Extended Abstract

Implementation of interactive technology tools to improve students' active learning and engagement in MATLAB programming

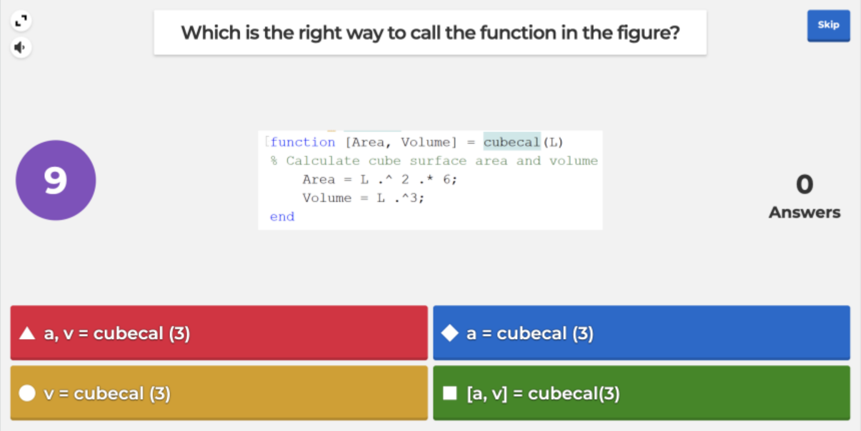
Xiyuan Liu
Department of Mechanical & Aerospace Engineering
Syracuse University

Many teaching technologies and digital platforms have been developed in the past decade to increase the student engagement and enhance the learning outcomes inside and outside the classroom. The trend in teaching a mechanical engineering course is transitioning from a traditional lecture-oriented passive classroom to a new and harmonious conjunction of interactive learning and practical hands-on experience. Recently, integration of flipped classrooms and gamification is designed and practiced in many different engineering courses, which has been demonstrated with a significant increase in learning interest, attitude, and student-instructor interaction[1]–[4]. The framework of such integration for person-centered education in engineering courses could potentially enable students to acquire self-reflective competencies and self-motivated problem solving. For example, Kahoot!, a flexible and simple tool that works on any devices, is being used widely at different levels of education system to create, play, and share games in the form of quizzes and homework[5]. Furthermore, new game features, like the Jumble game, have been introduced offering a different approach to the gaming experience[6]. Here, we demonstrated the use of eLearning techniques in a MATLAB programming course in the mechanical engineering program to provide immediate feedback and enhance student-instructor interactions.

ECS104 is a required course in the curriculum of Department of Mechanical and Aerospace Engineering. This course is designed to complement the analytical skills acquired in other courses with computer-based practice. One of the major student learning outcomes is to introduce MATLAB programming to the freshman engineering students, so they can practice programming skills for different analytical applications. The general class size for ECS 104 is 110 ~ 130 students each year. For the majority of the students, this usually is the first time to be exposed to a programming language, which is quite different from the course format that they are familiar with. Due to large class size, the students might not be able to receive enough attention or timely feedbacks on their work in or outside the classroom.

To better engage this large student cohort with the class content, I teach MATLAB programming in an interactive manner. In my lecture, new concepts are introduced followed with example problems that are specifically designed to match the lecture content. To solve this example problems in real-time, I post questions to students while programing in the MATLAB environment. Both the questions and real-time programming progress are projected onto the screens in front of the class, so students can participate in this interactive programming activities. Sometimes, students are provided with activity sheets and then asked to complete the code on them. Next, each student is paired with another classmate sitting closely, so they can correct each other's code. In this activity, students are encouraged to peer review their classmates' works by assessing the code performance, making comments, and suggesting improvements. This student peer review process could enhance student motivation and foster collaborative learning, and meanwhile help the instructor to manage large groups and promote the teaching effectiveness in a large-size classroom.

To further enhance the engagement, participation, and motivation of the students in this large class, I also adopt Kahoot! Quiz as interactive activities. Kahoot! is a game-based learning platform with instant student responses, thus it helps the instructor review student's knowledge, perform formative assessment, and give a break from traditional one-way classroom activities. It is designed to increase engagement, motivation, enjoyment, and concentration to improve classroom dynamics and student learning experience. In my class, 5-6 questions are displayed at the end of each lecture to not only evaluate the students' understanding on lecture content but also collect students' feedbacks on lecture/lab/homework learning experience. In an example of a Kahoot! question, students were asked to complete this question to reflect the concept of calling a user-defined function (Figure 1). One of the main learning outcomes for this specific lecture was to properly create and correctly use the user-defined functions in the MATLAB. This Kahoot! question was created to help students reflect and understand this most important concept covered during the lecture and provide me an opportunity to assess their daily learning outcomes. With these Kahoot! activities, students are exposed to a more playful, fun, and enjoyable learning experience on MATLAB programming principles and practice. Additionally, the students' opinion on course progression or my teaching styles also help me to better gauge the learning capacity of this cohort of students, so I can tailor my teaching pace and adjust my teaching styles according to the students' responses.



Which is the right way to call the function in the figure?

```
function [Area, Volume] = cubecal(L)
% Calculate cube surface area and volume
    Area = L .^ 2 .* 6;
    Volume = L .^3;
end
```

▲ a, v = cubecal (3)

◆ a = cubecal (3)

● v = cubecal (3)

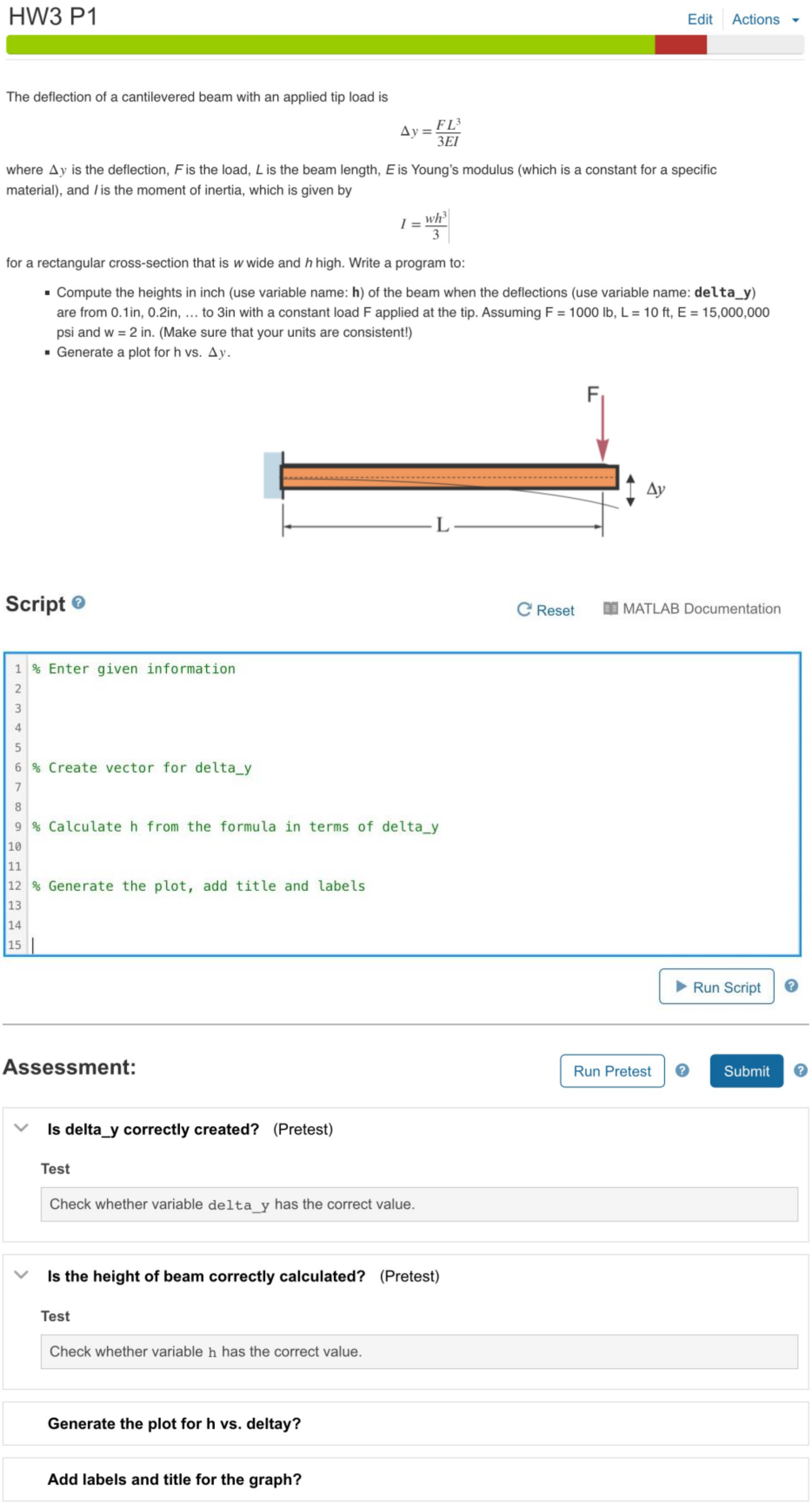
■ [a, v] = cubecal(3)

Figure 1. A kahoot! Quiz example on user-defined function

In addition to in-class eLearning activities with prompt student responses, I also use other digital platform that can offer students with detailed prompt feedback outside the classroom. More specifically, I adopted MATLAB Grader system and redesigned the homework questions to maximize student learning outcomes[7]. Traditionally, the homework in MATLAB programming was submitted in paper, graded by the Tas with human errors and returned to the students at least a week later. However, programming assignments are well suited to automated grading particularly in a test-driven modular-based system. MATLAB Grader provides a browser-based interface that can easily design, layout and dispatch homework assignments followed by automated grading and prompt feedback to students. Furthermore, this digital platform is web-based and did not require a local MATLAB installation to the personal computer.

For example, in homework 3, students were asked to complete a question related to beam deflection calculation which is a topic later introduced in the class of Statics (Figure 2). The students can write their codes in the "*Script*" and run the code for the answers. A template (showing in green comments) is provided by the instructor as hints to guide the students to solve the problem. The performance of the code will be assessed by pre-designed questions in the "*Assessment*". Under each assessment, I also provide some hints at the location where students could possibly make the mistakes. These hints can help them troubleshoot the codes for correct answers. With multiple attempts, students are motivated to practice more on each homework topic, learn troubleshooting techniques, and eventually be proficient at programming.

Overall, the MATLAB Grader platform allows me to scale the assessments for each problem, provide hints and guidelines to the students to complete a problem, and automatically grade the problem with timely responses. Based on the given responses, students have multiple opportunities to correct their codes accordingly. It has been demonstrated that with this platform, student learning outcomes and programming skills are significantly improved in a large class setting due to prompt student-instructor interactions. I also find that the "*Status Summary*" provides an overall view of the completion progression of the homework and delivered solutions in an ordered map.



The deflection of a cantilevered beam with an applied tip load is

$$\Delta y = \frac{FL^3}{3EI}$$

where Δy is the deflection, *F* is the load, *L* is the beam length, *E* is Young's modulus (which is a constant for a specific material), and *I* is the moment of inertia, which is given by

$$I = \frac{wh^3}{3}$$

for a rectangular cross-section that is *w* wide and *h* high. Write a program to:

- Compute the heights in inch (use variable name: **h**) of the beam when the deflections (use variable name: **delta_y**) are from 0.1in, 0.2in, … to 3in with a constant load F applied at the tip. Assuming F = 1000 lb, L = 10 ft, E = 15,000,000 psi and w = 2 in. (Make sure that your units are consistent!)
- Generate a plot for h vs. Δy.

Script

```
% Enter given information




% Create vector for delta_y


% Calculate h from the formula in terms of delta_y


% Generate the plot, add title and labels



```

Assessment:

Is delta_y correctly created? (Pretest)

Test

Check whether variable `delta_y` has the correct value.

Is the height of beam correctly calculated? (Pretest)

Test

Check whether variable `h` has the correct value.

Generate the plot for h vs. deltay?

Add labels and title for the graph?

Figure 2. A homework example in MATLAB grader with script guidelines and assessments.

To determine how these digital tools impact on student learning outcomes, I compared the students' performance in all three exams throughout the entire semester for the past three years (Figure 3(a)). I observed there was a significant increase of the average score in Exam 1 from 2018 to 2019, when I first introduced the digital tools of Kahoot! and MATLAB Grader. This data indicated that these digital tools had an immediate effect on the improvement of student understanding on the course materials and their performance in the exam. The impact became marginable for the following two exams, which might be due to the reason that students were getting used to the programming class setting. Introduction of these tools could engage the students on the hands-on programming at the beginning of the semester and prepare them for the significant challenges that were present in this class due to unfamiliar class setting and new demanding programming skills. Next, I compared the correction rate for one specific question that were asked in multiple different assessment settings: Kahoot! question on Monday when this question was asked for the first time right after lecture, Kahoot! question on Wednesday when this question was asked for the second time after the labs and lecture reiteration, homework provided after Wednesday lecture, and exam given a month later (Figure 3(b)). From the data, I observed that students' understanding on this question within the week was significantly improved from 71% correction rate on the Kahoot! Monday, to 87% on the Kahoot! Wednesday, and then 97% on the homework. The correction rate was dropped to 80% on the exam, considering for the factors of one-month time gap and stressful exam circumstance.

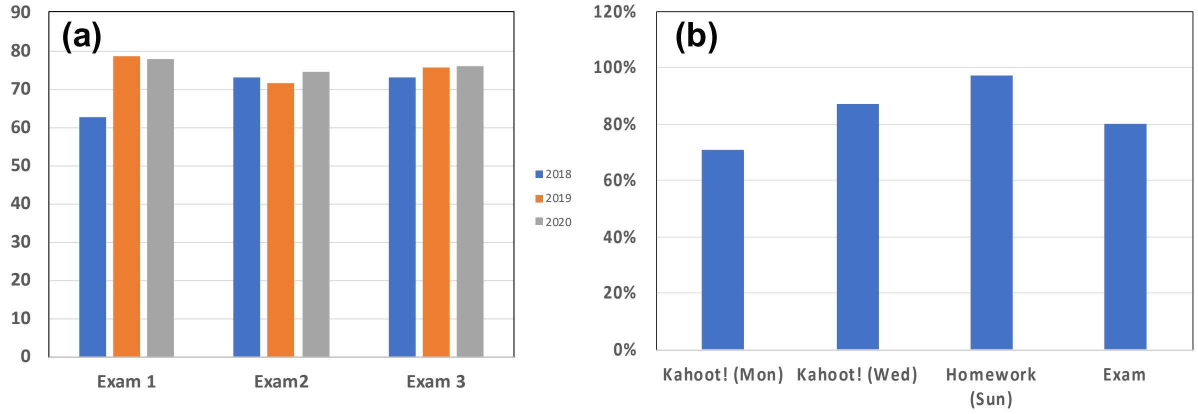


Figure 3. (a) Students' exam performance summary from year 2018 to 2020. (b) The correction rate on the same learning outcome using different assessments.

In summary, I believe that the gamification of in-class activities via Kahoot! and instructiveness of out-of-class activities via MATLAB Grader could significantly improve the pedagogical practices on active learning, enable an enjoyable learning environment, reinforce personalized learning experience, and promote close and prompt student-instructor interaction. These digital tools can positively affect classroom dynamics, improve student learning outcomes, and reduce student anxiety in a competitive but friendly atmosphere. Creating such interactive environments in a programming-centered engineering class can inspire students to actively participate in and outside the classes, allow students to enjoy their skill development and help students to improve their academic performance.

Reference

[1] B. Kerr, "The flipped classroom in engineering education: A survey of the research," in *2015 International Conference on Interactive Collaborative Learning (ICL)*, Sep. 2015, pp. 815–818. doi: 10.1109/ICL.2015.7318133.

[2] H. Baytiyeh and M. K. Naja, "Students' perceptions of the flipped classroom model in an engineering course: a case study," *European Journal of Engineering Education*, vol. 42, no. 6, pp. 1048–1061, Nov. 2017, doi: 10.1080/03043797.2016.1252905.

[3] G. Barata, S. Gama, J. Jorge, and D. Goncalves, "Engaging Engineering Students with Gamification," in *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, Sep. 2013, pp. 1–8. doi: 10.1109/VS-GAMES.2013.6624228.

[4] M. Milosz and E. Milosz, “Gamification in Engineering Education – a Preliminary Literature Review,” in *2020 IEEE Global Engineering Education Conference (EDUCON)*, Apr. 2020, pp. 1975–1979. doi: 10.1109/EDUCON45650.2020.9125108.

[5] “Welcome back to Kahoot! for schools,” *Kahoot!* https://kahoot.com/schools-u/ (accessed Mar. 29, 2022).

[6] “Kahoot! question types and game options to level up class engagement,” *Kahoot!*, Oct. 08, 2019. https://kahoot.com/blog/2019/10/08/new-question-types-and-game-options/ (accessed Mar. 29, 2022).

[7] “MATLAB Grader.” https://grader.mathworks.com/ (accessed Mar. 29, 2022).