33rd Annual Conference of the Australasian Association for Engineering Education (AAEE 2022)
4 - 7 December 2022 – Sydney, NSW

MATLAB Grader for Flexible Automated Assessment and Feedback in Large-Scale Engineering Subjects

Huey Yee Chan, Lionel Lam, and Raquel De Souza.
University of Melbourne
Corresponding Author Email: huey.chan@unimelb.edu.au

ABSTRACT

CONTEXT
One of the challenges in teaching large classes is the implementation of effective assessment strategies that deliver high quality, timely, and consistent feedback. This is particularly difficult in engineering subjects where assessment tasks involve programming concepts. It has previously been reported that automated assessment tools can be beneficial when there are large numbers of students. The introduction of automated assessment tools can significantly reduce the effort associated with manual marking, eliminate marking inconsistencies arising from having multiple graders, and most importantly, improve student self-learning by providing near immediate feedback.

PURPOSE
The objective of this project was to evaluate the implementation of MATLAB Grader as an automated assessment and feedback tool in three engineering subjects across different year levels. Various assessment strategies in the deployment of MATLAB Grader to maximise students learning and engagement will be explored in this paper, including the integration of MATLAB Grader into the University's current Learning Management System (LMS), Canvas.

APPROACH
Various deployment and grading strategies for MATLAB Grader were applied across three subjects: a first-year introductory engineering subject, a second-year biomedical engineering subject, and a Masters-level mechanical engineering subject. In the first-year subject, MATLAB programming is introduced as a basic problem-solving tool in the application of engineering concepts, while both the second-year and Masters-level subjects build on this foundation to explore more complex programming concepts and engineering applications. Student outcomes and responses were evaluated for MATLAB Grader assessments set up with unlimited attempts in the first and second-year subjects, versus those set up with limited attempts with pre-tests in the Masters-level subject.

OUTCOMES
Cohort-wide surveys on student experience with MATLAB Grader have yielded positive results in both the first and second-year subjects. Students generally appreciated the ease of access to assessment problems via the LMS and having unlimited attempts for their assessments. However, the quality of feedback provided has been identified as an area for improvement for both subjects. For the Masters-level subject, informal surveys and discussions with students also indicated feedback as a major area for improvement. The limited attempts with pre-tests strategy combined with minor technical issues have resulted in anxiety in a subset of the cohort when making submissions.

CONCLUSION
MATLAB Grader offers flexibility in automated assessment and feedback across different discipline-specific engineering subjects and various year levels. The deployment and effectiveness of the platform is highly dependent on the specific learning objectives and focus of a particular subject.

KEYWORDS
MATLAB Grader, automated assessment, feedback

Introduction

One of the challenges in teaching classes at scale is the implementation of effective assessment strategies that deliver high-quality, timely, and consistent feedback. This challenge is particularly difficult in engineering subjects where assessment tasks involve programming concepts. Here, Pears, et al. (2007) have found that automated assessment tools are beneficial when there are large numbers of students, making the manual marking of assignments unmanageable. Automated assessment tools can significantly reduce the effort associated with manual marking, eliminate marking inconsistencies arising from having multiple graders, and most importantly, improve student self-learning by providing near immediate feedback (Douce, Livingstone, & Orwell, 2005).

In recent years, there has been a significant shift towards automated grading using MATLAB Grader, a comprehensive all-in-one web-based environment for creating, executing, and assessing MATLAB coding assignments. Some examples include Boada & Vignoni (2021) who evaluated the implementation of MATLAB Grader assignments in their Mechatronics course, Smith (2019) who explored various strategies in the deployment of MATLAB Grader to maximise students learning and engagement in their first-year engineering subject, and Smith (2020) who integrated MATLAB Grader as an instructional tool to enhance learning in their Hybrid Vibrations course. However, these encouraging outcomes from the adoption of MATLAB Grader assessments across various engineering disciplines do not reflect its flexibility when implemented to target varying learning objectives in engineering subjects across multiple year levels.

This paper describes the implementation of MATLAB Grader as a teaching, assessment, and feedback tool in three subjects taught in the Faculty of Engineering and Information Technology (FEIT) at the University of Melbourne: a first-year introductory engineering subject (ENGR10004), a second-year biomedical engineering subject (BMEN20003), and a Masters-level mechanical engineering subject (MCEN90038). These subjects were chosen to showcase the flexibility of MATLAB Grader across a broad range of applications and year levels. ENGR10004 introduces students to programming by integrating MATLAB-based problem solving and engineering applications drawn from fluid mechanics, water treatment, and image processing. In BMEN20003, the engineering applications discussed are more motivational and contextual in nature: the subject focuses more heavily on building on the foundation laid by ENGR10004 to introduce more advanced programming concepts and MATLAB toolboxes. MCEN90038 is entirely application-based, and students are assumed to have sufficiently strong coding skills to focus on the implementation of engineering concepts without getting stuck learning programming syntax. Thus, MATLAB Grader is used more as a tool for teaching hands-on programming skills in ENGR10004 and BMEN20003, and more as a tool for automated large-scale assessment in MCEN90038.

Various assessment strategies in MATLAB Grader, including some described in Smith (2019) will be explored. The study will also evaluate the integration of MATLAB Grader into the University's current Learning Management System (LMS), Canvas. This is relevant as the LMS plays a crucial role in enhancing the learning experience by providing access to learning material and an avenue to deploy assessments as seamlessly as possible (Govender & Govender, 2010). The student experience with automated MATLAB Grader-based assessment was evaluated using user experience surveys.

Background

MATLAB Grader is a web-based platform for designing, deploying, and assessing MATLAB coding assignments. The platform provides real-time, automatically graded assessment for student work with features that offer educators the flexibility to design problems, apply grading rubric, and provide customised feedback for student attempts. The automated assessment and feedback features of MATLAB Grader make it a suitable and versatile tool to be applied across subjects across the different engineering disciplines and year levels.

ENGR10004 is a first-year undergraduate project-based subject with an average cohort size of 360 students. In this subject, students embark on a semester-long project as a team to design a water pumping, disinfection, and distribution system to supply drinking water sourced from an underground

well to a remote community. The project design involves modelling various subsystems relating to the engineering concepts of fluid mechanics, water treatment, and image processing. Within the subject, MATLAB is the adopted platform for modelling and implementing the relevant engineering concepts, and students develop introductory programming skills to achieve these objectives. MATLAB programming is taught during lectures and reinforced in weekly workshops to build towards its purpose in performing calculations, fine-tuning, and optimising the project design. As the primary assessment tool in the subject, MATLAB Grader is employed in two forms: (1) low-stakes assessment via regular practice exercises throughout the semester to help students build their programming skills through the application of the relevant engineering concepts, and (2) a one-off high-stakes End-of-Semester assessment to assess programming proficiency.

BMEN20003 is a second-year undergraduate subject with an average cohort size of 120 students. It is a core subject for students on the Biomedical Engineering track, and aims to advance students' programming skills from the foundational knowledge covered in ENGR10004. This includes the exploration of MATLAB's Symbolic Toolbox, random number generators and stochastic modelling, as well as the implementation of differential equation solvers in MATLAB. The subject has a secondary objective of introducing students to key concepts (mechanics, electromagnetism, probability and statistics, and systems biology) that will be encountered and expanded upon in downstream biomedical engineering subjects. That said, programming remains the primary focus of the subject – the engineering concepts are used more for context-setting, and to encourage students to appreciate the utility of programming across a diversity of biomedical engineering applications.

MCEN90038 is a Masters-level subject with an average cohort size of 200 students. In contrast to BMEN20003, this subject focuses entirely on engineering applications and not on the development of programming skills. Students learn about rigid-body dynamics and multi-rigid-body system dynamics in 3D, which involve lengthy calculations using linear algebra and vector-matrix operations at complexity levels that can sometimes be perceived to cloud the main intended learning outcomes. To alleviate this, MATLAB is employed as a computational tool for complex mathematical calculations, allowing students to focus their attention on the more important analytical concepts around system dynamics. As students in MCEN90038 are generally assumed to be quite proficient in MATLAB after having encountered it in undergraduate subjects such as ENGR10004, the applications of MATLAB in this subject are primarily focused on the use of advanced functionalities like the Symbolic Toolbox for problem solving, applied specifically to the modelling of rigid-body and multi-rigid-body dynamics. The use of MATLAB Grader in this subject is mainly designed to assess student understanding of the process behind the modelling of system dynamics; here, a certain level of analytical skills is expected but some flexibility around solution formulation is allowed.

Design and Implementation

ENGR10004

MATLAB Grader has been implemented in ENGR10004 since 2019 after the tool was renamed from Cody Coursework. The main motivation behind this initial adoption of MATLAB Grader was to utilise automated grading to reduce the amount of manual marking effort for the large cohort of over 750 students (pre-COVID). The MATLAB Grader assessments have since evolved into their current form with the realisation that the automated assessment platform offers many other advantages. In the most recent run of the subject (Semester 1 2022), the assessments consist of a suite of nine weekly homework assessments beginning from Week 3 and spanning the semester, and one End-of-Semester assessment in the final week. The homework assessments are made up of six MATLAB Grader problem sets and three online quizzes that cover a range of topics within the subject, progressing from fundamental knowledge of MATLAB programming to more complex applications in fluid mechanics, water treatment, and image processing. In general, the MATLAB Grader problem sets provide the avenue for programming practice for a particular topic and the online quizzes serve as the concluding exercise for each topic. Each MATLAB Grader problem set comprises between two to six problems, which are set up with unlimited attempts available over a period of 2 weeks. Final mark consolidation takes into consideration the best five of nine homework assessments to

make up 5% of the subject mark. Within this low-stakes assessment model, students have the flexibility of only attempting the minimum requirement without the pressure of penalties for not completing all nine homework assessments. Coupled with the unlimited attempts strategy that provides students the freedom to attempt problems as many times as necessary to get the correct answer, the homework assessments provide an effective avenue for MATLAB programming learning and practice, particularly for first-year students, most of whom are new to programming concepts.

The End-of-Semester assessment in the final week of semester is a larger scale assessment that contributes 10% to the subject mark. This set of seven MATLAB Grader problems is designed to be more challenging to assess students' ability to apply MATLAB as a computation, programming and problem-solving tool in various fluid mechanics, water treatment and image processing systems.

In addition to the above MATLAB Grader problems that are assessed, it is worth noting that an additional 150 or so unassessed practice-type problems across the various engineering topics within the subject are made available to students throughout the semester. This large repository of supplementary problems has provided an additional avenue for programming practice for students to attempt outside of the assessments. From past statistics, this extra resource is very popular in the weeks leading up to the high-stakes assessment at the end of semester, when students need practice to refresh their memory in MATLAB programming. Since the homework assessment problems are set up to be inaccessible outside of their respective 2-week assessment window, these supplementary practice problems serve as revision in preparation for the final assessment.

Being a large subject with a cohort size of over 300, additional measures had to be put in place to mitigate collusion among students as codes are not checked for similarity within MATLAB Grader. For the weekly homework assessments, this was achieved through a two-stage process of creating multiple similar versions of a problem and randomly assigning a version to a student. Examples of variation between different versions of a problem can include different parameter values, variable names, matrix sizes, mathematical expressions, and many others. While this method of collusion mitigation is not infallible, the chances of a student having the exact version of problems as their close contact(s) within each problem set is significantly reduced.

The End-of-Semester assessment, being a higher-stakes assessment lasting for 24 hours, goes a step further with a three-stage collusion mitigation process. First stage randomisation is achieved using p-codes embedded within MATLAB Grader problems to extract a single digit from the Student ID that a student manually inputs in the coding interface, which will then generate a specific set of parameters for use in the problem. In the second stage, multiple versions of the same problem are generated using p-codes with different student ID digit extraction methods to minimise the possibility of students with similar ID patterns having the same set of parameters. Finally, the third stage of randomisation assigns a different version of a problem to each student, similar to the process with the homework assessments described above.

BMEN20003

Inspired by the structure pioneered by ENGR10004, MATLAB Grader has been implemented in BMEN20003 since the subject ran for the first time in Semester 1 2021. MATLAB Grader is used in two of the major assessment components in this subject: weekly problem sets from Weeks 2 to 11 of the semester, and the Final Exam for the subject.

The weekly problem sets contribute 15% to the subject mark. Each set consists of five MATLAB Grader problems that aim to consolidate students' understanding of the programming concepts covered in the previous week of lectures. In consolidating students' final marks, the best five out of ten sets are considered. Like in ENGR10004, this has been done to offer students flexibility and to ensure that the weekly tasks are treated more as formative practice problems instead of a burden to be completed to avoid penalties. In line with this, as well as to allow students to develop their troubleshooting and debugging skills, all problems have been set up to allow for unlimited attempts.

The Final Exam is a high-stakes assessment that contributes 40% to the subject mark; it also has a hurdle requirement attached, which means that students must pass the exam to pass the subject. In

the most recent run of the subject, the exam consisted of seven MATLAB Grader problems, designed to be significantly more challenging than the weekly problem sets (similar in difficulty level to the longer form non-MATLAB Grader-based assignments that also feature in the subject). Because the cohort was significantly smaller than ENGR10004 as well as the tight exam time limit (2 hours 45 minutes), randomisation was seen as less necessary and was not implemented. Again, as troubleshooting and debugging skills are a key learning outcome in this subject, all exam problems were set up to allow for unlimited attempts.

MCEN90038

MATLAB Grader was implemented for the first time in this subject in Semester 1 2022. There are four formative assessments in MCEN90038 spread out across the semester; in total they make up 40% of the overall subject grade. The assessments are due in weeks 4, 7,10 and 11, and students have 2 weeks to complete each one. MATLAB Grader, is a partial assessment component within each of the four formative assessments that makes up a total of 25% of the overall subject grade. The MATLAB Grader problems assess students' proficiency in applying MATLAB programming to partially or fully analyse the dynamics of a system, starting with the principles of kinematics and leading on to dynamics, with increasing complexity over each assessment. For comparison, two strategies of implementation were chosen: unlimited attempts versus limited attempts with pre-tests. In the unlimited attempts version, all tests can be run an unlimited number of times. In the limited attempts version, only tests marked as pre-tests can be run unlimited, the remaining tests can only be checked a limited number of times. With the pre-tests in MCEN90038 problems, students will only receive feedback on initial modelling and selected intermediate solutions, but not the final solution(s). Although different in essence, both strategies still allow for troubleshooting of initial and intermediate steps prior to final submission. The first two assignments were deployed with unlimited attempts to ease students into MATLAB Grader and the final two assignments were deployed with a maximum of 3 attempts, with pre-tests. The intention behind the limited attempts with pre-tests strategy was to provide sufficient guidance to allow students to correct their mistakes without pinpointing the exact source of the error. This therefore allows some flexibility in the modelling to encourage deeper and more diverse analytical thinking.

LMS Integration

MATLAB Grader integration with Canvas was employed in all three subjects, where MATLAB Grader problems were set up as assignments via the Learning Tools Interoperability (LTI) interface. This integration is useful because it offers the convenience of allowing all assessment processes to occur within a single seamless interface. Students access the MATLAB Grader assignments directly from within Canvas and marks from automated grading are automatically updated in the Canvas Gradebook. This means that there is no need for students to exit the LMS to access an external web platform; there is also no need for educators to go through the extra step of importing marks manually at the end of each assessment period. By default, MATLAB Grader assignments are deployed with unlimited attempts during the set-up process, but there is the option to limit the number of attempts within the LTI for situations where this strategy is unsuitable, as previously described for MCEN90038. An additional advantage that Canvas integration offers is the Sections functionality that allows for the random allocation of different versions of MATLAB Grader problems to pre-assigned groups of students. Without this feature, the collusion mitigation measures previously described for ENGR10004 would not have been possible.

Outcomes and Discussion

Subject-wide surveys were conducted in ENGR10004 and BMEN20003 in Semester 1 2022 to gather data on student experience with MATLAB Grader. There were 42 respondents from ENGR10004 and 16 from BMEN20003. Survey questions were designed around the following five general categories with the aim to understand the student experience with the assessment platform:

1. Prior knowledge or experience in online programming assessments

2. Experience accessing and using the MATLAB Grader assessment platform
3. Quality and presentation of assessment questions
4. Quality of automated feedback provided
5. Overall practicality and effectiveness of the implementation of MATLAB Grader

In the first category, information on students' prior knowledge was gathered based on their self-perceived level of programming competency, from "None", "Minimal", "Introductory", "Good" to "Advanced". Responses to survey questions from the subsequent categories 2 to 5, focusing on user experience, were scored on a 5-point Likert scale, from 0 representing "Strongly Disagree" to 5 representing "Strongly Agree".

The survey outcomes from Category 1, as shown in Figure 1, revealed that most students in ENGR10004 (over 80%) had little to no programming experience prior to enrolling in the subject. This observation was not unexpected since this is a first-year entry-level programming subject on the engineering pathway. A large proportion of the BMEN20003 cohort have completed ENGR10004, so a shift in prior programming experience was observed, with a significantly higher percentage of students rating themselves as "Good". Due to the flexibility of the Melbourne Model, a significant proportion of this cohort also consists of lateral entry Masters students, meaning that they are entering the Master of Engineering from non-traditional pathways or from other universities. These students must complete certain foundational undergraduate subjects (such as BMEN20003 here) before enrolling in their Masters-level subjects. This group of students explains the presence of students in the BMEN20003 data set indicating that they have little to no programming experience.

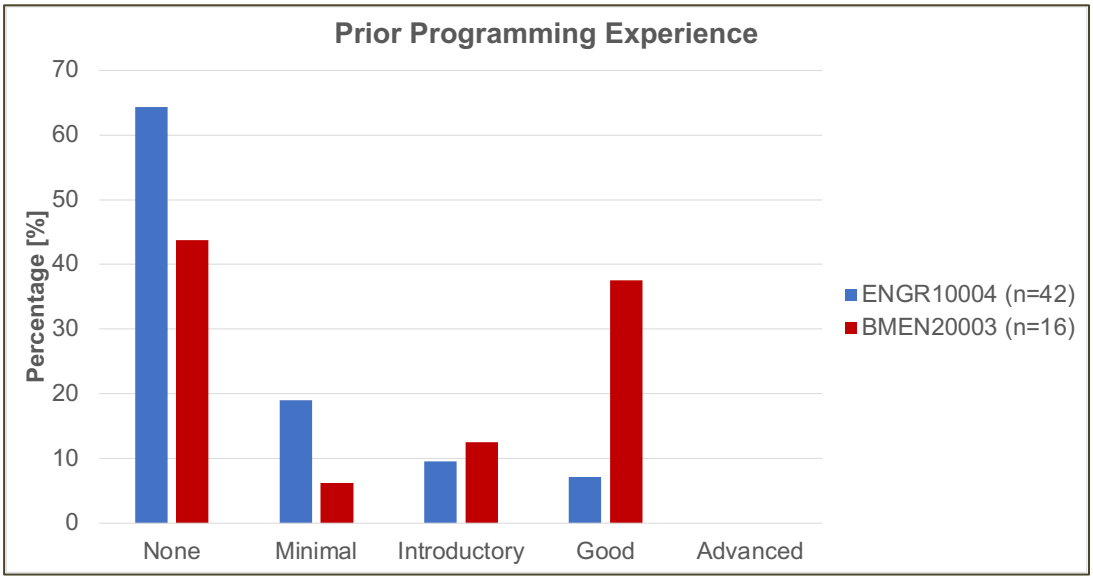

Figure 1: Graph representing student prior programming experience in ENGR10004 and BMEN20003.

Outcomes from the subsequent four categories on user experience, shown in Table 1, indicated that students had an overall positive experience with MATLAB Grader as the assessment platform. Based on the high ratings of above 4 from Category 2 for both subjects, it was encouraging that students did not face challenges accessing and using the MATLAB Grader interface as responses revealed that the assessment platform mostly ran without technical issues. The user experience coverage within this category was interpreted in two ways: firstly, the experience accessing MATLAB Grader problems from the LMS and secondly, using the MATLAB Grader interface for coding and submission for automated assessment. For the former, results indicated that MATLAB Grader LMS integration was effective in providing reliable and seamless access from within the LMS to the external assessment platform. As for the latter, despite an overall positive response, free-form comments indicated some underlying frustration with the perceived inflexibility of MATLAB Grader in BMEN20003. In BMEN20003, students are introduced to more advanced user output-based

functions beyond just the disp function. These include fprintf and sprint functions, along with the customisation of various more advanced visualisation-based functions (plot, quiver, and streamline, to name a few). Students are also introduced to many different data classes in MATLAB, and how to interconvert between them. Testing for the outputs of these functions require an exact match – for example, strings must match exactly with what is requested, down to spaces and periods, for answers to be marked as correct, which contributed to student frustration. Reflecting on this, perhaps this should not really be considered a technical drawback of MATLAB Grader, but should be used as a teaching moment: most students in BMEN20003 are growing into adept programmers and may be dealing for the first time with exactly how strictly logical programming languages can be. Despite this minor technical drawback, MATLAB Grader has provided an overall reliable and robust assessment platform in both ENGR10004 and BMEN20003, which is essential to provide students with confidence in their assessment, reducing the stress and anxiety of technical issues arising from the automated submission and marking process.

From Category 3, again ratings of above 4 showed that, overall, students in both ENGR10004 and BMEN20003 found the assessment questions well-structured and contained content that was relevant to their learning. This might be attributed to the immense effort that the subject team has invested over several rounds of implementation to restructure and fine-tune the MATLAB Grader assignment questions to best suit the learning objectives. There were however a minority in ENGR10004 who commented that the mark allocations for tests within certain problem sets could have been better apportioned, for example to take into consideration consequential errors in lengthy problems that may result in large penalties downstream.

From responses to the first subcategory within Category 5, students were in overwhelming agreement that the unlimited attempts option in the assessments was most helpful to their learning. While this gave students confidence to complete the work themselves through investigation and trial and error, there was concern that some students may abuse the system by merely using random guess-and-check methods to achieve marks without accountability, as discussed by Smith (2019). While we acknowledge that these concerns remain, we believe that these are outweighed by the clear advantage of enabling students to develop their programming, troubleshooting, and debugging skills through making mistakes and more importantly, learning from them.

Within the second subcategory of Category 5, it was encouraging to see that the collusion mitigation measures deployed in ENGR10004 were deemed by students to be effective in minimising collusion. While collusion mitigation could not be quantified from an assessor's perspective, students' positive outlook from the surveys has provided the subject team with the confidence to continue with similar collusion mitigation strategies in future implementations. Collusion was not a cause for concern with the smaller cohort in BMEN20003, as confirmed by the outcomes in Table 1.

One aspect that did not yield quite as positive an outcome was the quality of automated feedback from the assessments in Category 4. The versatility of incorporating feedback in the automated testing of MATLAB Grader assessments is of importance in both ENGR10004 and BMEN20003 as MATLAB Grader allows customised feedback for each test in addition to the default output and feedback/error messages. As ENGR10004 was the subject that pioneered MATLAB Grader, the feedback in the assessment problems has been carefully thought out and refined over the years. The quality of feedback in the current version of MATLAB Grader problems have greatly improved from the initial rounds of implementation where they were riddled with inconsistencies and inaccuracies, resulting in complaints from students for almost every homework assessment. Despite this progress and evolution over the years, the survey outcomes indicated that students still felt that feedback from the assessment problems lacked information that was effective in helping them identify their mistakes and correct them. Effective feedback is arguably subjective as it is dependent on many factors, for example the nature of an assessment question or a student's level of understanding. This aspect will continue to be a work in progress for ENGR10004 as an in-depth study and review into every assessment question will be required to identify the type and amount of feedback that is appropriate according to the relevant expectation of prior knowledge. Like ENGR10004, feedback was also a major area for future improvement for BMEN20003. Some of this is related to the perceived inflexibility of the tests, as previously described. In BMEN20003, students

who are also getting more comfortable and proficient with programming may decide to approach problems in ways different from that intended by the assessor. It is extremely challenging to write tests that account for all possibilities, particularly for problems involving more advanced programming (and engineering) concepts. One way to remedy this might be to further discretise the tests for such questions, and to design specific tests that function as checkpoints or signposts that nudge students down the intended solution pathway.

Table 1: User experience survey outcomes for ENGR10004 and BMEN20003.

Survey category/subcategories		**ENGR10004**	**BMEN20003**
2. Experience accessing and using the MATLAB Grader platform		4.30	4.40
3. Quality and presentation of assessment questions		4.26	4.35
4. Quality of automated feedback provided		3.93	3.67
5. Overall practicality and effectiveness of the implementation of MATLAB Grader	The unlimited attempts option was helpful	4.57	4.88
	Collusion was not an issue	4.29	4.25
	MATLAB Grader has helped overall understanding	4.05	4.31

In summary, the responses from the final subcategory of Category 5 suggest that MATLAB Grader has proven to be an effective teaching and automated assessment tool that has helped students' understanding in the first and second-year subjects where the learning focus is on building and assessing programming skills in engineering applications. The survey outcomes were helpful in identifying feedback as an area for improvement in moving forward with future implementations.

The Masters-level subject, MCEN90038, differs from the first and second-year subjects in the sense that students are assumed to be entering the subject with enough programming experience, such that assessments can focus more on in-depth application of engineering concepts rather than on programming skills. Since MATLAB Grader assessments were introduced for the first time in Semester 1 2022, a formal cohort-wide survey has not yet been conducted. Instead, user experience feedback was acquired from shorter informal surveys and non-structured discussions with students. A query into prior knowledge has unexpectedly revealed that despite being at the Masters level, some students have lower-than-expected MATLAB programming skills. As with BMEN20003, this could be explained by the presence of lateral entry students in MCEN90038 who may have come from other universities where MATLAB might not have been used as extensively.

From the end-user perspective, there was no mention of major technical problems accessing or using MATLAB Grader itself. However, there were some unexpected problems with submission timeouts and inconsistencies with the automated testing. In the first round of assessment, the tool worked as expected for about 95% of the cohort, but 5% encountered issues during submission. Even though students' solutions were correct, MATLAB Grader could not automatically assess their codes properly, either resulting in a timeout error, or erroneously marking solutions as incorrect. The exact problem has not yet been identified and further investigation is needed. Because of these inconsistencies, students who experienced such issues in their initial submissions were observed to develop a sense of insecurity when required to make submissions in subsequent assessments with limited attempts. Despite these setbacks however, most students appreciated the ease of use of the tool and considered it advantageous not having to focus on the lengthy calculations. As with ENGR10004 and BMEN20003, the feedback provided was considered insufficient and could be improved in future implementations.

Conclusion and Recommendations

MATLAB Grader is a versatile automated assessment platform that can be broadly applied across engineering courses apart from its primary purpose of reducing manual marking for large cohorts. From an educator's perspective, MATLAB Grader can be applied in different ways to offer flexibility in teaching and assessment across different engineering discipline-specific subjects and year levels, such as providing an avenue for programming practice to develop proficiency in programming skills, or providing automated assessment and feedback on their application in engineering problem-solving. The deployment and effectiveness of the platform are highly dependent on the specific learning objectives and focus of a particular subject.

From the end-user perspective, MATLAB Grader is a reliable and easy-to-use platform that integrates with the LMS for ease of access and the convenience of automated gradebook updates. While the implementation of MATLAB Grader in all three subjects generally resulted in positive student feedback, the quality of automated feedback was identified as a clear area for improvement. Future work in this area might involve the investigation of different feedback strategies. This will require the identification of common types of feedback and how they might be applied effectively or adapted across multiple applications/circumstances in the context of the MATLAB Grader platform.

Acknowledgements

The authors would like to acknowledge Mr Peter Koong and Mr Kaiwen Yang for assisting in the setup of MATLAB Grader in ENGR10004 and MCEN90038, respectively. They would also like thank Dr Shinsuke Matsubara (Senior Customer Success Engineer, MathWorks) for his guidance and assistance with troubleshooting around MATLAB Grader deployment and LMS integration.

References

Boada, Y., & Vignoni, A. (2021). Automated code evaluation of computer programming sessions with MATLAB Grader. *World Engineering Education Forum/Global Engineering Deans Council (WEEF/GEDC)*, (pp. 500-505).

Douce, C., Livingstone, D., & Orwell, J. (2005). Automatic test-based assessment of programming: A review. *Journal on Educational Resources in Computing*, 4-es.

Govender, I., & Govender, D. W. (2010). An Exploratory Study: The effectiveness of a Learning Management System (LMS) in the delivery of a face-to-face programming course. *International conference on education, training and informatics.*

Ivanović, M., Xinogalos, S., Pitner, T., & Savić, M. (2017). Technology enhanced learning in programming. *Education and Information Technologies*, (pp. 2981-3003).

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., . . . Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *Working group reports on Innovation and technology in computer science education (ITiCSE)*, (pp. 204-223).

Smith, J. S. (June 2019). Determining Optimal Deployment Strategies of MATLAB Autograder to Maximize Student Learning and Engagement. *American Society for Engineering Education (ASEE) Annual Conference & Exposition.* Tampa, Florida.

Smith, N. (June 2020). Integration of Instructional Technology Tools Including Matlab Grader to Enhance Learning in a Hybrid Vibrations Course. *American Society for Engineering Education (ASEE) Virtual Annual Conference.*

Copyright statement