Journal on Mathematics Educations
Vol. xx, No. xx (20xx), pp. xx–xx.

INTERACTIVE LEARNING IN MATHEMATICS: CODING THEORY AND DISCRETE ALGEBRA

Putranto H. Utomo[1], Ruud Pellikaan[2], Jan Willem Knopper[3]

[1]**Eindhoven University of Technology and Sebelas Maret University, p.h.utomo@tue.nl**
[2]**Eindhoven University of Technology, g.r.pellikaan@tue.nl**
[3]**Eindhoven University of Technology, j.w.knopper@tue.nl**

Abstract. The new era of information transfer, that is the internet technologies could improve the learning of mathematics. For the course Coding Theory and the course Discrete Algebra, we made online exercises. We did this for several reasons. The first is to increase the productivity of the teacher. The second is to help students study individually between classes. We use the Mathdox system as the engine for the online exercises. It allows real time computation and has more advance mathematics expression compared to a standard e-learning system. The exercises given online are short questions that help the students understand the lecture. Based on the questionnaire following the course, many students have given positive responses about the online exercises.

Key words and Phrases: e-Learning, interactive learning, assessment system.

1. INTRODUCTION

We live in a very different era from 40-50 years ago. The new era of information transfer, that is the availability of internet, cloud computing, Web 2.0, and the semantic web could improve the learning of mathematics.

The internet makes it possible to transfer information around the world, without distances playing a role. The cloud computing is very helpful in doing computation online, so we do not have to carry out all the computations on the devices itself. The Web 2.0 is an interactive medium where one can upload and download personal information. The semantic web is a kind of algorithm to analyze the context and the metadata so it has meaning to the user. We experience this kind of technology in Facebook [1] and Google's sites [2] for example.

For the course Coding Theory and the course Discrete Algebra, we created online exercises. We did this for several reasons. The first is to increase the productivity time for the teacher. Since he or she does not have to do tedious grading

2000 Mathematics Subject Classification: 97C70
Received: dd-mm-yyyy, accepted: dd-mm-yyyy.

of simple exercises, he or she is free to help the students in other ways. The second is to help students study individually. In this case, the students can try and do the exercises and redo as much as they want. Moreover, they will receive immediate feedback, which is known to be more effective in learning [6]. After these exercises there are some optional traditional exercises students can make on paper. It is also necessary to practice more complicated exercises for these courses. Thirdly, starting the course the students can make an entrance test that gives an assessment whether the assumed foreknowledge is sufficient. Lastly, we also give a small credit (10% of the grade) for the online exercises to motivate students to do the exercises.

Since one of the aims is to help the student in self-study and give immediate feedback, we need to make the exercises more flexible in the sense that it will not have the same exact question for the second time. One way to do this is using parameters in the exercise, so each time the exercise is opened, it will give a different instance.

2. THE MATHDOX SYSTEM

There are systems like Moodle that offer parameterized questions for simple mathematics. Our subjects need more advanced mathematics, so we chose to use the MathDox system. The MathDox system has been developed at the Eindhoven University of Technology [3, 5, 7, 8]. The MathDox format combines several XML formats such as DocBook [16], OpenMath [15], XForms [13], Jelly [17], and Xinclude [14]. These formats then could provide the user an interactive mathematics document upon request via the MathDox Player.

Beside having a more advanced mathematics object, the MathDox document can also be translated into Latex and XHTML format, so we could easily have other version of the document if necessary. Further advantages we would like to mention here is that this system is easily extendable using web-service calls. Hence it allows a real-time computation powered by an external computer algebra system (CAS) such as Mathematica [10] and Maxima [11]. Moreover, the system itself allows us to easily add more exercises and correct mistakes.

In order to help the user to enter a mathematical expression, MathDox provides also a tool called MathDox Formula Editor, which could be seen in Figure 2. With this toolbox, a user could interactively choose a mathematical symbol and expression in the input, so the user does not have to learn how to write a mathematical expression in Latex or other format.

Since the MathDox system itself is not a Learning Management System (LMS), we integrated the MathDox exercises with the e-learning system Moodle [4] in the site OnCourse [9].

3. PROCEDURE AND EXAMPLES

We use MathDox mainly as an interactive exercise system. It is easy to create exercises for the MathDox system using a LaTeX format. An example could be found in [12], which is easy to learn. The exercises given online are short questions that help the students understand the lecture. In general there are two types of questions in the MathDox exercise, that is multiple choice and open answer questions.

In the multiple choice type of question, we assume that there is only one correct answer. Figure 1 gives an example of a multiple choice question, and the corresponding source code is given in Figure 3.

The answer of a student in an open answer questions is difficult to correct automatically, especially when the type of the answer is in the form of equations or a matrix. An example of an open answer question is given in Figure 2. In this example, the MathDox Formula Editor is used for entering the answer. Hence the student could enter the answer easily without worrying about the format. For this question, the source code is given in Figure 4.

Figure 5 give an example of question that uses CAS to compute the answer. In line 7 ofthis example, we asked a CAS to compute the dimension of the code, that is the length subtracted by the Euler function of the minimum distance.

Since MathDox has a very good compatibility with LaTeX type setting, we create exercises under an environment called *exercise-graph*. Hence, the exercise should begin with `\begin{exercise-graph}` and end with `\end{exercise-graph}`.

Now, we look in more detail into the source code in Figure 4. In line 2, using the `\score` command, we could define an arbitrary score for the exercise. The *first* environment located in lines 3-16 is used to define variables, including random variables. Following the *first* environment, there are *interaction* environments. The first one is where the question is defined, while the second and the third ones are responses of a correct and a wrong answer, respectively.

4. CONCLUDING REMARKS

In this section we will presents the responses on the online exercises.

In almost every course at the Eindhoven University of Technology, a questionnaire is given for the students following the final exam. Concerning the Oncourse questions for Coding Theory that counted for 10% of the end grade of the course the questionnaire had as item: "Are you satisfied with this part of the examination?" The students answered with an average of 3.8 on a scale between 1 and 5, where 1 means "certainly not" and 5 means "sure I do".

We also analyzed the time the students devoted to do the exercises. On average, each student spent 4 hours and 22 minutes to do all the exercises in the Coding Theory course, which consists of 5 tests, where each test has around 15 exercises. Most students who started with the first test also continued to practice during the

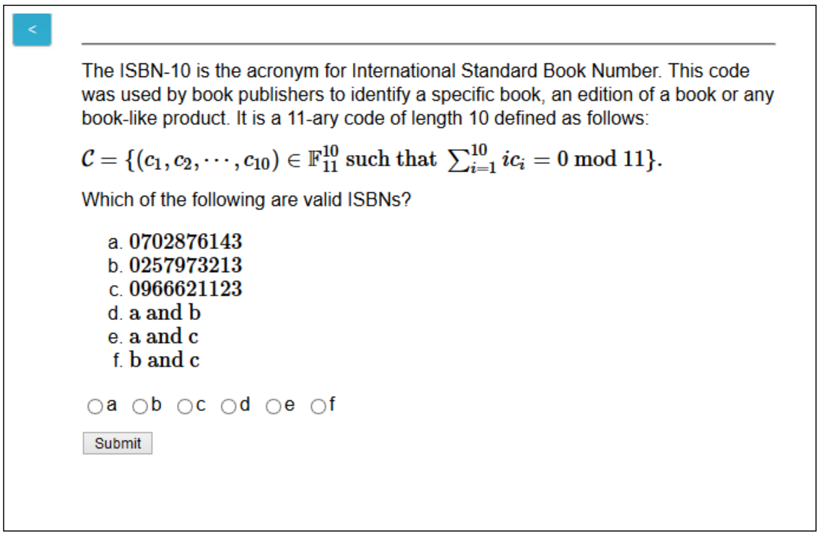

FIGURE 1. An example of a multiple choice question.

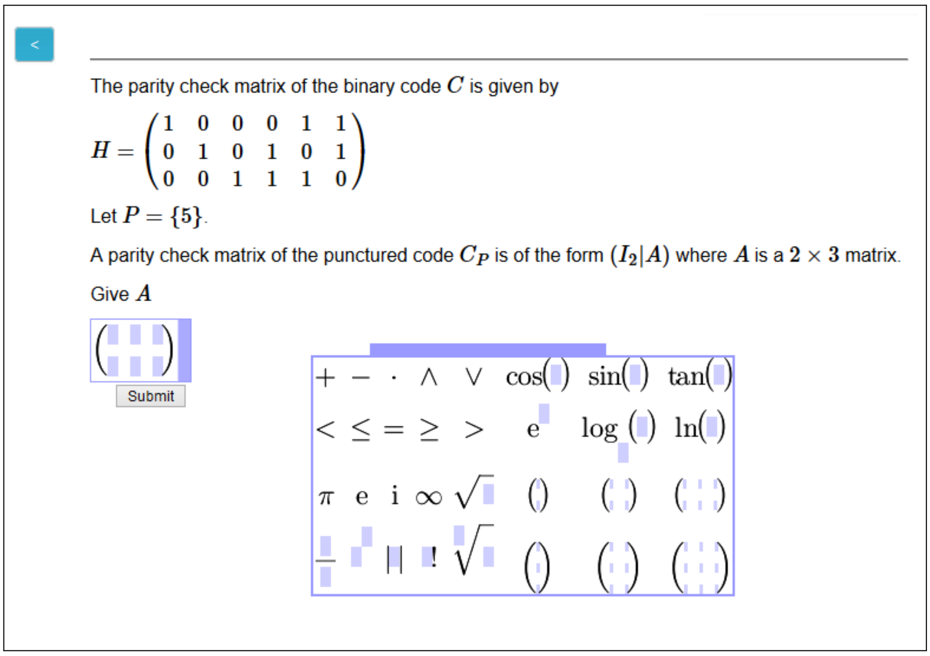

FIGURE 2. An example of open answer question.

whole course.

```
\begin{exercise-graph}
\score{100}
\begin{first}
  \setrandom{r}{1}{5}
  \setargument[evalpos]{choice_a}{[9608460476, 0198538030, 0702876143, 0077354761]}{\#r}
  \setargument[evalpos]{choice_b}{[0139541014, 0714322532, 0257973213, 0546743812]}{\#r}
  \setargument[evalpos]{choice_c}{[0702876143, 9608460476, 0966621123, 0198538030]}{\#r}
  \setargument[evalpos]{correct}{[\$e,\$a,\$c,\$b]}{\#r}
\end{first}
 \begin{interaction}{question}
   The ISBN-10 is the acronym for International Standard Book Number.
   This code was used by book publishers to identify a specific book,
   an edition of a book or any book-like product. It is a 11-ary code
   of length 10 defined as follows:\\
   $$ \mathcal{C} =\{ (c_1 ,c_2 ,\cdots,c_{10} )\in \mathbb{F} ^{10} _{11}
   \text{ such that } \sum  ^{10} _{i=1} ic_i = 0 \text{ mod } 11 \}. $$ \\
   Which of the following are valid ISBNs?
   \begin{scgroup}
     \begin{choice}{a}
       \out{choice_a}
       \target{wrong}
     \end{choice}
     \begin{choice}{b}
       \out{choice_b}
       \target{wrong}
     \end{choice}
     \begin{choice}{c}
       \out{choice_c}
       \target{wrong}
     \end{choice}
     \begin{choice}{d}
       \popcorn{"a and b "}
       \target{wrong}
     \end{choice}
     \begin{choice}{e}
       \popcorn{"a and c "}
       \target{correct}
     \end{choice}
     \begin{choice}{f}
       \popcorn{"b and c "}
       \target{wrong}
     \end{choice}
     \default{wrong}
   \end{scgroup}
 \end{interaction}{question}
 \begin{interaction}{wrong} \pen{50}
   This is not correct.
   \button{Try Again}{question}
 \end{interaction}{wrong}
 \begin{interaction}{correct} \correct
   This is indeed correct! You're score is: \out{score}.
 \end{interaction}{correct}
 \end{exercise-graph}
```

FIGURE 3. The source code for the question in Figure 1.

To conclude, the online exercises have benefits for both the teacher and the student.

```
\begin{exercise-graph}
  \score{100}
  \begin{first}
    \setrandom{r}{1}{6}
    \setargument[evalpos]{i}{[1,2,3,4,5]}{\#r}
    \setpc{parity}{matrix(matrixrow(1,0,0,0,1,1),
      matrixrow(0,1,0,1,0,1), matrixrow(0,0,1,1,1,0))}
    \setargument[evalpos]{ans}{[matrix(matrixrow(1,0,1),
      matrixrow(1,1,0)), matrix(matrixrow(0,1,1),  matrixrow(1,0,1)),
       matrix( matrixrow(0,1,1),  matrixrow(1,0,1)),
       matrix( matrixrow(0,1,1),  matrixrow(1,1,1)),
       matrix( matrixrow(1,1,1),  matrixrow(0,1,1))]}{\#r}
     \setpc{answer}{matrix(matrixrow(box,box,box), matrixrow(box,box,box))}
   \end{first}
 \begin{interaction}{question}
   The parity check  matrix of the binary code $C$ is given by
   $H=$\out{parity}. Let $P=\{ \out{i} \}$.
   A parity check  matrix of the punctured code $C_P$ is of  the form
   $(I_2|A)$ where $A$ is a $2 \times 3$ matrix. Give $A$
  \begin{answer-open}[pc]
     \begin{test-graph}{correct}
       \#answer=\#ans
     \end{test-graph}
     \default{wrong}
   \end{answer-open}
 \end{interaction}
 \begin{interaction}{correct} \correct
   That is indeed correct! Your score is: \out{score}.
 \end{interaction}
 \begin{interaction}{wrong} \pen{50}
   That is not correct.\\
   \button{Try again}{question}
 \end{interaction}
 \end{exercise-graph}
```

FIGURE 4. The source code of question in Figure 2.

```
\begin{exercise-graph}
  \score{100}
  \begin{first}
    \setrandom{r}{1}{7}
    \setargument[evalpos]{n}{[5,6,8,9,10,14]}{\#r}
    \setargument[evalpos]{d}{[5,3,4,3,5,7]}{\#r}
    \setpc[eval]{ans}{\#n - integer2.euler(\#d)}
  \end{first}
  \begin{interaction}{question}
     Let $C$ be the $F_q$-linear cyclic code of length \out{n}
     with generator polynomial the cyclotomic polynomial $\Phi _{\out{d}}(X)$.
     Give the dimension of $C$
       \begin{answer-open}[pc]
       \begin{test-graph}{correct}
         \#answer=\#ans
       \end{test-graph}
       \default{wrong}
     \end{answer-open}
   \end{interaction}
     \begin{interaction}{correct} \correct
     That is indeed correct! Your score is: \out{score}.
   \end{interaction}
     \begin{interaction}{wrong}
     That is not correct.\\
     \button{Try again}{question}
   \end{interaction}
   \end{exercise-graph}
```

FIGURE 5. The source code of question with CAS computation.

REFERENCES

[1] Facebook. `https://www.facebook.com`

[2] Google. `https://www.google.com`

[3] Mathdox. `http://www.mathdox.org`

[4] Moodle.`https://moodle.org`

[5] R. Verrijzer. Context in interactive mathematical documents : personalizing mathematics. PhD thesis. 2015. `http://repository.tue.nl/800756`

[6] B. Opitz, N. Ferdinand, A. Mecklinger. Timing Matters: The Impact of Immediate and Delayed Feedback on Artificial Language Learning. Frontiers in Human Neuroscience. 2011

[7] A. Cohen, H. Cuypers, J. Knopper, M. Spanbroek, and R. Verrizer. Mathdox: A system for interactive mathematics. In *Proceedings ED-MEDIA 2008, Vienna.*

[8] H. Cuypers, J. Knopper, and H. Sterk. MESS: the MathDox Exercise System. Electronic proceedings of the JEM-meeting, Aachen 2009.

[9] OnCourse. `https://oncourse.tue.nl`

[10] Mathematica. `https://www.wolfram.com/mathematica/`

[11] Maxima. `http://maxima.sourceforge.net`

[12] `http://evo01.win.tue.nl/mathadore/mathdox/latex/tutorial/exercise.xml`

[13] `https://www.w3.org/MarkUp/Forms/`

[14] `https://www.w3.org/TR/xinclude/`

[15] `https://www.openmath.org/`

[16] `http://docbook.org/`

[17] `http://commons.apache.org/proper/commons-jelly/`