

# Exercise templates with Sage Math

Pedro Cruz<sup>1,2</sup>, Paula Oliveira<sup>1</sup>, Dina Seabra<sup>1,2</sup>

<sup>1</sup> Universidade de Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

<sup>2</sup> Center for Research and Development in Mathematics and Applications, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

E-mail: {pedrocruz,paula.oliveira,dfcs}@ua.pt

## Abstract

Students can benefit from the study of a detailed answer of an exercise after trying to solve it in traditional pen-and-paper way as has been verified in literature. This basic fact was the incentive to the creation of a new package, for the growing community of users of Sage Math, that allows an author to add, change and delete parameterized exercise templates, with detailed answer, from one or more personal databases, as well as producing new exercises in a booklet style or exam form.

*2000 Mathematics Subject Classification.* **97U50.** 97U30.

*Keywords.* exercise template; database; Sage Math.

## 1 Introduction

MEGUA [16] is a software package for Sage Math [10], a computer algebra system that works in Linux, virtual linux for Windows, and Web environment, allowing a working group to take advantage of the Sage Notebook sharing capabilities. With this package an author can create, edit and share his own database of parameterized exercise templates and an user can search, select and produce  $\text{\LaTeX}$  and PDF files from the exercises on a database. Given a database, a teacher can produce text (as a PDF document) for class work, exams or homeworks. We remark that computer assessment is not implemented since other projects at our university deal with it [11].

Some ideas are shared from other works. In [1] it is presented a work that uses web access to a database of calculus problems using  $\text{\LaTeX}$ . Project STACK [3, 4] is a system for computer aided assessment (CAA) and teaching system for mathematics using Maxima [12], where the authors had particular attention to the teacher's needs in authoring questions of their own. The 'matbook' project [9] uses TeX to read a database of math exercises for a special class of students as well as providing solutions and instructions for teachers. Using the R language [13], the system described in [2] permits the automatic generation of standardized statistical exams which is especially

**Tbilisi Mathematical Journal** ? (????), pp. 1–10.

National Centre for Science and Technology & Amsterdam University Press.

useful for large-scale exams. It's important to mention works like [8] where exercises are generated according to an algorithm and not by an author.

The authors of this paper teach first-years mathematics at University of Aveiro, in Portugal, to a broad spectrum of courses, like tourism, finance, engineering and sciences, each of them requiring different levels of mathematics on their curricula.

The university entrants for these courses undertook different studies in mathematics, and some of them elected not to undertake any mathematical studies at all in the final years of their secondary schooling. Usually the teaching method consists on presenting and explaining the material to be learned followed by solving specific application exercises. Assessment is again done with exercises, frequently of the "same type" or using generalizations from standard class exercises.

This context inspired us to produce a tool to improve the following issues:

- Students with different backgrounds: each course curricula is designed considering some mathematics prerequisites. However, students that apply for those courses do not always fulfill them. There are students with different backgrounds thus requiring some subjects to be presented in a different way: starting from simple concepts to more complex ones. Usually teachers suggest these students the study of previous topics on their own and there are also preparation courses available. This issue has been studied in several countries [7] where an analysis of the effect of "remedial programmes" is presented. The tool we are developing could help teachers to prepare a "remedial programme" to students with a defective mathematics preparation. These students usually need more practice of the same type of exercises. This is an advantage of parameterized exercises implemented in MEGUA: a student can practice until he feels comfortable, because generating different exercises is quite immediate.
- Detailed answers: we believe that presenting full, detailed answer of an exercise, not just a short summary or numerical solution, is of utmost importance for first years' students. This is in accordance to the study in [5] where we can read: "This article demonstrates the feasibility and effectiveness of teaching several mathematics skills by presenting students with carefully chosen sequences of worked-out examples and problems – without lectures or other direct instruction."
- Viewpoints and misconceptions: during a lecture a teacher may become aware of new difficulties preventing his students to understand what he is teaching, so he has to put up strategies to overcome these difficulties. The possibility of changing a mathematical text is in accordance, for example, with [6] that warns that "the presentation of

mathematics material (...) could contribute to creation and strengthening of students misconceptions”.

- Variability and concepts: a concept (property, theorem, application and so on) can be understood by a person but its application to a variety of situations must be practiced. From the same exercise model one can derive other instances to cover different aspects of the same problem.

We developed MEGUA, a software package for Sage Math computer algebra system, because Sage Math concentrates many mathematical packages on a single tool available through pre-parsed Python [14] language. Typesetting of mathematical notation is done with  $\text{\LaTeX}$ . Summarizing, to use MEGUA package an author needs knowledge of  $\text{\LaTeX}$  and basic Python programming. Sage Math functionalities can be easily found in documentation [15] for basic mathematical subjects.

The following listing is an example of an exercise template in  $\text{\LaTeX}$ :

---

```
%PROBLEM Some Name Here
What is the primitive of  $\text{\$ap} x + \text{\$bp@()}$  ?
%ANSWER
We can present several steps and result is
 $\text{\$prim+C}$  for  $\text{\$C} \text{\in} \text{\mathbb{R}}$ .
```

---

LISTING 1.1. Exercise template in  $\text{\LaTeX}$

This example has two parameters ‘ap’ and ‘bp’, this one decorated with ‘bp@()’ to produce ‘(bp)’ if value is negative. Then, using a Python declaration like

---

```
class E28E28_primdirect_001(Exercise):
    def make_random(s):
        s.ap = ZZ.random_element(-4,4)
        s.bp = ZZ.random_element(-4,4)
    def solve(s):
        x = SR.var('x')
        s.prim = integrate(s.ap * x + s.bp)
```

---

LISTING 1.2. Python for the Exercise template

An exercise has a name, like ‘E28E28\_primdirect\_001’, and two functions ‘make\_random’ and ‘solve’: the first creates a random exercise and the second evaluates values for the solving details. Sage specific functions are:

`ZZ.random_element`, `SR.var` and `integrate`, where the first generates integer random elements, the second creates a symbolic variable and the third integrates the function.

The following section presents more details about MEGUA design options and uses. We conclude with a brief summary of case studies and possible future directions.

## 2 MEGUA package

The following are MEGUA design implementation options:

- **Currently for pen-and-paper work**

We design this package to help the traditional pen-and-paper study by allowing a student to read detailed and complete answers to problems. It wasn't given priority to computer evaluation of a student answer.

- **Authors can create and edit one or more local databases**

One user can create several databases of exercises, for example, one for each subject he teaches. On the other hand, this separation allows an author to decide what he wants to share with colleagues.

- **Several output formats**

MEGUA is designed to export selected exercises to different templates. Currently, it produces PDF files (and the corresponding  $\text{\LaTeX}$  file) from a set of exercises, with or without solution, and an HTML index of database contents.

- **Classification**

Each exercise developed must be classified. It has a summary containing a description of the objectives and its mathematical classification (we are using MSC identifiers). Besides this, it has a label in format 'section; subsection; subsubsection' to be placed in a booklet as well as to be referenced in the index of the database it belongs. Finally, the exercise has a suggestive name (rather than an exercise number) so that students can easily remember it.

- **Designed for translations**

In MEGUA this is done in two ways. When an author produces an exercise, he is writing  $\text{\LaTeX}$  text and also producing Python code. In this way, the exercise text is separated from the 'programming' that determines how variables are replaced by values (or formulas). Thus, a translator has no difficulty translating text since he does not need to read programming code. The second strategy implemented is for the software package itself. There is a template directory for each natural language inside MEGUA. For example, there are  $\text{\LaTeX}$

templates ready for universal characters, like for instance portuguese accents. So, new languages could be easily added.

- **Operating System**

For the time being, stable versions of Sage Math run on a Linux machine (stand alone or server) using the web browser or the command line. There are virtualizations for running Sage Math in Windows machines. The MEGUA package is ready to work in a command line shell or with a web browser through Sage Notebook.

- **Team work**

Sage Notebook provides a natural way to work over the web for team work where users are able to share worksheets. We are using a worksheet for each exercise template for better organization. Using it via Linux command line one has means well known to programmers: we keep each text file as an exercise template, everything organized in directories and a versioning software ([18]) can be used to share work and keep records of changes.

## 2.1 Session example

We start by opening a new file (or new worksheet on notebook). It is necessary to declare the package

---

```
from megua.all import *
meg = MegBook('/home/pedro/calc-exercises.sqlite')
```

---

LISTING 1.3. MEGUA package declaration

The second line opens or creates a database 'calc-exercises.sqlite' at '/home/pedro' directory. Then, below in the file (or in another cell of the notebook) we declare the exercise:

---

```
txt = r'''
%SUMMARY Matrices; Types of Matrices
About matrix types and transposition.
%PROBLEM Type of a matrix
Which is the order of the matrix given by:
    $$ A = matrixA ? $$
Write down its transpose.
%ANSWER
Matrix $A$ has $nr$ rows and $nc$ columns so its order is $nr \times nc$.
Transposing a matrix consists on using rows from the first matrix as
columns in the second matrix, for instance, column 1 in $A$ will be
```

row 1 in the transpose of  $A$ . The transpose is a matrix with  $nr$  columns and  $nc$  rows, so its order is  $nc \times nr$

```
$$ matrixATranpose $$
```

```
class E12X34_matrices_001(Exercise):
    def make_random(s):
        s.nr = ZZ.random_element(2,5)
        s.nc = ZZ.random_element(2,5)
        s.matrixA = random_matrix(ZZ,s.nr,s.nc)
    def solve(s):
        s.matrixATranspose = s.someMatrix.transpose()
'''
meg.save(txt)
```

---

LISTING 1.4. Full template

The `txt=r'''` part defines a string in which we write all  $\text{\LaTeX}$  code and Python class. MEGUA parses this `txt` string and tests its content. Afterwards, if errors are not encountered, it is stored into the database using `save`.

In the example, the parameters used in the text part are ‘matrixA’, ‘matrixATranspose’, ‘nr’ and ‘nc’. These parameters are going to be replaced by values defined in ‘make\_random’ and ‘solve’. The advantage of separating parameters instantiation in two parts is that one can assign it’s own set of values to parameters, and for that set, request the evaluation of output values. This topic is described in more detail in the next section.

## 2.2 Parameters

As remarked in the previous section, in the exercise text one uses variables that are to be replaced by numeric values or formulas. These variables can be letters or letters and numbers, like ‘x11’ or ‘yslope’.

In this item added the functionality rules of transformation. For example, if the variables ‘name1’, ‘name2’ and ‘name3’ are defined as below:

---

```
name1 = -12.123456
name2 = -34.32
name3 = 1
```

---

LISTING 1.5. Variable assignment

then, this table shows transformation filters

parameter	after substitution	comment
<code>name1</code>	<code>-12.123456</code>	straight replacement;
<code>name1@()</code>	<code>(-12.123456)</code>	if negative put <code>(·)</code> around;
<code>name1@f(2.3g)</code>	<code>-12.1</code>	use of <i>printf</i> C language notation;
<code>name2@s{sin}</code>	<code>0.42857465435</code>	call <code>sin(name2)</code> ;
<code>name3@c{"text0","text1"}</code>	<code>text1</code>	choose one string of the list starting at 0.

TABLE 1. Decorators on parameters

### 2.3 Using exercises

MEGUA implements these possibilities:

- Produces a  $\text{\LaTeX}$  file for a single exercise instantiated with some random set of values. This can be done with ‘new’ command: `meg.new('Exercise_Key')`
- Produces a booklet from a selected set of exercises.
- Produces exams, even different exams for each student, with a selection of parameterized exercises.

All these tools can export  $\text{\LaTeX}$  to a file to be modified and compiled elsewhere.

### 2.4 Creating booklets or exams

Having a database of exercises allows us to select exercises and produce a  $\text{\LaTeX}$  document, using a template like Listing 1.6.

---

```

\documentclass{article}
\begin{document}
{{ put_here("E26B05_polynomial_001") }}
{{ put_here("E26B05_exponential_001") }}
{{ put_here("E26B05_powers_001") }}
{{ put_here("E26B05_logarithm_001") }}
{{ put_here("E26B05_rational_001") }}
\end{document}

```

---

LISTING 1.6. Booklet template

In this case five exercises will be written on a work file (worksheets, exams, homework, ...), using random values. What is exposed from each exercise can also be controlled: only the problem text, problem and answer and other details. It is also possible to specify particular values for each exercise.

### 3 Conclusion

Students can be engaged on homeworks if they know they will find the detailed answers later as a means to correct their mathematical reasoning. This mechanism was used in a few courses at University of Aveiro, and students appreciation was very positive. Some of them explicitly stated that it was a great help on their preparation for exams and during assessments improved answers were found.

We know that there are other software tools, like STACK [4], that have achieved this same results but, to our best knowledge, this is the first candidate to develop a tool for Sage Math that allows the creation of author database(s) of parameterized exercises using Sage Math directly in command line or notebook mode.

This tool also proved to be useful to teachers since exercises and their detailed answers are available online inside campus, thus, permitting a rapid preparation of study materials.

**Future work.** We consider new capabilities for the MEGUA package in several ways.  $\text{\LaTeX}$  is the current input format and PDF the output. We are considering to add variations of these two. For example, input could be prepared for MathJAX (a subset of  $\text{\LaTeX}$ ) and the output to be HTML. An advantage is that one can easily create copies of exercises to work in other markup or typesetting systems or adapt it to other scientific fields. Besides a detailed answer, one could add dialogs to check the student's knowledge before a detailed answer is presented. This could be done using Javascript or PHP on template text. We plan to embed MEGUA in Sage Notebook so that a student can call for an exercise inside a notebook worksheet and solve it as a homework or an exam. Also, some inclusion of a computer assessment system for mathematics and science in general (like [4, 17]) is future working direction.

### Acknowledgements

This work was partially supported by *FEDER* funds through *COMPETE*–Operational Programme Factors of Competitiveness (“Programa Operacional Factores de Competitividade”) and by Portuguese funds through the *Center for Research and Development in Mathematics and Applications* (University of Aveiro) and the Portuguese Foundation for Science and Technology (“FCT–Fundação para a Ciência e a Tecnologia”), within project PEst-C/MAT/UI4106/2011 with *COMPETE* number FCOMP-01-0124-FEDER-022690.

The authors would like to thank Mathematics Department for means to develop this package, the STIC (Services for Technologies of Information and Communication) for technical support and also to all Sage users and developers that have answered our questions.

## References

- [1] Kalle Saastamoinen, Jaakko Ketola, Tuukka Kurppa and Liisa Torikka. Enabling Web access to a database of calculus problems using  $\text{\LaTeX}$ , PHP and  $\text{\LaTeX2HTML}$ . *TUGboat Proceedings of the 2003 Annual Meeting*, 24(1), 2003.
- [2] B. Grn B and A. Zeileis. Automatic Generation of Exams in R. *Journal of Statistical Software*, 29(10):1–14, 2009.
- [3] C. J. Sangwin and M. J. Grove. STACK: addressing the needs of the “neglected learners”. *Proceedings of the WebAlt Conference*, 2005.
- [4] C. J. Sangwin. Assessing elementary algebra with STACK. *International Journal of Mathematical Education in Science and Technology*, 38(8):987–1002, 2007.
- [5] Xinming Zhu and Herbert A. Simon. Learning Mathematics From Examples and by Doing. *Cognition and Instruction*, 4(3):137–166, 1987.
- [6] Ann Kajaľndera and Miroslav Lovrić. Mathematics textbooks and their potential role in supporting misconceptions. *International Journal of Mathematical Education in Science and Technology*, 40(2):173–181, 2009.
- [7] Cristina Varsavskya. Chances of success in and engagement with mathematics for students who enter university with a weak mathematics background. *International Journal of Mathematical Education in Science and Technology*, 41(8):1037–1049, 2010.
- [8] Ana Paula Tomás, José Paulo Leal and Marcos Aurélio Domingues. A Web Application for Mathematics Education. *Conference ‘Advances in Web Based Learning - ICWL 2007’, United Kingdom. Lecture Notes in Computer Science 4823*, Springer-Verlag, 2008.
- [9] Pter Szab and Andrs Hrask. Managing a math exercise database with  $\text{\LaTeX}$ . *TUGboat, Proceedings of EuroTEX 2006*, 27(1), 2006.
- [10] W. A. Stein et al., *Sage Mathematics Software (Version 4.8.0)*, The Sage Development Team, 2012, <http://www.sagemath.org>.
- [11] Paula Oliveira and Sabrina Silva. An overview of PmatE: developing software for all degrees of teaching. *Proceedings of the International Conference in Mathematics Sciences and Sciences Education, June 11-14, University of Aveiro*, 2006.

- [12] The Maxima Group. Maxima, a Computer Algebra System. Version 5.18.1 (2009). <http://maxima.sourceforge.net/>
- [13] R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. 2011. <http://www.R-project.org>
- [14] Online Documentation of Python Language:  
<http://docs.python.org/>.
- [15] Online Documentation of Sage Math:  
<http://www.sagemath.org/help.html>.
- [16] Documentation: <http://megua.readthedocs.org>.  
Code: <http://code.google.com/p/megua/>
- [17] Moodle, version 1.9 (2007) <http://moodle.org>.
- [18] Versioning on wikipedia:  
[http://en.wikipedia.org/wiki/Software\\_versioning](http://en.wikipedia.org/wiki/Software_versioning)