Teaching Mathematics in Online Courses
An Interactive Feedback and Assessment Tool

Edgar Seemann
Furtwangen University, Villingen-Schwenningen, Germany

Keywords: Online Courses, Mathematics, Assessment, Grading, Teaching, Tutoring.

Abstract: Online courses often require students to work self-dependently using books or video material. For abstract subjects such as mathematics this is particularly challenging. To improve student motivation and learning results, we propose an interactive feedback and assessment tool tailored to math exercises. Our system is able to process and analyze mathematical expressions using an underlying computer algebra system. It allows teachers to create exercises with a much wider range of question types as it is possible with today's learning management systems, which are mostly restricted to multiple choice questions. We can provide automatic individual feedback to students for almost any kind of mathematical exercise. Thus, making it easier for students to practice and study math in a self-dependent manner.

1 INTRODUCTION

Mathematics is the foundation of every engineering discipline. That is why teachers want students to reach a certain level of mathematical understanding. Meaning, that they should be able to apply learned techniques to new problems or even real world problems.

There is a consensus among teachers, that in order to master mathematical problem solving, students need to practice.They need to practice calculation methods e.g. how to compute the derivative for a certain type of function. And they need multiple examples to internalize the underlying ideas and to be able to generalize a technique to other tasks or settings.

In on-campus university classes students try to solve exercise problems and get feedback and assistance from their teachers or teaching assistants. Providing adequate feedback is one of the most important steps in the learning process. Without it, students often internalize incorrect rules or develop incorrect notions of mathematical problems and methods.

This feedback is ideally immediate or at least provided in a timely manner. Students need to be encouraged with positive feedback, if they solved a problem correctly. And they need to be made aware of mistakes as soon as possible. Note that this does not necessarily mean that step by step instructions are provided for each exercise. The teacher should be able to control the complexity of the exercises and the granularity of feedback to foster deep learning without frustrating the students.

For online courses this poses a particular challenge. On the one hand, communication between the teacher and students is still and always will be more difficult than in a face-to-face conversation (even though virtual classroom software is improving). On the other hand, the communication is, to greater extent, asynchronous.

Online courses are often taught through a series of videos combined with a discussion forum. Feedback to student questions is therefore typically delayed by hours or days. Live virtual classroom sessions typically only take place at the beginning or the end of larger learning units. Finally, with the advent of Massive Open Online Courses (MOOC) the teacher to student ratio makes individual feedback a challenge.

Without appropriate communication and feedback many students fail to achieve the required skills. In fact, the success rate for online courses is in most cases considerably below success rates of on-campus courses with a comparable student body. In (Collins, 2013) and (Thrun, 2013) the difference in success rate is as large 40% for an online course compared to the respective on-campus class in spring 2013.

In order to improve teaching of mathematical problem solving skills in online courses, we believe it is necessary to provide meaningful, individual feedback to every student, no matter whether there are 30 or 30000 students in a class. Learning management

Seemann E..
Teaching Mathematics in Online Courses - An Interactive Feedback and Assessment Tool.
DOI: 10.5220/0004939204150420
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 415-420
ISBN: 978-989-758-020-8

software e.g. (Moodle, 2013; OLAT, 2013) provides interactive (self-)tests, which allow students to practice and evaluate their skills. Thus, obtaining feedback on their learning progress. Unfortunately, giving automatic feedback for mathematical exercises is difficult. Responses may not only involve numeric values, but expressions may also include functions or mathematical sets which have many equivalent representations (e.g. $x(x+1)$ is equivalent to x^2+x). These mathematic expressions cannot be handled properly by state-of-the-art learning management systems.

For a meaningful feedback an interactive learning tool needs to better "understand" these mathematical expressions. As a first step in this direction we therefore propose an automatic feedback and evaluation system tailored to mathematical exercises. Our system leverages existing computer algebra software in order to evaluate student responses and to provide immediate individual feedback. It allows teachers to easily create custom content with the granularity of feedback of their choice.

With the automatic feedback functionality turned off, the system can also be used for efficient assessment and grading of large numbers of students.

2 ELECTRONIC SUPPORT FOR MATH TEACHING

In abstract subjects such as mathematics feedback plays an important role in the learning process. While self-studying from a book or video is important, students require much more support compared to facts-based subjects (e.g. history).

Traditionally, this support is to a large extent provided through face-to-face communication with teachers, teaching assistants or fellow students. When teaching online, we need to replace this valuable form of feedback using different means.

2.1 Virtual Classrooms

Virtual classroom software has gained widespread acceptance in recent years. Universities use commercial services like Adobe Connect or Google Helpouts [1] for live sessions between students and teachers. Virtual classroom software provides real-time audio and video communication. With the included whiteboard and screen sharing capabilities this setup can mimic the lecture style of large on-campus lectures. That is, a professor is explaining and students are listening.

[1] http://www.adobe.com/products/adobeconnect.html, http://helpouts.google.com

Student questions in these virtual classrooms are often tedious due to organizational issues (who is allowed to talk at which point?) and technical problems (microphone setup, network delays etc.). More over, the software or network often only allows for a limited number of participants. When groups are small, virtual classroom can, however, be a decent tool to provide feedback. Particularly, since touchscreens of today's notebooks and tablets make it easy to write mathematical equations and formulas.

2.2 Forums

Forums are another popular tool for online learning. Communication is asynchronous and thus does not provide immediate feedback. Students may ask a question and a member of the teaching staff responds later on. While in some cases this might result in students working harder to solve a problem all by themselves, the delayed feedback often also disrupts and postpones the learning process.

Forums require the teaching staff to spend a large amount of time formulating written answers to student questions, which is tedious and does not scale well to large amount of students.

2.3 Learning Management and Tutoring Systems

Existing learning management systems e.g. (Moodle, 2013; OLAT, 2013) and tutoring systems e.g. (Koedinger and Corbett, 2006; Melis and Siekmann, 2004) offer the functionality to create electronic exercise sheets. This allows students to practice and to obtain immediate feedback to their solutions. These systems may also display context-sensitive information, hints and instruction to guide students towards reasonable next steps.

LMS: Learning Management Systems. Generally, teachers are required to reformulate mathematical problems in a way, that is supported by the LMS. That is, in order to allow automatic feedback, exercises have been in the form of multiple choice or fixed answer questions.

Since these systems do not understand mathematical expressions and their equivalent representations it is impossible to use more complex expressions as a solution. Exercise questions allowing free-text fields are of course possible, but they require manual intervention, which leads to a delayed feedback.

Overall, current LMS limit the selection of possible exercises considerably, which is why most math teachers avoid using these systems.

ITS: Intelligent Tutoring Systems. In the context of this paper, we divide intelligent tutoring systems into two categories. One category of systems focuses on building student profiles while observing student performance in online tests and provides hints for the learner on its personal weaknesses and strengths (Schiaffino et al., 2008; Cheung et al., 2003),. These systems often build upon simple multiple choice and fixed answer questions and aggregate their results. They can tell a student e.g., that she/he has made many mistakes in the exercises on a certain topic. However, these systems cannot deal with mathematical expressions.

The other category focuses on domain specific problems. In the area of mathematics, there exist a number of these tutoring systems (Beal et al., 1998; Melis et al., 2009; Koedinger and Corbett, 2006). To our knowledge none of these systems leverage capabilities of symbolic mathematics software to support students. In fact, most of these systems focus on early high school level maths.

It is often hard for teachers to develop their own content for existing intelligent tutor systems. In fact, very few systems have detailed instructions on how to do this and in many cases programming skills are required. The Carnegie Cognitive Authoring Tools (Aleven et al., 2006) e.g. require authors to know Java or Adobe Flash. This is certainly one of the reasons, why the adoption of intelligent tutoring systems has been rather disappointing.

3 AN INTERACTIVE FEEDBACK AND ASSESSMENT TOOL

As pointed out in the sections above today's learning management systems are not prepared to deal with mathematical exercises.

The system proposed in this paper tries to combine interactive exercises with the capabilities of compute algebra systems. Our goal is to build a practice and assessment tool which is capable of "understanding" mathematical expressions and as a result give meaningful feedback and assistance to the students. Our system allows teachers to use a much wider range of exercise and question types as it is currently possible.

Contrary to many tutoring systems, we aim for math problems in general and do not want to restrict ourselves to a certain subdomain (e.g. dealing with fractions). That is, whether high school teachers want to design exercises to train the expansion of simple mathematical expressions or if university professors design exercises to solve differential equations, the capabilities of the underlying mathematical software should allow to provide meaningful feedback. More over, We want to make it simple for teachers to develop their own content without programming skills (see section 3.1).

3.1 Example Exercise

To get a better understanding of how the proposed system works, we present a short example exercise. We show how an exercise may be defined by the teacher and how the system reacts to user input. The technical details on how an appropriate user feedback is accomplished will be explained in subsequent sections.

Questions are displayed in a modern web-based user interface. Input fields allow students to enter their solutions and an appropriate feedback is returned once response has been submitted. In Figure 1 a student has provided an incorrect answer. Obviously he/she computed the derivative of numerator and denominator independently and the system displays the feedback directly below the input.

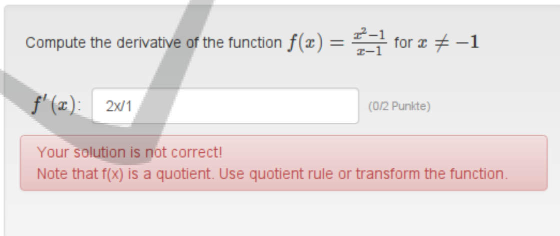

Figure 1: A hint is displayed if the provided solution is incorrect.

If a correct solution is entered, the input field turns green. Optionally, an exercise may yield points. Points may be used as a motivation or for grading purposes (see Figure 2).

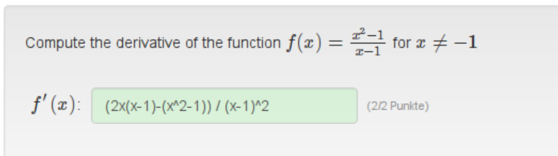

Figure 2: Correct solutions are marked green.

Finally, since the system recognizes mathematical equivalences, the user may input a different representation of the solution. In the above example the relatively complicated formula is, in fact, nothing else than the constant function 1 for values of $x \neq 1$ (see Figure 3).

Note that the system could also display additional feedback for correct but overly complicated solutions.

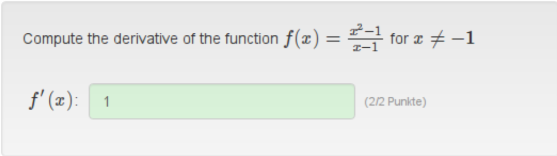

Figure 3: Equivalent solutions are recognized by the underlying symbolic math software.

Thus, hinting the student to a quicker way of solving the exercise.

Content Creation. Writing custom exercises is inspired by the syntax of markdown (Gruber, 2004), which has become quite popular in recent years. We use a relatively simple text format, with some control characters. The text format is later translated to HTML.

A new exercise is created by starting a line with three question marks followed by the title. Question text may contain LaTeX-style formulas (see below). LaTeX is still the most common format for defining math exercise sheets and teachers are familiar with it.

Since specifying a solution requires the definition of multiple values and properties, we opted for XML-syntax. A `field`-tag defines the expected solution type. In the example below this is a symbolic math value, i.e. a function, with the variable `x`. Embedded —answer—-tags define the behaviour depending on different user inputs. In the case below, the input of the value 1 results in returning points for the correct answer. The input $2x$ results in a customized feedback for the incorrect answer.

```
??? Derivatives
a) Compute the derivative of the function
$f(x)=\frac{x^2-1}{x-1}$ for $x\neq 1$
<field:symbolic var="x">
   <answer="1" points="2">
   <answer="2x" feedback="Note that f(x)
   is a quotient. Use quotient rule or
   transform the function.">
</field>
```

3.2 Solution Types of Mathematical Exercises

In order to allow teachers to use a wide range of different math exercises in an e-learning system, we need to be able to represent their respective solutions. In the following, we want to discuss the most common types of solutions for mathematical exercises. We will also point out the difficulties of representing and handling these solutions in an e-learning system.

Numbers. Numbers are probably the most common solution type for a math exercise. Even simple numbers have different ways of representation, which need special treatment in an e-learning system. The number 0.75 e.g. may be represented by the fraction $\frac{3}{4}$. If we want to allow students to enter either form, the e-learning system has to be aware of these representations. We also have to take care of irrational numbers like square roots or logarithms (e.g. $\sqrt{5}$).

One could argue that results could be restricted to rounded decimal numbers. Many teachers, however, prefer students to be able to solve exercises without a calculator. Solving a quadratic equation e.g. may easily lead to an irrational result, thus requiring a calculator to obtain a decimal representation.

Vectors and Matrices. For vector computations, we need to represent both matrices and vectors. In fact, since each entry in a vector or a matrix is a number, we also have to be aware of number representations as described above.

Intervals. Inequations often lead to number intervals as a solution. The inequation $x^2 - 4 < 0$ e.g. has all numbers between -2 and 2 as a solution. Intervals are essentially a pair of numbers with additional information of whether the interval is open or closed at the respective boundary.

Sets. Sets are unordered lists of numbers, which belong to a solution. The simplest example is the solution of a quadratic equation. The equation $x^2 - 1 = 0$ e.g. has the solutions -1 and 1.

Both sets and intervals can be combined using union, intersection or complement operators.

Functions. Exercises may also yield functions as a solution, e.g. when a derivative needs to be computed. There are many different possibilities to represent the same function, e.g. $x(x+1)$ is equivalent to $x^2 + x$. An e-learning system needs symbolic calculation capabilities to deal with these.

Geometry and Proofs. Finally, some exercises require students to sketch a geometric object or graph. Others require them to proof a mathematical property. Both of these two types are out of scope of this paper.

4 SYSTEM IMPLEMENTATION

In the following we will discuss how user input may be verified leveraging the capabilities of existing math software. From the results obtained by this software, we generate a meaningful feedback for the students.

4.1 Computer Algebra Software

Mathematical software packages have become extremely capable. Tools like Maple or Matlab (Maplesoft, 2013; MathWorks, 2013) are widespread in academia. They are capable of manipulating mathematical expressions (e.g. solving an equation). While these are also great tools to be used independently of an e-learning system, they essentially require users to learn a programming language. In some cases this is more difficult for students than learning the mathematics itself.

We therefore hide this complexity from the user and do the manipulation of the user input transparently in the backend of our system.

4.2 "Understanding" Mathematical Expressions

Verification of user input depends on the solution type of an exercise. As we have seen in section 3.2 math exercises may have various different types of answers. In the following, we will show how to verify answers for these different types.

Numbers. While numbers have different representations, we can compare them by their values. In our implementation this is accomplished by subtracting the number provided by the student from the actual solution.

In Matlab this can be expressed with the following code, which is executed when students submit their results.

```
isCorrect = 0;
diff = abs(input-solution);
if (diff<epsilon) isCorrect = 1;
```

Using the absolute value makes sure that we obtain a positive number for the difference. Note that number representations in a computer are never exact due to the limited number of bits available. Consequently, the difference of floating point numbers should never be compared to 0, but rather we need to make sure, that the difference is smaller than a certain epsilon. This also accommodates for rounding errors. In our implementation, we use an epsilon value of 0.001.

Vectors and Matrices. In order to compare matrices, we essentially have to compare the numbers for the corresponding rows and columns. Fortunately mathematic software makes this comparison easy. In fact, when using Matlab we can even avoid implementing nested loops.

```
isCorrect = 0;
diffmatrix = abs(input-solution);
maximum = max(max(diffmatrix));
if (maximum<epsilon) isCorrect = 1;
```

Note that we compute the difference matrix and then find the global maximum of rows and columns in the matrix. If this maximum is smaller then epsilon, the matrices are considered equal.

Intervals. Intervals are described by their endpoints and information of whether these are included in the interval or not. The interval $(1,3]$ e.g. includes all numbers between 1 and 3 excluding the number 1, but including 3.

To compare intervals, we need to parse the expression. First we look for the type of parentheses or brackets. Then, we split the left and right endpoints. Two intervals are considered equivalent, when their brackets match and when their endpoint numbers are identical with respect to the criteria described in paragraph 4.2.

Sets. Sets are unsorted lists of numbers. For verification, we sort the numbers to obtained an ordered list. This ordered list can now be handled in the same manner as a vector.

An implementation using MATLAB could look like this:

```
isCorrect = 0;
diffvector = abs(sort(input)-sort(solution));
maximum = max(max(diffvector));
if (maximum<epsilon) isCorrect = 1;
```

Functions. Comparing function representations is the most difficult among the presented solution types. We need to manipulate the provided expressions to make them match.

Given a user input of $x(x+1)$ and a solution x^2+x, we need to expand the input expression in order to allow a proper comparison. A straight forward expansion is not always sufficient to compare two symbolic expressions. In many cases, we have to do further simplifications, either for the user input and or the solution.

It might e.g. be necessary to simplify logarithms, exponentials or radicals. In such a manner, we can simplify an expression like $\frac{e^x-1}{1+e^{\frac{x}{2}}}$ to $e^{\frac{x}{2}}$. The algorithmic implementation details to perform these manipulations are described in (Fateman, 1972). They are readily available in state-of-the-art computer algebra systems.

In our implementation, we use the math software Maxima (Schelter, 2013) to do symbolic calculations. We compare function representations using the following code lines:

```
diff(x) := input(x)-solution(x);
simplified(x) := radcan(diff(x));
result(x) := expand(simplified(x));
```

Finally, we check whether the resulting function `result(x)` equals to zero.

5 CONCLUSION

In this paper, we have shown how an interactive feedback and assessment system tailored to math exercises can be implemented. The proposed system provides meaningful immediate feedback to students while practicing math. Thus, allowing them to better learn on their own without requiring human assistance in the form of a teacher. This helps to alleviate some of the frustrations student encounter when dealing with mathematic problem solving and is particularly important for online courses.

We leverage the capabilities of existing math software packages to analyze user input and to decide which feedback to generate. Thus, making the system a general math tool, where exercises may range form simple high school math to university level calculus and algebra.

For teachers, the system allows the use of a much wider variety of question types compared to standard e-learning systems, which are mostly restricted to simple multiple choice questions. It is easy for teachers to create their own exercises. Contrary to most tutoring systems no special programming skills are required. Exercises may be defined in a simple text format where formulas may be entered with the familiar LaTeX syntax.

REFERENCES

Aleven, V., McLaren, B., Sewall, J., and Koedinger, K. (2006). The cognitive tutor authoring tools (ctat): Preliminary evaluation of efficiency gains. In *ITS 2006*.

Beal, C., Beck, J., and Woolf, B. (1998). Impact of intelligent computer instruction on girls' math self concept and beliefs in the value of math. In *Annual meeting of the American Educational Research Association*.

Cheung, B., Hui, L., Zhang, J., and Yiu, S. (2003). Smarttutor: An intelligent tutoring system in web-based adult education. In *Journal of Systems and Software*, pages 11–25.

Collins, E. (2013). *Preliminary Summary SJSU+ augmented online learning environment pilot project*. http://www.sjsu.edu/chemistry/People/Faculty/Collins_Research_Page/AOLE%20Report%20-September%2010%202013%20final.pdf.

Fateman, R. J. (1972). Essays in algebraic simplification. In *PhD Thesis at the Massachusetts Institute of Technology*.

Gruber, J. (2004). *Markdown text-to-HTML conversion tool*. http://daringfireball.net/.

Koedinger, K. and Corbett, A. (2006). Cognitive tutors: Technology bringing learning science to the classroom. In *K. Sawyer (Ed.), The Cambridge Handbook of the Learning Sciences, Cambridge University Press*, pages 61–78.

Maplesoft (2013). *Maple - Technical computing software for engineers*. http://www.maplesoft.com/products/maple/.

MathWorks (2013). *MATLAB - The language of technical computing*. http://www.mathworks.com/products/matlab/.

Melis, E., Goguadze, G., Libbrecht, P., and Ullrich, C. (2009). Activemath - a learning platform with semantic web features. In *Ontologies and Semantic Web for e-Learning*.

Melis, E. and Siekmann, J. (2004). Activemath: An intelligent tutoring system for mathematics. In *International Conference on Artificial Intelligence and Soft Computing*.

Moodle (2013). *Moodle Learning Management System*. http://moodle.org.

OLAT (2013). *Online Learning and Training*. http://olat.org.

Schelter, W. (2013). *Maxima - Computer algebra system*. http://maxima.sourceforge.net.

Schiaffino, S., Garcia, P., and Amandi, A. (2008). eteacher: Providing personalized assistance to e-learning students. In *Computers and Education*, pages 1744–1754.

Thrun, S. (2013). *Update on our SJSU Plus Summer Pilot*. http://blog.udacity.com/2013/08/sebastian-thrun-update- on- our- sjsu-plus.html.