Method to Create Multiple Choice Exercises for Computer Algebra System

Tatsuyoshi Hamada¹(✉) , Yoshiyuki Nakagawa² , and Makoto Tamura³

¹ Nihon University, Kameino, Fujisawa, Kanagawa 252-0880, Japan
hamada.tatsuyoshi@nihon-u.ac.jp

² Ryukoku University, Fukakusa, Fushimi-ku, Kyoto 612-8577, Japan
nakagawa@mail.ryukoku.ac.jp

³ Osaka Sangyo University, Nakagaito, Daito, Osaka 574-8530, Japan
mtamura@edo.osaka-sandai.ac.jp

Abstract. When studying mathematics, it is important to solve exercises of an appropriate level. Recently, web-based assessment systems with a computer algebra system (CAS), e.g., Moodle with Stack and the Möbius platform with Maple, have become popular. Such web-based systems are convenient; however, they have some problems relative to inputting and evaluating mathematical formulas. In addition, when considering and solving mathematical problems, handwriting mathematics is important. We want management system of paper-oriented exercises. Auto multiple choice (AMC), which was developed by Alexis Bienvenue, is open source software for creating and managing multiple choice questionnaires with automated marking. L^AT_EX is the native AMC language for questionnaire descriptions. We propose to combine AMC and CAS using LuaT_EX, which is a T_EX-based computer typesetting system with an embedded Lua scripting engine. We can embed CAS scripts into LuaT_EX source, and, by creating exercises with CAS, we can generate various problems with random coefficients or terms. By providing various patterns of practice problems and facilitating discussions with each student, we expect sufficient educational benefits of providing opportunities to communicate about mathematical concepts and algorithms among students.

Keywords: CAS · Multiple choice exercises · Calculus

1 Introduction

Exercises are important when learning mathematics, especially if we really want to understand something. We can find many exercises of various levels in textbooks. However, solving exercises in textbooks can be difficult for some students who are not good at calculations. Teachers must prepare and mark various levels of exercises for students. Feedback to students after evaluation should be given quickly; however, for many students, evaluation is very hard work for teachers.

This work was supported by KAKENHI 16K05139. The authors would like to thank Enago (www.enago.jp) for the English language review.

© Springer Nature Switzerland AG 2020

A. M. Bigatti et al. (Eds.): ICMS 2020, LNCS 12097, pp. 419–425, 2020.

https://doi.org/10.1007/978-3-030-52200-1_41

Recently, web-based assessment systems that use a computer algebra system (CAS), e.g., Moodle with Stack [1], which uses the Maxima [2] open source computer algebra system, and the Möbius platform [3] with Maple [4], have become popular. Creating exercises for web-based educational system has been investigated by Yoshitomi [5], where he attempted to generate question data for linear algebra. However, with web-based systems, there are problems related to inputting and evaluating mathematical formulas. Some studies have investigated mathematical input methods, e.g., [6–9]. However, there is no standard method for typical usage. In addition, handwritten work is important when considering and solving mathematical problems.

We would like to exploit the power of CAS to create mathematical exercises with auto multiple choice (AMC), which was developed by Alexis Bienvenüe [10]. AMC is open source software for creating and managing multiple choice questionnaires with automated marking. $\LaTeX$ is the native AMC language for questionnaire descriptions. The potential of AMC has been examined by many researchers and educators, e.g., [11] and [12]. Milana Lima dos Santos et al. presented results from the application of examination papers for engineering courses developed using Matlab/Octave scripts and the AMC, [13]. The first and second author presented the applications of AMC at the workshop in Japan, [19].

We use Lua $\TeX$ [14] for AMC. Lua $\TeX$ is an extended version of pdf $\TeX$ that uses Lua as an embedded scripting language. Lua [15] is a powerful and lightweight scripting language, that supports procedural programming and data descriptions. Using Lua’s “table” data structure, we can implement arrays, records, lists, queues and sets efficiently.

In addition, we can embed Maxima scripts into Lua $\TeX$ source. Maxima is the most famous general purpose open source computer algebra system (cf. [16]). By creating exercises using Maxima, we can generate various calculus problems with random coefficients or terms. By offering various practice problem pattern and facilitating discussions for each student, we expect sufficient educational effects of providing opportunities to communicate about mathematical concepts and algorithms among students.

2 AMC and CAS

AMC is a set of utilities that use of multiple choice questionnaires written in plain text or $\LaTeX$, as well automated correction and grading from scans of answer sheets using optical mark recognition. The following software packages are required to use AMC: $\LaTeX$, the ImageMagick image processing libraries, OpenCV, and Perl with Gtk2-Perl and Glade::XML for the graphical user interface. AMC has effective documentation, the developers of AMC operate a community support website and French and English user forums.

For users who are not ready to use $\LaTeX$, AMC includes a filter to process simple plain text files in AMC-TXT format, which is a type of markdown script. However, we embed CAS script in $\LaTeX$ code (we do not use AMC-TXT). AMC can create questions and answers in random order for each sheet, and we can distribute unique exercises for each student. Mathematical questions with randomized statements are supported by the original AMC with the fp

L^AT_EX package or LuaT_EX. However, these are for random coefficients of fixed or floating point numbers, (not rational numbers), and they do not support mathematical symbolic formulations. Thus we must create naturally symbolic formulated exercises with rational numbers. Initially, it may a little time to create problems, because of the incomprehensible error messages of L^AT_EX and CAS. However, we can recycle sets of questions easily. Using automated grading and creating answers, teacher's time is much saved. We found that students are less likely to give up when using the proposed approach, in fact, even after the lecture, many students didn't give up for solving problems.

In the following, we describe a simple example. In this source code, we changed the mark box to an oval using the `\AMCboxDimensions` L^AT_EX command, and the script of code acquisition was omitted. Code acquisition can be performed easily using the `\AMCcodeGridInt[options]{key}{n}` L^AT_EX command, e.g., to allow each student to enter her/his student id on an answer sheet.

The `\onecopy` L^AT_EX command produces as many distinct realizations of the test as desired (10 in our case).

```
\documentclass[a4paper]{article}
\usepackage[box,completemulti,lang=EN]{automultiplechoice}
\usepackage{luacode, tikz}
\newcommand*{\var}[1]{\luaexec{tex.print(#1)}}
\begin{document}
\AMCboxDimensions{shape=oval,width=1.8ex,height=2.5ex}
\AMCcodeVspace=0em
\luaexec{math.randomseed(20200713)}
\begin{luacode*}
function execMaxima(cmd)
  local texcmd="echo 'tex1(..cmd..)';'|maxima --very-quiet"
  local hdl=io.popen(texcmd, "r")
  local content=string.gsub(hdl:read("*all"), "\n", "")
  hdl:close()
  return content
end
\end{luacode*}
\onecopy{10}{
  %% start of the header
  % Code acquisition script
  %% end of the header
  %% start of the questions
  \begin{question}{1st_question}
  ...
  \end{question}
  \begin{question}{2nd_question}
  ...
  \end{question}
  ...
  %% end of the questions
}
\end{document}
```

We used Maxima in Lua $\TeX$ to create exercises. Lua $\TeX$ supports Lua as embedded scripting. The `io.popen()` function of Lua is used to execute command line arguments. Note that this function is system dependent and is not available on all platforms. We used MathLibre [17] or the Debian GNU/Linux 10.3 “buster” release. One of the authors (Y. Nakagawa) is currently checking Lua scripting under Microsoft Windows (with some modifications). Lua $\TeX$ supports the `\directlua` command, which can sometimes be tricky. When executing Lua code within $\TeX$ using the `\directlua` command, there is no easy way to use the percent character, and counting backslashes can be difficult. We used `luacode` environments and the `\luaexec` command from the `luacode` package [18]. We set the default $\LaTeX$ engine using “`lualatex --shell-escape`” in AMC preferences to execute CAS script¹.

First, we call the environment `luacode` in our $\TeX$ source code.

```
\usepackage{luacode}
```

The following $\TeX$ command is used to evaluate variables as $\TeX$ output.

```
\newcommand*{\var}[1]{\luaexec{tex.print(#1)}}
```

To fix the random seed to obtain the same results across different typesetting, we must set random seed after the `\begin{document}`.

```
\luaexec{math.randomseed(20200713)}
```

We define the new `execMaxima()` function in `luacode*`. In this script, `cmd` is the argument for an arbitrary Maxima script.

```
\begin{luacode*}
function execMaxima(cmd)
  local texcmd="echo 'tex1(..cmd..)';'|maxima --very-quiet"
  local hdl=io.popen(texcmd, "r")
  local content=string.gsub(hdl:read("*all"), "\n", "")
  hdl:close()
  return content
end
\end{luacode*}
```

We used Maxima in the article. However many of CASs support $\LaTeX$ formatting conventions; thus you can use your preferred CAS, e.g., Sage, Maple, or Mathematica.

```
texcmd="sage -c 'print(latex(..cmd..))'"
```

```
texcmd="echo 'latex(..cmd..)';'| maple -q"
```

```
texcmd="wolframscript -code 'TeXForm[..cmd..]'"
```

¹ Enabling “shell-escape” by default is dangerous, because it makes the $\LaTeX$ binary execute arbitrary shell commands in $\LaTeX$ files.

3 Example

The following example is a partial differentiation exercise. Here, coefficients a, b, s, t in this script are random numbers created using the Lua `math.random()` function. The string concatenation operator in Lua is denoted by two dots (`..`), the `exp` function represents an exponential function and `diff(expr, x)` returns the derivative of `expr` relative to variable x in Maxima.

```
\begin{question}{pdiff01}
\luaexec{
  a=math.random(2, 9);
  b=math.random(2, 9);
  s=(-1)^(math.random(0, 1));
  t=(-1)^(math.random(0, 1));
  g=s*a..'x+'..t*b..'y';
  g1=s*(a-1)..'x+'..t*b..'y';
  g2=s*a..'x+'..t*(b-1)..'y';
  f='exp('..g..)';
  f1='exp('..g1..)';
  f2='exp('..g2..)';
  formula=execMaxima(f);
  correct1=execMaxima('diff('..f..', x)');
  wrong1=execMaxima('diff('..f1..', x)');
  wrong2=execMaxima('diff('..f2..', x)');
  wrong3=execMaxima('diff('..f..', x)/'..a*s);
  wrong4=execMaxima('diff('..f1..', x)/'..a*s);
}

Find \((f_{x})\) where \((f(x, y)=\var{formula})\).

\begin{choiceshoriz}
  \correctchoice{\(\var{correct1}\)}
  \wrongchoice{\(\var{wrong1}\)}
  \wrongchoice{\(\var{wrong2}\)}
  \wrongchoice{\(\var{wrong3}\)}
  \wrongchoice{\(\var{wrong4}\)}
\end{choiceshoriz}
\end{question}
```

The result of upper script is here (Fig. 1).

1 Find f_x where $f(x, y) = e^{7x-4y}$.

☐ $\frac{6e^{6x-4y}}{7}$

☐ $7e^{7x-4y}$

☐ e^{7x-4y}

☐ $6e^{6x-4y}$

☐ $7e^{7x-3y}$

Fig. 1. A question for partial differentiation

4 Conclusions

This paper has proposed an automated approach to create several patterns of calculus exercises for different students at the undergraduate level. The primary advantages of the proposed approach are summarized as follows:

- Initially, it may take a little time to create problems using this system. However with random coefficients and the CAS, we can recycle sets of questions, easily.
- The proposed approach provides different types of problems for different student.
- The proposed approach facilitates discussions about different problems for each student, and it provides opportunities to communicate about mathematical concepts and algorithms.
- We found that students are less likely to give up when using the proposed approach.

We believe that automatically creating mathematical exercises at appropriate levels is significant challenge in mathematical software research.

The Scripts demonstrated in this paper are available at the GitHub public repository [20].

References

1. STACK—The University of Edinburgh. <https://www.ed.ac.uk/maths/stack/>. Accessed 27 Mar 2020
2. Maxima, A Computer Algebra System. <http://maxima.sourceforge.net/>. Accessed 27 Mar 2020
3. Möbius platform. <https://www.digitaled.com/platform>. Accessed 27 Mar 2020
4. Maple. <https://www.maplesoft.com/>. Accessed 27 Mar 2020
5. Yoshitomi, K.: Generation of abundant multi-choice or STACK type questions using CAS for random assignments. In: Davenport, J.H., Kauers, M., Labahn, G., Urban, J. (eds.) ICMS 2018. LNCS, vol. 10931, pp. 492–497. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96418-8_58
6. Shirai, S., et al.: Intelligent editor for authoring educational materials in mathematics e-learning systems. In: Davenport, J.H., Kauers, M., Labahn, G., Urban, J. (eds.) ICMS 2018. LNCS, vol. 10931, pp. 431–437. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96418-8_51
7. Fujimoto, M.: An implementation method of a CAS with a handwriting interface on tablet devices. In: Hong, H., Yap, C. (eds.) ICMS 2014. LNCS, vol. 8592, pp. 545–548. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44199-2_82
8. Su, W., Wang, P.S., Li, L.: A touch-based mathematical expression editor. In: Hong, H., Yap, C. (eds.) ICMS 2014. LNCS, vol. 8592, pp. 635–640. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44199-2_95
9. EquateIO. <https://equatio.texthelp.com/>. Accessed 27 Mar 2020
10. Auto Multiple Choice. <https://www.auto-multiple-choice.net/>. Accessed 27 Mar 2020

11. Siclet, C.: Teaching of mathematics thanks to multiple choice questionnaires. In: 27th EAEEIE Annual Conference (EAEEIE) (2017)
12. Clancy, I., Marcus-Quinn, A.: Exploring the possibilities of automated feedback for third level students/Esplorare le possibilità di feedback automatizzato per gli studenti universitari. *Form@re* **19**(3), 247–256 (2019). Gale Academic OneFile. Accessed 6 May 2020
13. dos Santos, M.L., Schmidt, H.P., Manassero, G., Pellini, E.L.: Automated design for engineering student examinations using Matlab/Octave scripts and the auto multiple choice package. In: 2019 IEEE World Conference on Engineering Education (EDUNINE), Lima, Peru, pp. 1–5 (2019)
14. LuaTeX. <http://www.luatex.org/>. Accessed 27 Mar 2020
15. Lua. <https://www.lua.org/>. Accessed 27 Mar 2020
16. Joyner, D.: OSCAS: maxima. *ACM Commun. Comput. Algebra* **40**(3/4), 108–111 (2006)
17. MathLibre. <http://www.mathlibre.org/>. Accessed 27 Mar 2020
18. Luacode. <https://ctan.org/pkg/luacode>. Accessed 27 Mar 2020
19. Hamada, T., Nakagawa, Y.: An application of CAS for creating multiple choice exercises. *Suuriken-Kokyuroku* **2142**, 108–116 (2019)
20. Hamada, T.: mc-calculus. <https://github.com/knxm/mc-calculus>. Accessed 8 May 2020