STACKification: automating assessments in tertiary mathematics

Ben Davies[1], Teresa Smart[2], Eirini Geraniou[2] and Cosette Crisan[2]

[1]University College London, Department of Mathematics, London, United Kingdom;
ben.m.j.davies@ucl.ac.uk;

[2]University College London, Institute of Education, London, United Kingdom;
teresa.smart@ucl.ac.uk; e.geraniou@ucl.ac.uk; c.crisan@ucl.ac.uk

In this paper, we report on four university lecturers' first-time experiences with computer-aided assessments. They were required to automate a significant proportion of the pre-existing weekly coursework for modules in first- or second-year undergraduate mathematics using STACK. We consider lecturers' perspectives on the role of computer-aided assessments in course design for undergraduate mathematics; the knowledge of technical aspects required to implement STACK-based assessments; and the perceived merits of automated assessment for different aspects of mathematical study. We conclude with a series of reflections upon our departmental practice and the process of enculturating mathematicians into the realm of automated assessment.

Keywords: Automated assessment, instructional design, mathematics coursework, thematic analysis.

Introduction.

In this paper, we focus on the introduction of STACK (a **S**ystem for **T**eaching and **A**ssessment using a **C**omputer algebra **K**ernal) to a Russell Group University in London. In particular, we set out to study a department-wide initiative where lecturers are expected to implement the majority of coursework using STACK. Students' weekly submission of handwritten solutions to problem sheets transitioned to the use of the STACK online environment which automatically assesses their answers and provides feedback.

The COVID-19 pandemic has dramatically increased the urgency and extent to which tertiary education has transitioned online. However, we understand this to be an acceleration of changes already underway in many parts of the tertiary sector. While we position our research as having general applications independent of the global health circumstances, we must acknowledge the environment in which this data was collected. Computer-Aided Assessment (CAA) has been on the agenda for the department from which we report for several years. However, the immediacy of the transition away from traditional handwritten assessments is, in large part, the result of the urgent need for remote, contactless instruction.

Given the urgency with which lecturers were required to automate their assessments, the default workflow for the majority of modules focused on the 'translation' or 'STACKification' of existing materials into CAAs. Some scholars may argue that this workflow is inherently flawed, and that effective CAAs should be generated in isolation, free from the restrictions of human graders (Sangwin, 2013). In the interests of space, we prefer to acknowledge the pragmatism of STACKification, and conjecture that many others using STACK for the first time are likely to follow a similar workflow. The process of STACKification warrants structured investigation, independent of scholarly arguments regarding the optimal origins of CAAs.

We report on semi-structured interviews with four lecturers and two postgraduate students, employed to support the design and implementation of STACK-based assessment across the department. All participants have been involved with the project for less than one year, and none had any prior experience with STACK (or any other CAA) prior to the project. The first author of this paper is also in the department and is responsible for co-leading the development of STACK-based resources.

Assessment in tertiary mathematics, and the increasing role of CAA.

Despite decades of innovation in assessment methods and tools, closed-book written examinations continue to dominate assessments for tertiary mathematics (Iannone & Simpson, 2011). Recent decades have seen an increase in the variety of assessment methods available to practitioners, but many of these innovations have struggled to gain popularity beyond the researcher communities in which they are developed. Researchers have highlighted the value of low-stakes formative assessments (Black & Wiliam, 2010), and called for greater assessment variety across undergraduate degrees. In this paper, we focus on Computer-Aided Assessment (CAA) and its role in a balanced 'assessment diet' (Iannnone and Simpson, 2011) alongside other modes including written and oral modes.

The last decade has seen significant growth in the availability of CAA technologies, from which STACK has emerged as a major player in the assessment of tertiary mathematics (Fahlgren et al., 2021). STACK uses a computer algebra system to evaluate students' responses against a wide array of mathematical properties. Unlike many of its predecessors that invoke little more than string matching or numerical equivalence, STACK uses a computer algebra system, based on open-source Maxima, to establish numeric and algebraic properties of students' answers. While STACK can be used for summative assessment, 'the actual potential lies in the possibilities for formative assessment; eliciting evidence of student understanding and providing feedback that moves learners forward' (Fahlgren, et al., p. 74). A detailed exposition of the affordance of STACK can be found in Sangwin (2013), and on stack-assessment.org. This software is currently 'used by universities, commercial [entities] and developers in over 15 countries' (www.stack-assessment.org, Sept 13, 2021) and can be integrated with a wide suite of Virtual Learning Environments including Moodle and ILIAS.

Recent developments with STACK have included a fully integrated online module in introductory university mathematics (Kinnear, 2019), and an exploration of task design for proof-based mathematics (Bickerton & Sangwin, 2021). Kinnear (2019) outlines an exemplary approach to embedding CAA in an introductory course for tertiary mathematics students. The author notes the time- and resource-intensive process required to fully integrate the technology, but from preliminary results, concludes that these investments were worthwhile for both instructor and student. Bickerton and Sangwin (2021), on the other hand, focused on higher level concepts associated with proof and argumentation. These authors provided a suite of design suggestions for proof comprehension tasks using STACK, including faded worked examples, reading comprehension activities and example generation tasks. Again, while time intensive to generate, such tasks appear to have the potential to contribute greatly to the varied assessment diet suggested by Iannone and Simpson (2011).

In this paper, we discuss the development of CAA in STACK by first-time users in one particular department of mathematics. While we did not set out to replicate Kinnear (2019), or to explicitly implement the design suggestions of Bickerton and Sangwin (2021), these works provide an important grounding against which to compare our own progress.

Aims, Research Questions and Methodology.

Consistent with the traditions of Design-based Research (Cobb et al., 2003), the aims of this research are two-fold; namely to develop our theoretical understanding of the assessments we design as a department, and to improve upon both our understanding of the design-process, and the assessment materials we offer our students in future iterations of the relevant modules.

Our research questions for the study reported in this paper are:

RQ1: What challenges are faced by first-time STACK users when implementing CAA assessments in tertiary mathematics?

RQ2: What are mathematicians' views and approaches to implementing CAA in tertiary mathematics?

Methods.

Participants.

Four lecturers (referred to as L1 – L4) and two postgraduate students (S1 and S2) participated in semi-structured interviews with two members of the research team (also the authors of this paper). S1 and S2 were members of a larger design team including two full-time faculty and six postgraduate students employed at different times throughout the year. Each lecturer was the leader of at least one undergraduate module and was responsible for overseeing the design of their own assessments. The extent to which lecturers engaged with the design team varied substantially.

Procedures and materials.

All interviews were conducted via Zoom, running between 35 and 45 minutes, and comprised two parts. First, participants were asked a series of questions about their experiences designing and implementing STACK-based assessments. The interviewers also asked about relationships between various members of the design team; the process of 'translating' existing items into CAAs in STACK; their level of satisfaction with their existing bank of STACK-based tasks; and what they would like to improve upon in future iterations of their STACK assessment. The second part of the interview was a stimulated reflection task. One week before their interview, participants were asked to select their favourite, and least favourite tasks to which they had contributed. Interviewers then asked a series of questions about each task, probing for information about the perceived strengths and weaknesses of CAA in general.

Data analysis.

In this first instance, a member of the research team watched each interview multiple times, tidying the imperfect automated transcripts in real-time. A series of latent themes were then identified, with supporting excerpts extracted iteratively through several passes through the data. A preliminary report was then produced, highlighting four themes with supporting excerpts and commentary for review by other members of the research team. This report forms the basis of the results section to follow.

Results.

Our thematic analysis (Braun & Clarke, 2006) identified three themes related to the design of STACK-based assessments by first-time users: *1) the process of STACKification, 2) technical challenges with coding in STACK,* and *3) the role of CAA in undergraduate math.*

Theme 1: The process for STACKification.

The course lecturer had for each of their modules a set of problem sheets that they used to provide homework and assessment tasks for their students. Despite variations in other aspects of their approach to CAA, all four lecturers adopted a surprisingly similar four-step workflow for translating their existing materials into automated assessments using STACK.

Phase one: Lecturers would parse their list of existing questions to identify which they believed would make suitable CAA items. This often involved identifying answers that required limited or simple input, and items that could be coded with relatively little technical expertise. L4 noted that "you cannot simply take an exercise sheet and immediately turn into STACK. It requires some effort [to identify appropriate items]". Only L3 focused on including items that were "most critical to capture the coverage of the material" when selecting items for CAA. All four lecturers worked largely independently on this phase, although two lecturers did consult the project leadership team for advice on which items were most suitable to automate.

In many cases, the mathematical content of existing questions would be preserved, but the response required from the student would be altered to suit the STACK environment. For example, with items from the Introductory Analysis course, the lecturer would choose a short series of proofs that were important for students to know and understand. Since STACK cannot currently facilitate the evaluation of student-produced proofs, the design team proposed a series of reading comprehension activities akin to those proposed by Bickerton and Sangwin (2021) that would still assess students' understanding of the proof in the absence of a 'prove that'-style task. In some cases, a series of multiple-choice items similar to those discussed in Mejía-Ramos et al. (2017) were also appropriate.

Phase two: In consultation with the design team, 'preSTACKed' documents were produced for most items. These were most frequently written in LaTeX, and resembled pseudocode outlining the design feature a future coder should implement. These included the types of inputs required from students, the scope and placement of random variables, and the specific question text to be shown to students. In some cases, this preSTACKing phase was a lot less structured, and simply comprised an itemized list of questions to be coded.

Phase three: These preSTACKed documents were then translated into functioning code. For three of the four lecturers, these preSTACKed documents were posted on a shared workflow tracker, to be picked up by the design team. By contrast, L1 did the majority of their own coding, consulting others only when "there was some finessing that I wasn't aware or didn't know how to do".

The design team collaborated frequently, checking each other's work, and coding additional question when the member responsible did not have time. This coding process worked well when the postgraduate student was familiar with the mathematical concepts and methods being developed in the module. However, S2 noted that "some of the hardest second year modules that I didn't take…I found hard, especially when the preSTACK document was vague [and] there was a lot of having to speak to the lecturer, [asking] how do you actually do this?”. We return to this back-and-forth dialogue between designers and lecturers later.

Phase four: After initial coding was completed, lecturers were invited to review each item and encouraged to check the code for the intend functionality. Given the inexperience of the design team, several items had early bugs. In some cases, variations on correct answers were marked incorrect (e.g. an answer such as 4/2 would be marked incorrect when the desired solution was the integer 2), and vice versa. As a tool, STACK gives tremendous control to the user regarding how to assess such variations and can facilitate the vast majority of desired responses in each case. However, given the inexperience of the coding team and the speed at which items needed to be produced, bugs of this nature were frequent in the early stages of the project and caused significant problems to lecturers and students.

Open communication between the lecturer and the coding team on checking how the STACK quizzes would be seen by the students was really important. L3 noted that “There were occasional things where the solutions that have been typed in weren't in the notation that I would teach and so I changed those. Little formatting things and a bit of debugging, so I would have a go at the questions and sometimes I came across errors and got them fixed before the students hit them, but other times, of course I didn't find them until the students found them and then we had to debug them live”.

We expect that these teething issues will reduce in future iterations of these modules. However, we note their significance here because of their impact on attitudes to the value of the technology, in particular with respect to (automated)-assessment, discussed later in this manuscript.

Theme 2: Challenges in early implementations of new STACK materials.

Lecturers tended to focus on assessing procedural tasks (in the sense of Sfard, 1991) in which a numeric or simple algebraic expression could be entered by the student. We, the research team, note that in theory, STACK has the capacity to implement a wide variety of question formats accessing a range of different understandings and approaches. However, anything beyond numeric or algebraic equivalence tests proved to be a significant challenge in many cases.

L1 noted that when the answer to the problem involved surds, STACK had no difficulty when the square root was in the numerator, but when it was in the denominator and the student rationalised the denominator, STACK “could not see that this was a correct answer”. Interestingly, this excerpt doesn’t draw a distinction between the capacity of the tool, and the capacity of a given implementation. While our data does not facilitate a more in-depth discussion on this point, we conjecture that this attitude may have been a barrier to higher quality design in some cases.

And L3 noted that when students were required to type in formulae “then one function of STACK that I hadn't really realized is, if you make one mistake, one small mistake [typing in a formula] which could be just a typo, it blanks all your answers”. S1 noted the importance of students needing to be shown how to input formulae correctly in STACK, for example how to input Greek letters such as lambda and theta, and how to input terms with subscripts such as x_0.

In contrast, L3 highlight one particularly successful episode, in which the coders initially "struggled because there's more than one right answer. So, in the end they worked out a really cunning way to work out whether the student's [solution] was correct". S1 also recognised the need for creativity with STACK, and appeared to understand that a solution should exist, even if it couldn't be implemented in this case. STACK recognises algebraic equivalence, but students can write the solution of a differential equation in many different ways and "you kind of have to think a bit more about all the all the possible answers that the students could give you".

L2 also highlighted problems associated with the inputting formulae: "One of the big challenges of STACK is making sure you get it right, because the system is only good as it is accurate, so if you have a mistake in your answer in STACK, then the whole, the whole thing is pointless".

Further professional development for lecturers and coders, and in some cases of students (as pointed out by L3) will seek to minimize the problems raised in this section. However, we note that even experienced coders have difficulties in this regard. To readers considering using STACK in the future, we recommend having a robust system of peer-review in place before AND after implementation with a student cohort.

Theme 3: The role of CAA in different content domains.

All four lecturers started with problem sheets that had been used as homework and assessment activities in their previous teaching. They felt that STACK could handle examples that required a numerical or simple algebraic answers but were reticent to explore opportunities to assess more conceptual aspects of their module curriculum using STACK. For example, S1 noted the ease of assessing calculus: "[it] was fairly straightforward because it involved fairly straightforward kind of mathematical methods so we had weekly quizzes for that". However, they asserted that the answers needed to be "well defined". L4 felt similarly, claiming that problems requiring students to input formulae can lead to difficulties "because formula can be written in slightly different ways and sometimes it doesn't recognize these things as the same".

While questions that required a numerical or algebraic answer could be easily STACKified in most cases, it was more difficult to test theoretical knowledge and proofs. L1 asserted that "when it comes to proofs one would use a normal Moodle (VLE) quiz and do some kind you know very smart multiple-choice type of question". Similarly, L2 claimed that "Not all [examples] were suitable because some of the questions involve some theorem or some proving which possibly could be STACKed or, if you like, but I couldn't see a way to do that, so I concentrated on questions with numerical answers".

Further, L4 questioned how a simple numerical or algebraic answer in STACK could show the students had understood the theory and methods they had been taught. "[In my course] it's not a matter of manipulating formula like in school, right. It's a matter of showing that you understand what's going on and it's somehow difficult to transform it into computer-based assessment". This lecturer went on to query the suitability of STACK "at a serious university... In a very good math department, you have to show that you understand, then you have to write, and explain". L4 did concede that "STACK is more suitable for an ancillary course [for non-math majors], but still, it's somehow lame even for chemists". In contrast, however, L2 felt that if you defined the question carefully then the student would have to understand the methods and the theory in order to get the right answer. This sentiment was also echoed by L3.

These excerpts suggest that the scope and merit of STACK vary greatly for different parts of the undergraduate maths curriculum. In this manuscript, we intentionally abstain from passing judgement on the commentary of L4 and others. Here, we prefer simply to report on the perspectives offered by our participants and reflect on ways in which we can improve our offering to students in future iterations of these courses. Bickerton and Sangwin (2021) propose a series of alternatives for STACK-based assessment of proof that seek to address many of the limitations raised by L4. We acknowledge that these alternatives are time consuming to implement and not applicable in all cases. However, we suspect that none of our four lecturers are aware of this recent work and intend to provide some professional development workshops in the future. In doing so, we seek to broaden the range of tasks offered to our students and the range of conceptual understanding accessed by our STACK-based assessments.

Discussion.

From our interviews with lecturers and postgraduate students, we identified three themes with consequences for the future development for the CAAs at our university and for the wider community planning to implement CAA for the first time.

First, we enumerate the process of ‘STACKification’, in which traditionally handwritten coursework tasks can be translated into CAAs using STACK. While the process has several possible refinements, it is interesting to note the relative uniformity with which this process was used by all four lecturers. Future iterations of these courses will involve cyclic redevelopments of many items, adding new features, resolving bugs, or adding more detailed feedback.

Second, we have identified a primary challenge for first-time users of STACK associated with evaluating algebraic equivalence in various forms. In several cases, lecturers and members of the design team were aware than an alternative coding solution should exist but could not execute a solution within the time constraints afforded. Again, these concerns will diminish with time, and as a department, we now have the opportunity to revisit those items that did not function as expected.

Finally, and perhaps most importantly, we considered lecturers’ perspectives on the role of CAA in tertiary mathematics more generally. All four lecturers acknowledged that STACK had the potential to assess at least some proportion of the undergraduate curriculum. However, these were heavily weighted toward applied mathematics, and to more procedural (rather than conceptual) tasks. Of particular note was L4’s belief in the inability to assess mathematical proof using STACK or other forms of CAA. It is unclear from the data available whether these perspectives would change with further professional development, focused on the potential for STACK to assess a wider array of question formats.

One final challenge not yet discussed lay in the design and implementation of feedback. The automation of personalised feedback proved to be a time intensive process, with most lecturers providing at most a correctness evaluation and a general solution for each question. We report on this feature of our data in more detail in future publications, focusing on routes for realising the potential for productive formative assessment discussed by Fahlgren et al. (2021).

Final remarks and future development.

At the time of writing, the STACK project at our university has been running for approximately 12 months. While we have now had a large bank of STACK-based items, the process of successfully integrating CAA into our curriculum is an on-going challenge. In the future, we will develop a greater variety of question forms, with a clearer focus on the learning outcomes for students and lecturers, and a more rigorous consideration of the formative and summative roles that these assessments play in our courses. This will feature a structured research programme intended to understand students' and lecturers' experiences with CAAs, and further iterations of the design-based research cycle we began with the data reported here.

Acknowledgment

The authors would like to thank the Department of Mathematics at UCL for their financial support of this research, and our colleagues who contributed their time and insight during the interview process.

References

Bickerton, R. T., & Sangwin, C. J. (2021). Practical Online Assessment of Mathematical Proof. *International Journal of Mathematical Education in Science and Technology*. https://doi.org/10.1080/0020739X.2021.1896813.

Black, P., & Wiliam, D. (2010). Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, *92*(1), 81–90. https://doi.org/10.1177/003172171009200119.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology.*, *3*(2), 77–101.

Cobb, P., DiSessa, D., Lehrer, R., & Schuable, L. (2003). Design experiments in educational research. *Educational Research*, *32*(1), 9–13. https://doi.org/10.3102/0013189X032001009.

Fahlgren, M., Brunström, M., Filling, F., Bjramheiður, K., Pinkernell, G., & Weigand, H. (2021). Technology-rich assessment in mathematics. In A. Clark-Wilson, A. Donevska-Todorova, E. Faggiano, J. Trgalova, & H. Weigand (Eds.), *Mathematics Education in the Digital Age* (1st ed., pp. 69–83). Routledge.

Iannone, P., & Simpson, A. (2011). The summative assessment diet: How we assess in mathematics degrees. *Teaching Mathematics and Its Applications*, *30*(4), 186–196. https://doi.org/10.1093/teamat/hrr017.

Kinnear, G. (2019). Delivering an online course using STACK. *Contributions to the 1st International STACK Conference 2018*. https://doi.org/10.5281/zenodo.2565969.

Mejía-Ramos, J. P., Lew, K., de la Torre, J., & Weber, K. (2017). Developing and validating proof comprehension tests in undergraduate mathematics. *Research in Mathematics Education, 19*(2), 130–146. https://doi.org/10.1080/14794802.2017.1325776.

Sangwin, C. J. (2013). *Computer-aided assessment of mathematics*. Oxford University Press.

Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational Studies in Mathematics*, *22*(1), 1–36. https://doi.org/10.1007/BF00302715.