

Using MatLab Grader for assessment analytics, automated grading and improving coding literacy in engineering students

Dr Laura Corner, written up by Dr Samuel Saunders

Summary

Students in any subject are often anxious about coding and using programming languages to perform specific functions. However, this is an essential skill in many engineering disciplines, particularly in research and data analysis. The beginners' course in MatLab was, however, largely lecture-based and did not give adequate provision for practicing and applying coding skills. A revised version of the module located in PC suites, and which opted to use MatLab Grader to administer assessment activities, was therefore designed and implemented. These assessments gave students the chance to apply their skills by directly writing code into the interface, and the system then automatically marked the submissions and provided the lecturer with extensive analytics around student performance.

Please briefly describe the activity undertaken for the case study

Students on the beginners' course in the MatLab programming language have, hitherto, been unable to practice their coding skills in an applied environment, necessitating a significant redesign of the course. A substantial part of this redesign constituted changes to the assessments, which have been reformatted to now require students to actively use their coding skills to perform specific tasks within the MatLab Grader environment. MatLab Grader is a Canvas-compatible application that provides students with a determined set of coding problems that they must use their coding knowledge to solve. Students write their code and test whether it performs the required function. At the end of the assessment, students are automatically told which of the problems they managed to accurately solve and which they did not, before they are allowed to retry the assessment as many times as it takes for them to successfully solve all of the problems. This generates a significant and useful set of data analytics that the lecturer can use to spot problem areas across large student cohorts. For example, the software records the number of attempts it takes the student to solve them all of the coding problems correctly, as well as which problems students got right and which they did not. The lecturer can then view and digest at their leisure after the assessments are complete.

How was the activity implemented?

The module was, in the first instance, moved from a lecture format into a practice-based format inside a PC suite. This allowed students the time and space to practically apply their newly-learned coding skills and to formatively develop their abilities. In terms of assessment, the MatLab Grader assessment initially (in AY 2021-22) took the form of an open-book examination where students completed the assessment in a 40-minute window that they themselves trigger by accessing the exercise in the Canvas space. However, this was negatively received by the students, who felt that 40 minutes was not long enough to allow for repeated attempts and that it caused them to rush through the assessment. It was also not particularly authentic; there are relatively few situations where a professional would be asked to complete the exercise in such a short time-scale with little to no access to external resources. Consequently, the decision was made to make the assessments take place over a 5-day window for AY 2022-23. Students could access the assessment in their own time and on a repeated basis, perhaps making one or two attempts before leaving and returning to it at another point over the 5-day period. This has been both better received by the students (no complaints were submitted, and a number of positive comments about the amount of time allocated to practicing code were also received), and a brief analysis of the analytics from both sets of students seems to show that neither methodology has a particular advantage to completing it. For example, 17% of the first cohort completed the assessment on their first attempt, while 22% of the second cohort completed it on their first attempt. However, there is a clear difference between those in the first and second cohorts who did not solve the problems *at all*; in the first cohort, 22% did not solve the assessment, while in the second only 7% did not solve it. How many attempts the student takes to solve all of the problems does not affect their overall mark.

Has this activity improved programme provision and student experience, and if so how?

This exercise has significantly improved the programme provision and the students' experience. Using MatLab code is an essential function in some engineering disciplines and has relevance to students' research activities when they complete their data analysis for their undergraduate dissertations and/or beyond. The practical application of skills also has the potential to develop students' digital fluency in using unfamiliar or previously-intimidating software, and also helps to improve their confidence in using esoteric applications to perform discipline-specific activities. For lecturers, the activity helps with workload, as the assessments are automatically-marked and fed back, and the provision of comprehensive analytics provides opportunities to target specific changes to the programme in future iterations of it, if there are particular heat-areas where a significant number of students struggled. It is also a more authentic form of assessment, in that it replicates activities that students may be doing in their professional lives upon graduation (in a multitude of disciplinary contexts).

Did you experience any challenges in implementation? If so, how did you overcome these?

The largest challenge to implementation was the format of the assessment itself. In AY 2021-22, the assessment was open for 40 minutes and, in essence, took the form of a timed examination. This was not particularly well received by students who felt that it was rushed, and it did not reflect a particularly authentic assessment experience. To remedy this, the assessment was redesigned into a 5-day take-home open book exam, where students are able to access it at will across the 5-day period. This comes with its own challenges; naturally students are able to 'cheat' by simply using wider resources to help them to solve the problem, however this is also 'authentic' in that there are few situations post-graduation where students will not have access to wider resources to help aid them, and it could be argued that this constitutes an appropriate part of the assessment process.

How does this case study relate to the hallmarks and attributes you have selected?

This case study touches on a number of Hallmarks and Attributes from the Liverpool Curriculum Framework. Again, it helps to foster both Digital Fluency and Confidence in students, in that they are applying specific and specialist skills in technological software in a discipline-specific way. This helps them to feel more connected to their curriculum and to their discipline, and reinforces, in essence, the message that they are capable of doing so. It is also an Authentic Assessment, in that asks students to apply their knowledge in a context that will emerge in their later professional careers, and also asks them to solve problems that they are likely to need to solve in future contexts (such as validating graphical data and its labels). Finally, it is also quite flexibly administered, which helps to make it Inclusive. Students are able to access the assessment in their own time and in any way that is comfortable to them over a significant period of time, and are able to solve the assessment using any resources that they feel will be necessary to help them.

How could this case study be transferred to other disciplines?

The format of this case study is relatively easy to transfer – take-home exams that can be opened over a window of time are relatively common, and can be applied in any number of disciplines using Canvas Quizzes. It is worth remembering the accessibility of online examinations and allowing students sufficient time to access the material. Canvas Quizzes allows for a significant number of question types, such as categorisation, essay, fill-in-the-blank, multiple choice, formula or true/false, among others (for further guidance on Canvas Quizzes, consult CIE's [DigiGuide: Recommended Approach to Online Exams in Canvas](#), dedicated to this topic). This overcomes the issue of specificity, in that this exercise focuses on MatLab as the language under assessment; Canvas Quizzes allow users to specify correct answers in any format, and so other languages (or non-

programming based assessments) can take this form. However, the analytics for Canvas Quizzes are slightly less comprehensive.

If someone else were to implement the activity in your case study, what advice would you give them?

Using analytics in assessments to identify problem-spots (or, indeed, spots of good practice) is slightly underdeveloped and should perhaps be redressed. Wherever assessment activities can yield useful data for programme amendment or enhancement, this opportunity should be taken as far as possible.



© 2023 by the University of Liverpool, Centre for Innovation in Education.

Using MatLab Grader for assessment analytics, automated grading and improving coding literacy in engineering students is made available under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).