Jörg Härterich (2019). Using randomized quizzes in undergraduate linear algebra and multivariable calculus. In: Contributions to the 1st International STACK conference 2018. Friedrich-Alexander-Universität Erlangen-Nürnberg: Fürth, Germany.

Using randomized quizzes in undergraduate linear algebra and multivariable calculus

Jörg Härterich[1]

Abstract: Using STACK (System for Teaching and Assessment using a Computer algebra Kernel) questions as computer-aided assessment (CAA) in the weekly homework assignments requires some extra work from the instructor to design good exercises compared with similar written tasks. This article describes some ways how to design and use randomized STACK quizzes in order to justify the extra effort. In addition, some data on the reception and use of the computer-aided assignments by the students is presented.

Keywords: computer-aided assessment; randomization; multiple attempts; undergraduate courses

1 Computer-aided assessments

Assignments and assessments are part of many undergraduate mathematics modules. Computer-aided tests and exams play an increasing role replacing in particular those questions that focus on procedural knowledge. Compared to mathematics lessons in schools which often closely follow one textbook with a large number of customized exercises, courses at universities vary considerably from lecturer to lecturer not only in style but also with respect to choice and emphasis of topics. Therefore the accompanying exercises typically are carefully chosen and precisely adjusted to the corresponding lecture. Moreover, since in each week considerably more topics are covered than in school with only a relatively small number of exercises, there is often only one exercise dealing with one subject matter. For this reason it requires some effort for lecturers to transform some or all of their exercises in CAA. In this article, I will describe from my subjective point of view some of the benefits that can be obtained by using computer-aided questions for the weekly assessment in particular by allowing for multiple attempts with intermediate feedback. I will also describe some of the experiences from turning old

[1] Ruhr-Universität Bochum, Fakultät für Mathematik, Universitätsstr.150, 44780 Bochum, Germany, joerg.haerterich@rub.de

exercises into randomized CAA with STACK and give some data about the use of the various test opportunities by first and second year students.

1.1 Turning written exercises into randomized CAA questions

Since several years STACK questions have partially replaced written homework in my courses on mathematics for physicists. This mandatory three-semester course covers single-variable calculus, linear algebra and multi-variable calculus. The curriculum is very close to the curriculum for math students with a focus on formal definitions and proofs. Students are expected to justify their calculations and should be able to come up with small proofs, so that students which complete this course with good success can later take more advanced mathematics courses.

Like many undergraduate service courses at German universities the courses at Ruhr University Bochum consist of two 90-minute lectures per week accompanied by a 90-minute exercise class as a preparation for the weekly homework exercises. Solutions are marked and count as bonus points toward the final grade.

As being able to argue mathematically is one of the most important learning objectives of the course I did not aim at replacing the complete weekly homework with computer-aided tests but decided to turn one of the four weekly exercises into a STACK exercise.

Since there were already a number of exercises available from previous years most STACK quizzes are variants of exercises that used to be written homework. Implementing these exercises as CAA questions takes much more preparation time than putting the same or a similar question onto an exercise sheet (again), so I will give some good reasons for this in the next section.

From the practical point of view not every exercise is equally well suited to be transformed into a randomized STACK question. It is therefore useful to run over the exercises in advance and classify them by assigning each exercise to one of the following groups:

1. Exercises that are not suitable for CAA. These are exercises where students have to argue, give a proof or have to interpret their results. Inapt are also exercises with many intermediate steps or those which are formulated so open that giving a particular answer form for the solution might already give too much away.
2. Exercises that can be converted into CAA in a straightforward way, either without randomization or with a randomization that produces very similar versions of the original question. Good randomization can nevertheless be a nontrivial matter.
3. Exercises that gain when they are turned into CAA questions. This includes questions where students are asked to give examples and the computer-algebra system then checks the mathematical properties but also problems where it is helpful to display additional graphics.

With respect to the RAT (Replacement-Amplification-Transformation) framework, see [HTS06], which describes the use of technology in learning with the aid of three categories replacement (technology just provides an electronic copy of the traditional teaching methods and material), amplification (technology improves the efficiency and productivity of the material without fundamental changes) and transformation (technology allows previously inconceivable forms of instruction and learning) one should aim for amplification instead of replacement.

1.2 Allowing for multiple attempts

Strictly speaking, assessments represent a method to determine what students know or can do at a particular moment. Usually, one distinguishes between two different types of assessment according to its purpose. While the goal of *formative* assessment consists of providing feedback that can be used by students to improve their learning, *summative* assessment is done at the end of an instructional unit and evaluates student learning of this unit.

In the traditional setting described above assessment is a mixture between formative and summative assessment. It is formative in the sense that all answers are marked and students get some feedback about their solutions. It is however also summative because points count towards the final grade and typically there is only one exercise per topic and no possibility to retry and improve. Unfortunately, students often only glance over the comments with a focus on the number of points they have received. If this number matches their expectation they do not have a closer look at the comments. This indicates that for most students the summative part seems to be more important than the formative one although considerable effort is put into the feedback.

One way to deal with this issue consists of allowing the students several but not an unlimited number of attempts for each problem. Since no person (neither instructor nor teaching assistant) is involved in the feedback and marking process it costs no extra effort to let students try the same type of exercise several times.

For most exercises three attempts has turned out to be a reasonable choice. The instructor not only avoids requests from students to erase their attempt because of some stupid misunderstanding or error if there is only one attempt, but students are also motivated to deal with the feedback in order to improve at the next attempt, see also the remarks in Greenhow [Gr15]. For an alternative approach to the same problem using intermediate steps, see Glasmachers & Kallweit [GK19] in this volume.

The following table shows the use of the STACK exercises in three different courses *Mathematics for physicists* at Ruhr University Bochum, Germany, from three different years (the first year of the respective courses is indicated in parantheses). The quiz questions were produced via the moodle activity "test" with the large majority of questions realized as STACK question type including randomization.

While this table shows the usual decline in student numbers over the first semesters it also gives some evidence how students use the STACK exercises. The opportunity to hand in solutions several times (usually three attempts are allowed) is used by many students and yields on average an improvement of roughly 30% when the best score is compared to the score in the first attempt. Around 70% of the students made complete use of the possibilities, i.e. they either achieved maximum score or used all of their attempts (or both...). Among those who did not use all their attempts even though an improvement would be possible there is a large group who hand in their first attempt very late and probably do not have time to work on a second solution.

Course (first year of studies)	**Average number of students**	**Average number of attempts**	**Relative improvement (last vs. first)**	**Complete use (maximum points or maximum number of attempts)**
Math 1 (2019)	86.00	1.82	30.7 %	67.2 %
Math 2 (2018)	60.25	1.75	30.0 %	75.1 %
Math 3 (2017)	48.25	2.30	56.8 %	72.7 %

Tab.1: Data on multiple attempts in three different courses

Students are allowed to discuss exercises (in fact there would be no way to prevent them from doing so) but the randomization rules out that only short answer, correct numbers or multiple choice answers are exchanged. Instead by dealing with different variants of the same problem one might even hope for discussions to occur on a more conceptual level, about similarities and differences between those variants, general strategies and a deeper understanding of the solution.

2 Practical tips for question authors

There are two main challenges faced by instructors in turning exercises into CAA: Incorporating randomization into the previously "static" exercises is not always straightforward and giving feedback in advance before knowing the students' answers requires some experience and knowledge of typical errors. In this note, I will only deal with the first aspect and only on a more general level. A list of concrete useful techniques for linear algebra problems can be found in Steele's guideline [St05].

2.1 Keeping the same difficulty level

Typically the goal of randomization consists in producing different looking questions of the same type and of roughly the same difficulty. In many situations this is not achieved by simply choosing some parameters from a given range. As an illustration consider the fraction problem

$$\frac{a}{6} + \frac{b}{5} = ?$$

which by choosing $a, b \in \{10, 11, 12, \dots, 19\}$ at random yields 100 different questions. However, the choice $a = 12, b = 15$ leads to a problem that is considerably easier than the first version above. So in this case one might be better off to let $a \in \{11,13,15,17,19\}$ only and $b \neq 15$. Fortunately, STACK possesses some nice randomization commands, see below.

As another example from linear algebra one might ask to put the equation for a plane given as

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + s \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + t \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \vec{a} + s \cdot \vec{u} + t \cdot \vec{v}$$

into parameter-free form. If one (very naively) just chooses all of the coefficients a_j, u_j and v_j at random e.g. from $\{-3, -2, -1, 0, 1, 2, 3\}$ several problems can arise: The equation might not even describe a plane if either of $\vec{u}$ or $\vec{v}$ is the zero vector or if both vectors are linearly dependent.

Even if the numbers are chosen in a way that defines a plane it could be more or less difficult to bring this equation into parameter-free form. Although it requires more effort from the question author one should consider a different way of randomization here either starting with the parameter-free equation and working backward to generate vectors $\vec{a}, \vec{u}$ and $\vec{v}$ in a controlled way or by using some restricted choice for the vectors $\vec{u}$ and $\vec{v}$. In general, the more advanced the problem mathematically is the more you have to ponder about randomization in a fair way.

2.2 Using STACK's randomization options

Compared to other computer assessment systems and also compared to MAXIMA, the underlying computer algebra system, STACK possesses a large variety of randomization commands. While the command `rand(n)` gives an integer chosen at random from $\{0, 1, 2, \dots, n-1\}$ which is the analogue of the MAXIMA command `random(n)` STACK also offers several other possibilities. Some of them like `rand(list)` where a member of a list is chosen at random are simply shortcuts for something that could be achieved with the usual MAXIMA random command, e.g.

```
n:rand(length(list))+1;
a:list[n];
```

Others however would be more complicated to build with the available MAXIMA commands. For instance, STACK allows to restrict the choice by using `rand_with_prohib(lower,upper,exlist)` to select a number between `lower` and `upper` excluding all values from `exlist`.

Another nice possibility is the selection of n elements from `list` with `rand_selection(list,n)`. This is an easy way to generate a list ofdifferent random entries, e.g. for the construction of a matrix with different eigenvalues. Even controlled repetitions are easily implemented as any number can be chosen by `rand_selection(list,n)` if it occurs multiple times in the list. In the above example one could use this to easily construct matrices with algebraically double but not triple eigenvalues.

2.3 Discouraging the use of Wolfram Alpha

One problem of CAA is that only the result is checked and the derivation is not taken into account. For standard problems like taking the inverse of a matrix this allows students to find the answer with software such as Wolfram Alpha with very little effort. Even if students know that they are expected to do the calculation by hand in the exam they are tempted to use the software. One way to make it a little more difficult consists of asking some particular intermediate results, e.g. when applying the Gauß-Jordan algorithm to find the inverse of a matrix one can also ask for an intermediate step where the original matrix has been brought into triangular form but is not yet a diagonal matrix. Of course, any mathematician will very quickly come up with ways to find these intermediate solutions with Wolfram Alpha, too, but for novices this is at least some obstacle.

2.4 Using (trivial) symmetries for randomization

There are topics where it is difficult to construct a large number of similar examples and where changing coefficients in a problem randomly does not work. In these situations it is sometimes still possible to use "mathematically trivial" symmetries to generate "different problems", either by interchanging variables or by changing some signs.

For example, looking for local extrema of

$$f(x,y,z) = x^3 - x^2y + 3yz - z^2$$

is more or less the same as looking for local extrema of

$$g(x,y,z) = xz^2 - 3xy + y^2 - z^3$$

but the similarity is for most students at least not obvious. Another trivial but often not obvious change in the task formulation can be made if a geometrical problems is invariant with respect to a shift.

2.5 Using MAXIMA to check answers

There are problems where it is difficult to anticipate which choice of random parameters will yield "sufficiently reasonable" answers, e.g. if the Gram Schmidt orthonormalization method is applied to three integer vectors the result can become quite intricate even for innocent looking vectors. Since the restrictions on MAXIMA commands within STACK do not allow to check results and redo the process if some conditions are violated, one might try to do this outside of STACK. To this end, the STACK code is modified by replacing the random choices by loops. In addition the MAXIMA code can be augmented in order to check whether the results satisfy some prescribed conditions.

As an example, the following MAXIMA code gives a list of all symmetric 3×3-matrices with integer entries between -2 and 2 and determinant 6:

```
list:[]$
for i1:-2 thru 2 do
 (M:matrix([0,0,0],[0,0,0],[0,0,0]), M[1,1]:i1,
  for i2:-1 thru 1 do (M[1,2]:i2, M[2,1]:i2,
    for i3:-1 thru 1 do (M[1,3]:i3, M[3,1]:i3,
      for i4:-1 thru 1 do (M[2,2]:i4,
        for i5:-1 thru 1 do ( M[2,3]:i5, M[3,2]:i5,
          for i6:-1 thru 1 do (M[3,3]:i6,
            if determinant(M)=6 then
                list:append(list,[M])))))))$
display(list);
```

The corresponding list can then be used to pick one of those matrices at random. Probably there exists a more theoretical ways to generate this list of matrices but one can easily think of more complicated problems. In addition, in many situation one might not bother to find *all* objects with certain properties but for the randomization a list of five, ten or hundred will be fine.

2.6 Sharing tested questions

Independent of the instructor's experience implementing exercises as CAA takes usually more effort than posing it as a written exercise. It is therefore highly desirable to exchange some STACK exercises with colleagues since not everybody needs to reinvent the wheel. Since the requirements in different courses vary, instructors will usually prefer licenses where modification of the exercises is explicitly allowed. There are a few repositories where STACK questions on various subjects can be found, e.g. the DOMAIN database [Do18] where exercises in different languages are provided with a CC-BY-SA license.

3 Adoption by students

After working with computer-aided assignments for several years some strategy for getting students to engage with the questions has become obvious. Students prefer clear patterns and simple rules, e.g. there is always exactly one computer-based question per week and the deadline is always on the same day and time.

If mistakes occur (LaTeX formulas not displaying correctly, correct answers marked as wrong, feedback not matching the answers,...) it is mandatory to react very quickly. Students tend to believe that quizzes provided electronically must be correct and sometimes spend frustratingly much time to get the answers right before they turn to the instructor.

The following diagram shows data from a student survey in the course *Mathematics for physicists 1* in winter term 2018/19. It shows that student satisfaction both with the number of computer-aided assessments as well as with the feedback is quite good.

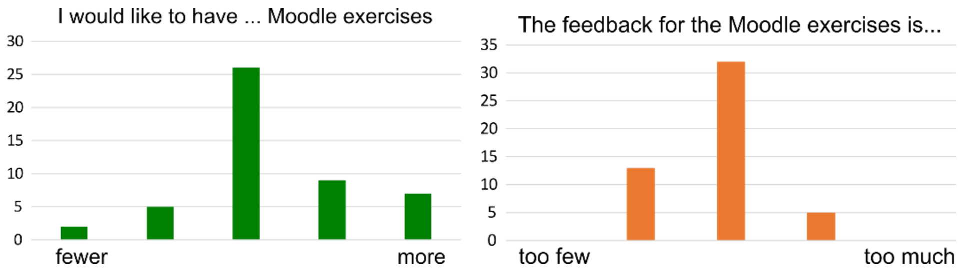

Fig.1: Results from a student survey (n=49)

The next diagram shows the percentage of students who made complete use (maximum score or maximum number of attempts) related to their score in the first attempt.

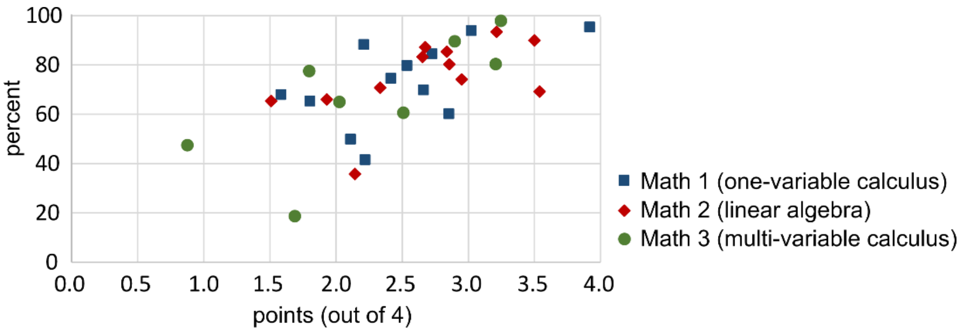

Fig.2: Use of multiple attempts vs. result of the first attempt

It is interesting to note that in all three courses a relatively weak first attempt does not encourage students to engage in more attempts even though there is more space for improvement. Rather, first scores which are close to the maximum score seem to motivate students to try again. One might therefore suspect that for more difficult questions additional measures like a more detailed feedback might increase engagement.

4 Concluding remarks

Transforming a set of written exercises into computer-aided assessments requires some additional work from the question author since randomization is not always trivial and a detailed feedback assumes good knowledge of typical errors. Experience backed by data from student surveys shows that this additional effort pays off if students are allowed to attempt the computer-aided assessments several times.

References

[Do18] DOMAIN, Database of Math Instructions, Retrieved March 1, 2019 from https://db.ak-mathe-digital.de.

[GR15] Greenhow, M.: Effective computer-aided assessment of mathematics: principles, practice and results. Teach. Math. Appl. 34, 117-137, 2015.

[GK19] Glasmachers, E.; Kallweit, M.: Adaptive Selbstlernaufgaben mit STACK. In: Contributions to the 1st International STACK conference 2018, 1-8, 2019.

[HTS06] Hughes, J.; Thomas, R.; Scharber, C.: Assessing Technology Integration: The RAT – Replacement, Amplification, and Transformation - Framework. In: Proc. SITE 2006 – Society for Information Technology & Teacher Education International Conference, 1616-1620, 2006.

[St05] Steele, J.D.: Setting Linear Algebra Problems, Preprint, University of New South Wales, 2005.