Moodle is accessed via the web, i.e. the student can do this from any computer.
- The students are introduced to the use of Moodle and AiM in a session at the start of term. Thus, no lecture time is stolen from the individual modules.
- The system is set up automatically for all undergraduate modules so that any interested lecturer is free to start using the system at any time.
- The marks for both computerised and conventionally marked problems are collected and combined automatically by Moodle. The students can check their marks and reassure themselves that they are correct.
- After the due date the students can download typeset solution sheets containing solutions to both the conventional problems and the randomised ones.

It is important to realise that only some questions lend themselves to computerisation. It is, therefore, important that the system is designed in such a way that it can be used efficiently even on only a small proportion of the assignment questions in a module. It also helps to overcome initial scepticism if lecturers are given the possibility to start with just one or two computerised questions in their module in order to familiarise themselves with the concept.

The impact that the introduction of the new system has had on the experience of students and lecturers has been the object of a study by Richard Walker, the university’s learning technologist, [6].

This article has concentrated on presenting the use of AiM and Moodle in the weekly homework assignments. Of course, the introduction of the software opens up several other opportunities that are not discussed here.

Any university department that is teaching some mathematics is likely to profit from introducing a similar system to the one described in this article. It is hoped to make the software available for this purpose with the help of a JISC grant. If this grant is approved then the resources to help interested departments with the installation and initial rollout of the system will also be available. If you would like to be kept informed of future developments please register at http://maths.york.ac.uk/serving.maths/. There is also a discussion forum here where you can raise any questions not addressed in this article.

References