A MATLAB Assignment Framework for Engineering Education that Automates Grading

Scott C. Rowe and Charles R. Nuttelman
Western Carolina University / University of Colorado Boulder

Abstract

Herein a MATLAB framework is detailed that uses programming to deliver engineering education. This approach departs from traditional engineering curricula that use programming as a supplement to content comprehension, versus the vehicle for content evaluation. Specifically, the new MATLAB framework consists of homework assignments that are entirely student coded, instantly validated, and ultimately computer graded. Notably, "fully programmatic" homework synthesizes both essential engineering concepts and marketable coding skills. The framework's flexibility has enabled full homework automation in diverse classes that include fluid dynamics, circuits, controls and separations. A quantitative survey of 83 former students compared the new approach to classical written homework. When compared to written homework 98% of alumni thought programmatic homework was more modern, 95% of alumni thought programmatic homework was similarly or more impactful and 93% of alumni viewed programmatic homework as similar or better workplace preparation.

Keywords

MATLAB, programming, engineering education

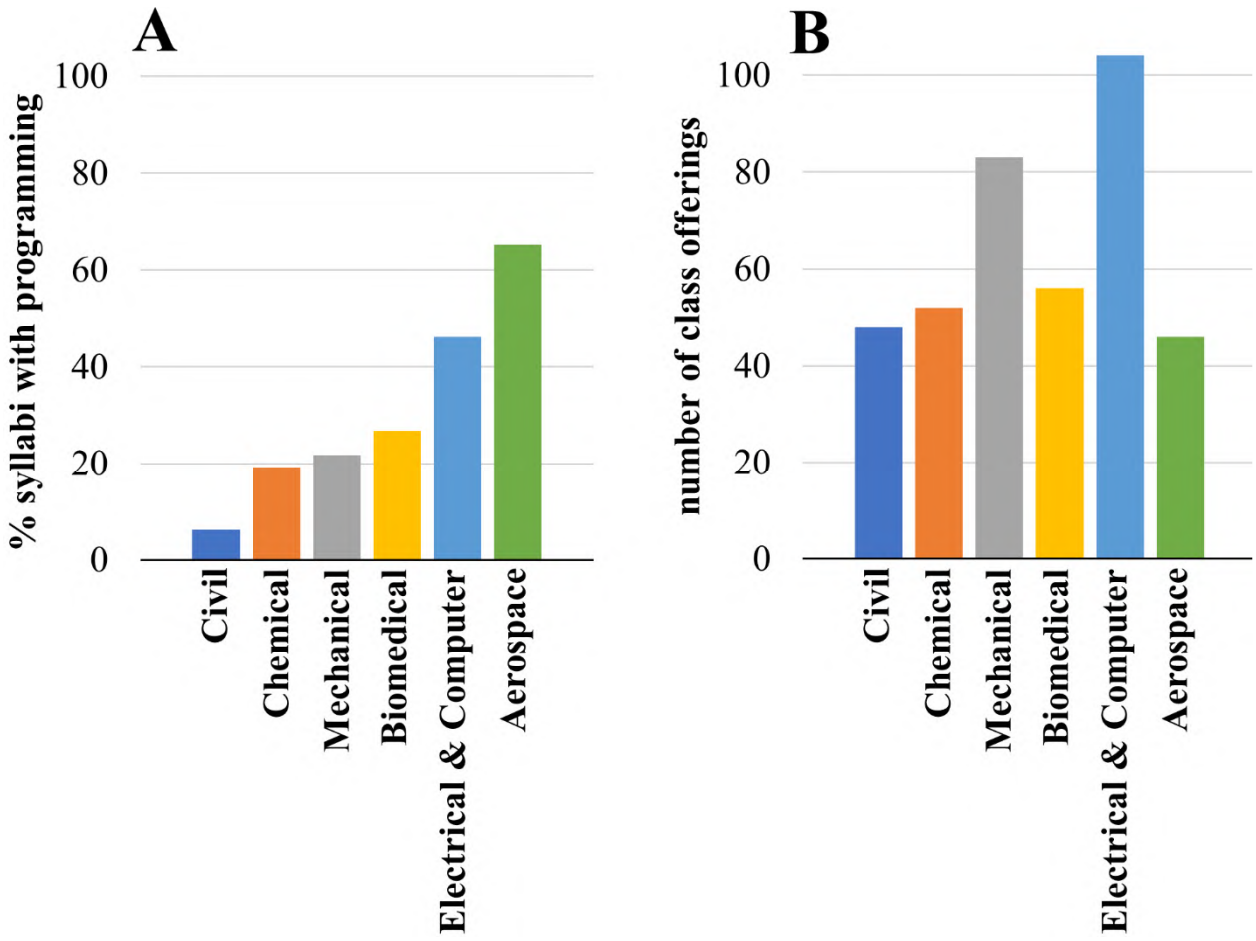
Introduction

We live in an era where online learning and content digitization have placed new pressures on universities.[1,2] To remain competitive in the information age college education likely must evolve. Given that engineering departments already offer dedicated programming classes, blending the wider engineering curriculum with computer science is a natural way to enhance the engineering degree.[3] In this context engineering is a medium students use to learn validated programming paradigms, including structured programming, event driven programming, and object oriented programming. These are key concepts, in high demand, that are often required for the simulation and understanding of modern engineering systems.[4–6] To advance the communication of programming concepts in routine engineering instruction we disclose a framework for fully "programmatic homework" in traditional engineering classes.

New engineering students are typically introduced to programming in a standalone class and approximately 75% of these courses teach the MATLAB programming environment.[3] However, outside a dedicated programming class the prevalence of coding across the wider engineering curriculum is unclear. Throughout the literature there are numerous examples of MATLAB activities that reinforce engineering content, but usually the code in these exercises is provided to students prepackaged with minimal or no added programming required.[5,7–16] Although these activities can embellish and buttress content, rarely do they integrate programming into class delivery.[17]

The integration of programming in classes is likely disclosed by course syllabi. To explore the prevalence of programming in engineering coursework syllabi from the University of Texas Austin were screened for "programming" and "MATLAB" keywords. The University of Texas Austin Cockrell College of Engineering seemed appropriate for examination given its large size, wide course offerings, top 20 ranking, and publicly databased syllabi.[18] As shown in Figure 1 younger engineering disciplines (Electrical, Aerospace and Biomedical) mention programming in syllabi more frequently than older engineering disciplines (Civil, Mechanical and Chemical), at least at the University of Texas Austin. A facile and flexible framework for programmatic homework in engineering courses, as described here, could increase the prevalence of coding integration across all disciplines.

Figure 1: A) The frequency 2018 syllabi that mentioned "MATLAB" and/or "programming" at the University of Texas Austin organized by engineering department. **B)** the number of unique 2018 classes offered at the University of Texas Austin in each engineering department.



Studies have shown that written homework suffers from the widespread availability of solutions manuals whose content students copy and submit.[19] Previously 33% of engineering students were found to submit copied solutions and 47% of students saw no ethical issue with these practices.[20,21] Learning gains may arise when a student copies solutions for submission, but plagerism likely interferes with full student engagement with and retainment of new knowledge.[22] The **Methods** section below reveals how the new fully programmatic homework framework combats this issue.

Whether the programmatic homework framework is more effective than traditional written homework is unknown. To explore this question students' perceived engagement with and learning gains from programmatic homework were queried among graduates exposed to this approach in 2020 during CHEN 4570 Process Control at the University of Colorado Boulder. The **Results** section quantitatively shows that alumni found MATLAB homework assignments more modern, more effective, and more career relevant than written homework. Followup studies that evaluate the effectiveness of programmatic homework via exam performance are underway.

Methods

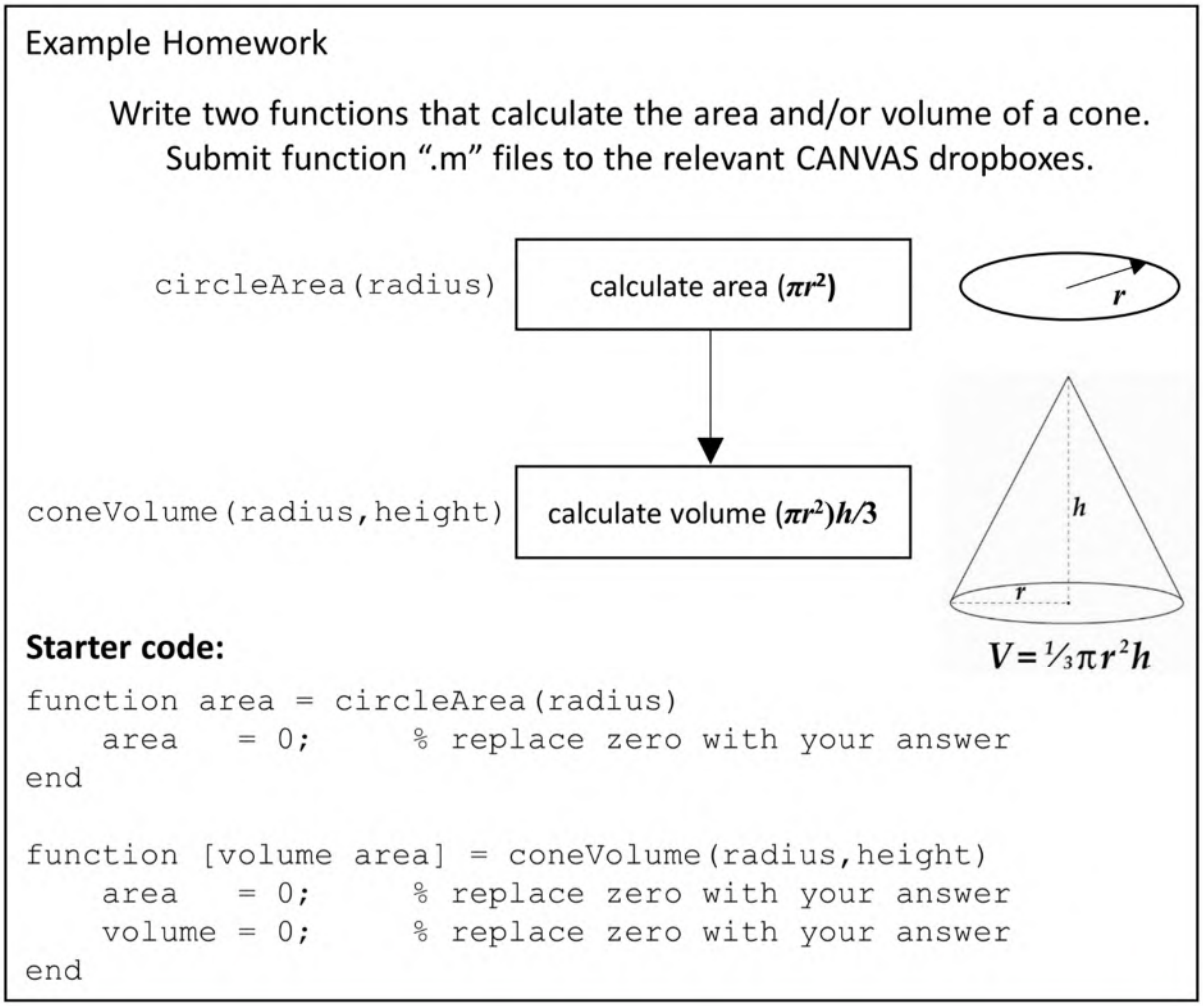
As with any assignment, an instructor must carefully craft programming problems for student implementation. Here student work conforms to the function archetype common to computer science, a structure that enforces consistent and gradable student code. Although this approach was the foundation for homeworks in multiple classes (fluid dynamics, circuits, controls and separations), a simple example is outlined here to illustrate the programmatic homework method. Specifically, Figure 2 shows the example assignment, which directs a student to implement basic geometric calculations in MATLAB code.

The example homework directive (Figure 2) provides instructions and function prototypes for student completion. Code success requires strict adherence to each provided function prototype and attention to labelled function inputs and outputs, atop correct calculations. Similarly, instructors should clearly comment code to assist student execution.

Figure 3 shows the overall homework workflow. Alongside the homework directive a `tester.p` file is always available for the verification of student work. The `tester.p` MATLAB "pcode" is an encrypted key to the assignment that can be downloaded and run on student computers to validate their work and assignment progress. Figure 4 shows how the `tester.p` is built from the sourcecode of an assignment key. Figure 5 shows how students run the `tester.p` locally to produce validation results.

Students upload completed work to CANVAS dropboxes setup for each function prototype in a given assignment. As shown in Figure 3, the instructor downloads submissions from each dropbox into a local folder whose name matches the assignment function prototype. A `process.m` function parses and organizes downloaded submissions into a folder on the instructor's computer for each student. This process is shown in Figure 6. A subsequent call to the `HWgrader.m` visits each student folder to grade student work, which is illustrated in Figure 7. Although the CANVAS learning platform was shown here, presumably the framework can be adapted to other learning environments.

Figure 2: An example homework to illustrate the MATLAB programmatic homework framework.



Results

To assess the impact of fully programmatic homework on students alumni were surveyed for their opinion of coding exercises in engineering education. Specifically, all 149 students from the 2020 University of Colorado CHEN 4570 Instrumentation and Process Controls class were targeted on LinkedIn.com with a message that requested curricular feedback. Seventeen of these alumni were unfindable on Linkedin.com. The remaining 132 solicited alumni were classified according to their self-reported LinkedIn.com status as follows:[23]

1) technical engineer (96 alumni); alumnus with a job in either process engineering, bioprocess engineering, engineering consulting, or technical sales.
2) graduate studies (20 alumni); alumnus in doctoral, masters, business, law, or medical studies.
3) government role (4 alumni); alumnus with a job in the United States government.
4) nontechnical or unknown status (12 alumni): alumnus in a nontechnical or undisclosed role.

Figure 3: The MATLAB programmatic homework workflow.

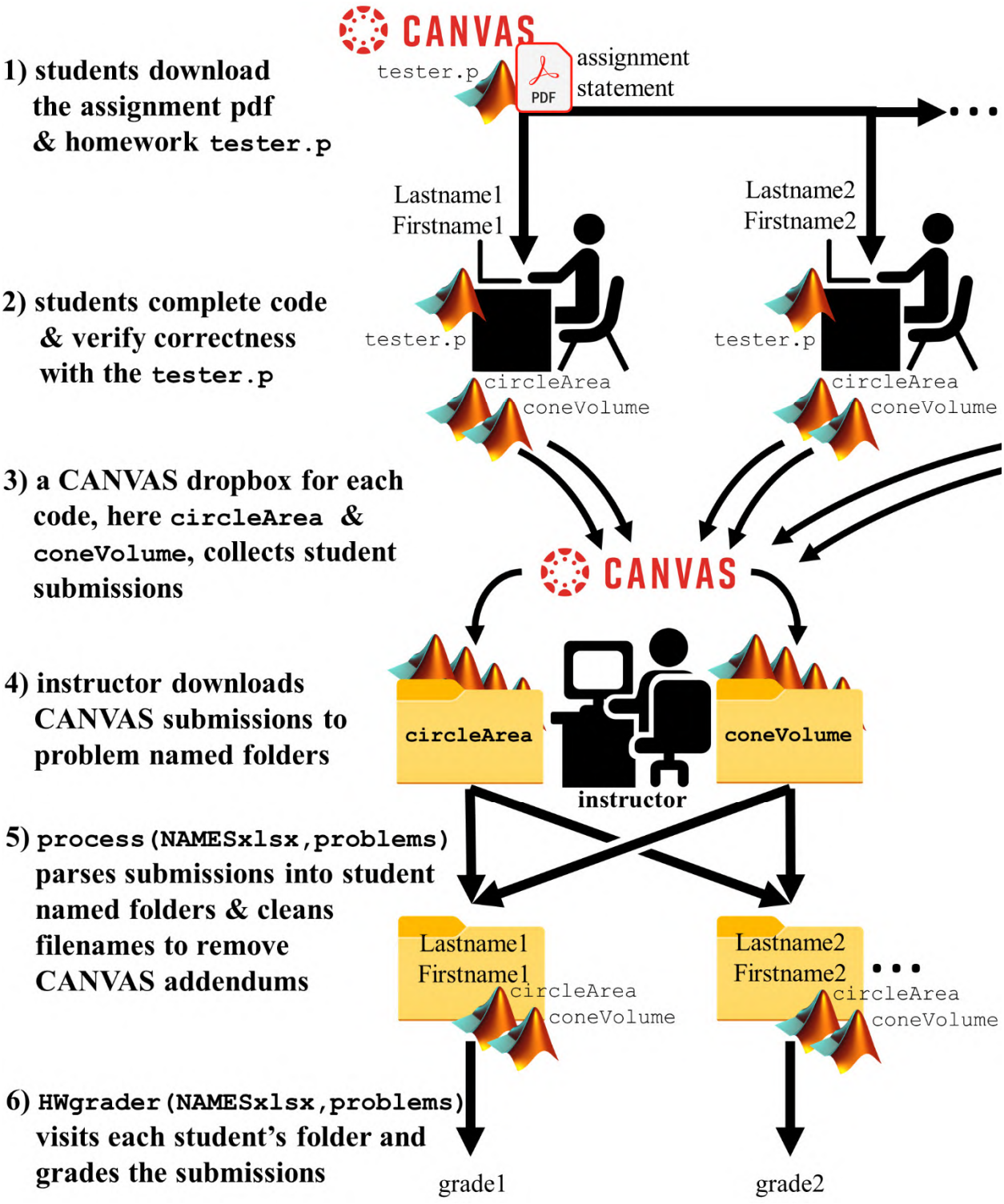


Figure 4) The `tester` sourcecode that students can run to validate their work is compiled into a "pcode" encrypted executable for secure distribution to a class.

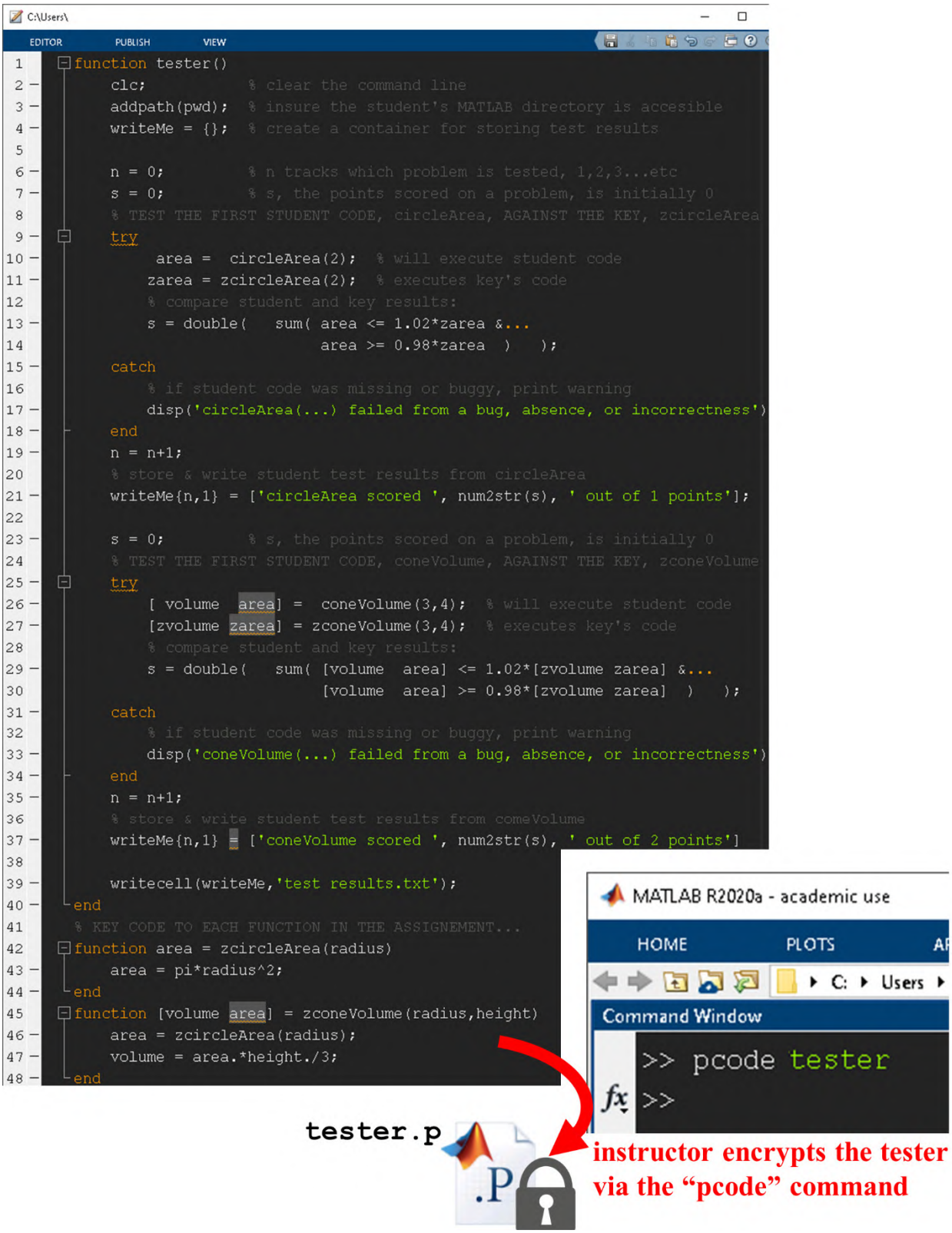


Figure 5) The student's approach to assignment assessment with the encrypted `tester.p` homework validation tool.

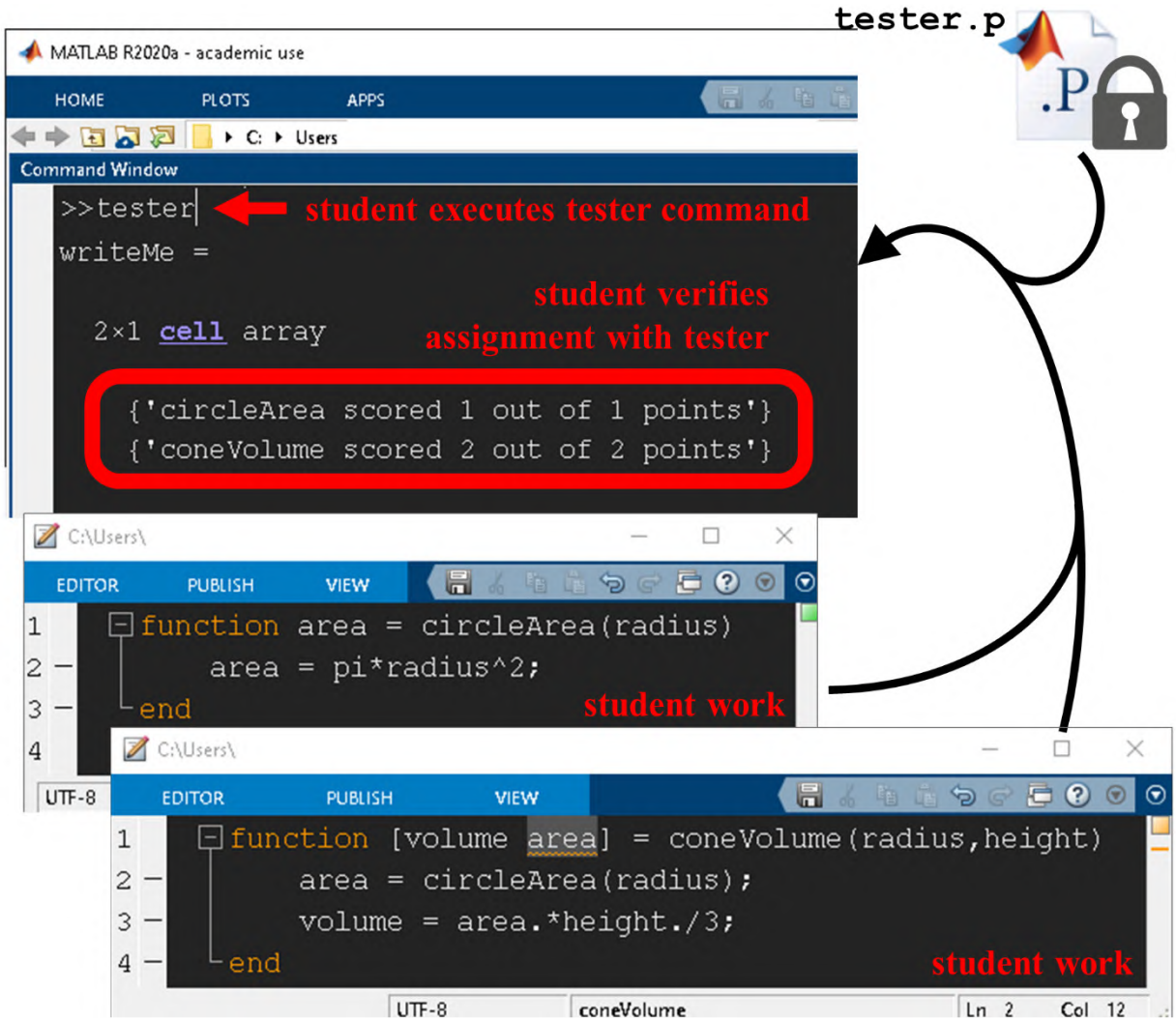


Figure 8 summarizes the professional breakdown of class alumni. Of the 132 alumni messaged, 83 voluntarily completed an anonymous "Google Form" that queried their opinions of fully programmatic homework. Thus, minimally 42 industry representatives completed the survey, although contributions from industry could be higher. If sample population demographics reflect observed occupational roles 60 industry representatives likely submitted surveys.

The 2020 alumni survey included three quantitative and one qualitative question:

1) Likert Scale: "Please compare the modernity of MATLAB coding homeworks against written homeworks."
2) Likert Scale: "Please compare the educational impact of MATLAB coding homeworks against written homeworks."
3) Likert Scale: "Please compare the career preparation provided by MATLAB homeworks against written homeworks."
4) Free Response: "Please offer any positive and/or negative opinions regarding the MATLAB coding homeworks."

Figure 6: The `process.m` function organizes student CANVAS submission for grading.

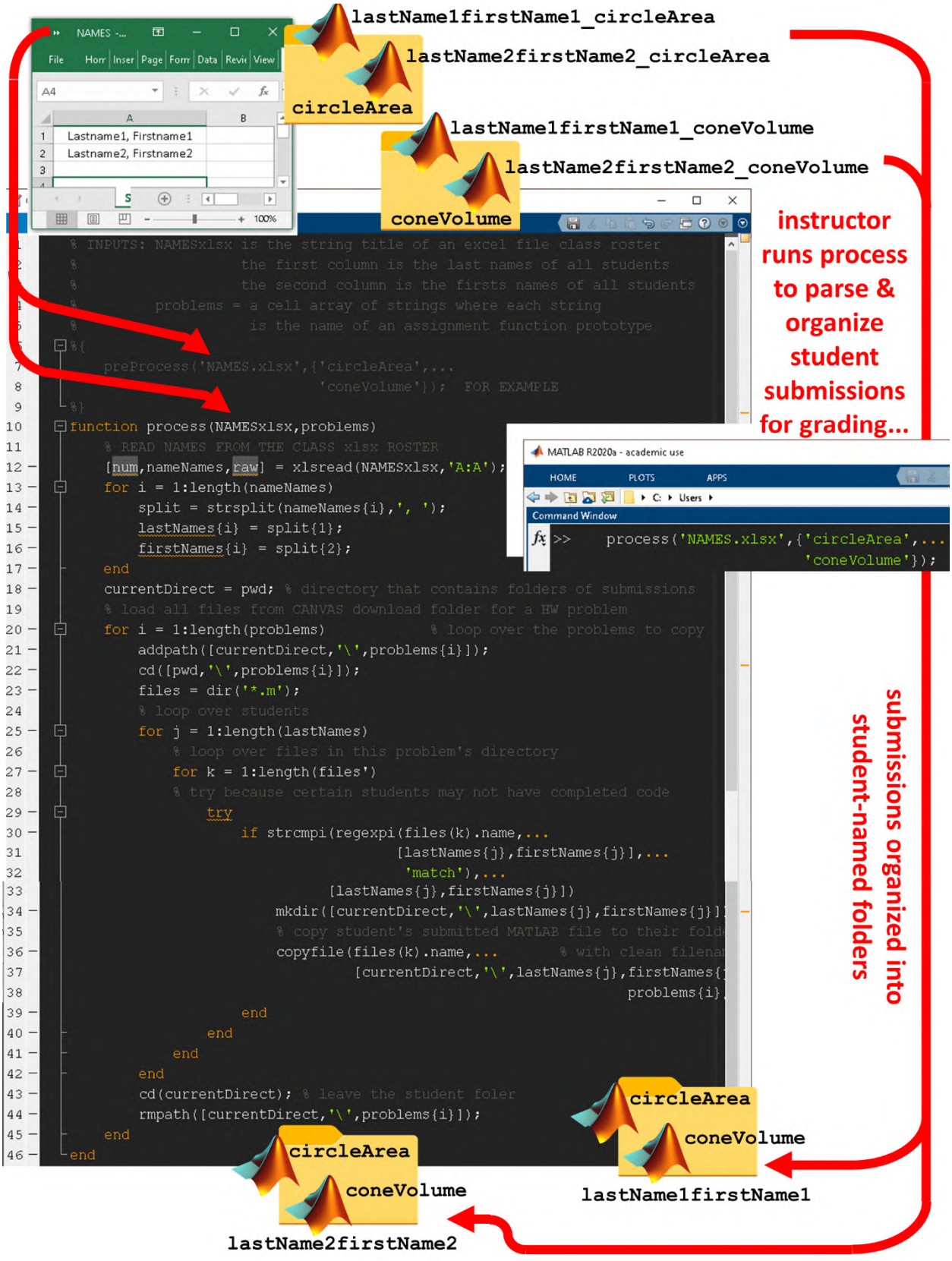
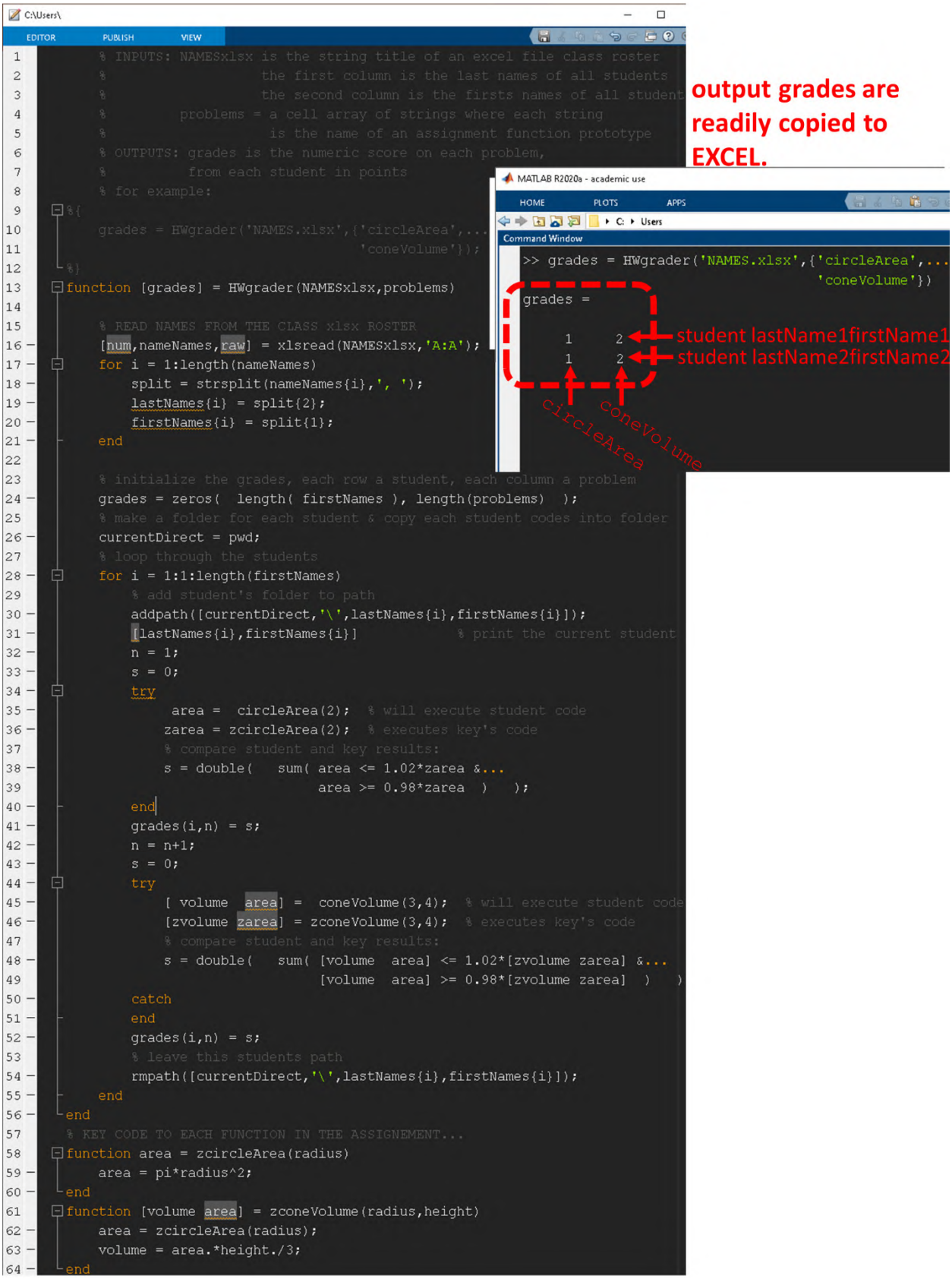


Figure 7: The `HWgrader.m` function visits each student folder's submissions & provides results.




```
% INPUTS: NAMESxlsx is the string title of an excel file class roster
%                   the first column is the last names of all students
%                   the second column is the firsts names of all student
%         problems = a cell array of strings where each string
%                    is the name of an assignment function prototype
% OUTPUTS: grades is the numeric score on each problem,
%          from each student in points
% for example:
%{
grades = HWgrader('NAMES.xlsx',{'circleArea',...
                                'coneVolume'});
%}
function [grades] = HWgrader(NAMESxlsx,problems)

% READ NAMES FROM THE CLASS xlsx ROSTER
[num,nameNames,raw] = xlsread(NAMESxlsx,'A:A');
for i = 1:length(nameNames)
    split = strsplit(nameNames{i},', ');
    lastNames{i} = split{2};
    firstNames{i} = split{1};
end

% initialize the grades, each row a student, each column a problem
grades = zeros(  length( firstNames ), length(problems)  );
% make a folder for each student & copy each student codes into folder
currentDirect = pwd;
% loop through the students
for i = 1:1:length(firstNames)
    % add student's folder to path
    addpath([currentDirect,'\',lastNames{i},firstNames{i}]);
    [lastNames{i},firstNames{i}]            % print the current student
    n = 1;
    s = 0;
    try
         area =  circleArea(2);  % will execute student code
        zarea = zcircleArea(2);  % executes key's code
        % compare student and key results:
        s = double(   sum( area <= 1.02*zarea &...
                           area >= 0.98*zarea  )   );
    end
    grades(i,n) = s;
    n = n+1;
    s = 0;
    try
        [ volume  area] =  coneVolume(3,4);  % will execute student code
        [zvolume zarea] = zconeVolume(3,4);  % executes key's code
        % compare student and key results:
        s = double(   sum( [volume  area] <= 1.02*[zvolume zarea] &...
                           [volume  area] >= 0.98*[zvolume zarea]  )   )
    catch
    end
    grades(i,n) = s;
    % leave this students path
    rmpath([currentDirect,'\',lastNames{i},firstNames{i}]);
end
end
% KEY CODE TO EACH FUNCTION IN THE ASSIGNEMENT...
function area = zcircleArea(radius)
    area = pi*radius^2;
end
function [volume area] = zconeVolume(radius,height)
    area = zcircleArea(radius);
    volume = area.*height./3;
end
```

Figure 9 shows survey results from the 83 participants. Relative to written homework alumni viewed programmatic homework as more modern, more impactful and better career preparation. Only one alumnus thought written homeowork was more modern (Figure 9). Similarly, only four alumni believe written homework harbors more educational impact (Figure 9). Finally, relative to programming homework, only 6 alumni thought written homework offered better career preparation (Figure 9). Thus, survey respondents showed a strong bias towards programmatic homework.

Questionnaire qualitative responses showed similar bias and favoritism towards programmatic, versus written, homework. Of the 83 survey participants, 67 provided comments and opinions on MATLAB coding homeworks. For brevity exemplative feedback is quoted here. Several students noted the workplace relevance of programmatic homeworks, for example:

"I didn't realize how beneficial it was to be comfortable with using MATLAB until after I got employed."

"[Programmatic homeworks] were much more difficult than the written assignments, but prepared me for an industry role much better."

Other alumni noted the need for programming in core curricula, for example:

"Coding is an important subject that is needed to be taught or at least introduced to engineering students through their education."

"Coding is by far the most important skill I learned during my entire undergraduate education…I think it is invaluable for students to learn how to translate theory into practical coding applications, because that is how problems are solved in the world. I heard somewhere that the true test for knowing anything is being able to express that thing in code."

A notable trend was evident among the 67 comments. Specifically, 12% of respondents expressed an interest in the conversion of MATLAB assignments into Python assignments. Relative to MATLAB these alumni cited Python as freeware with high industry prevalence.

Figure 8: Self-reported LinkedIn.com status of the 132 alumni solicited for survey feedback.

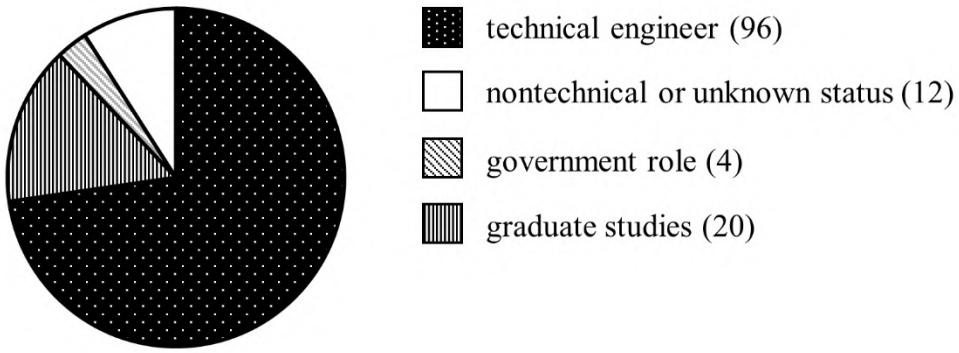
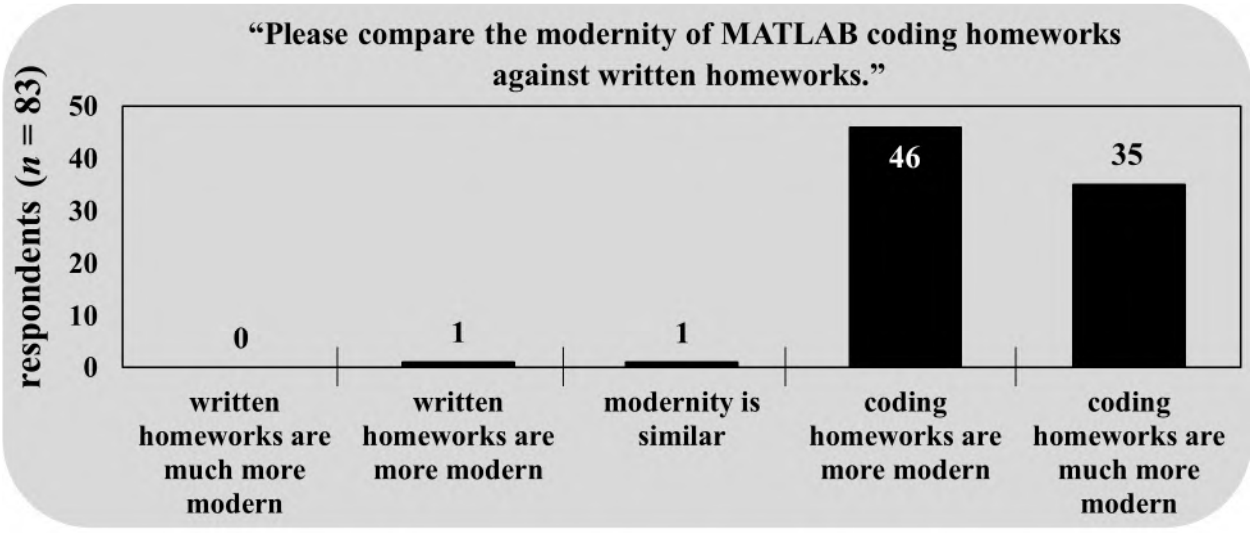
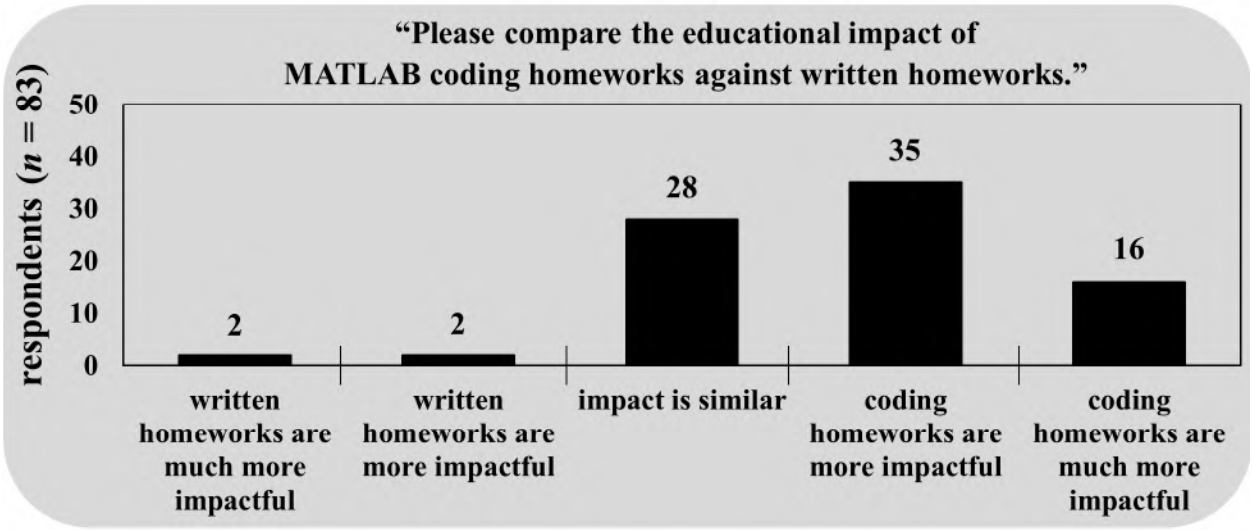
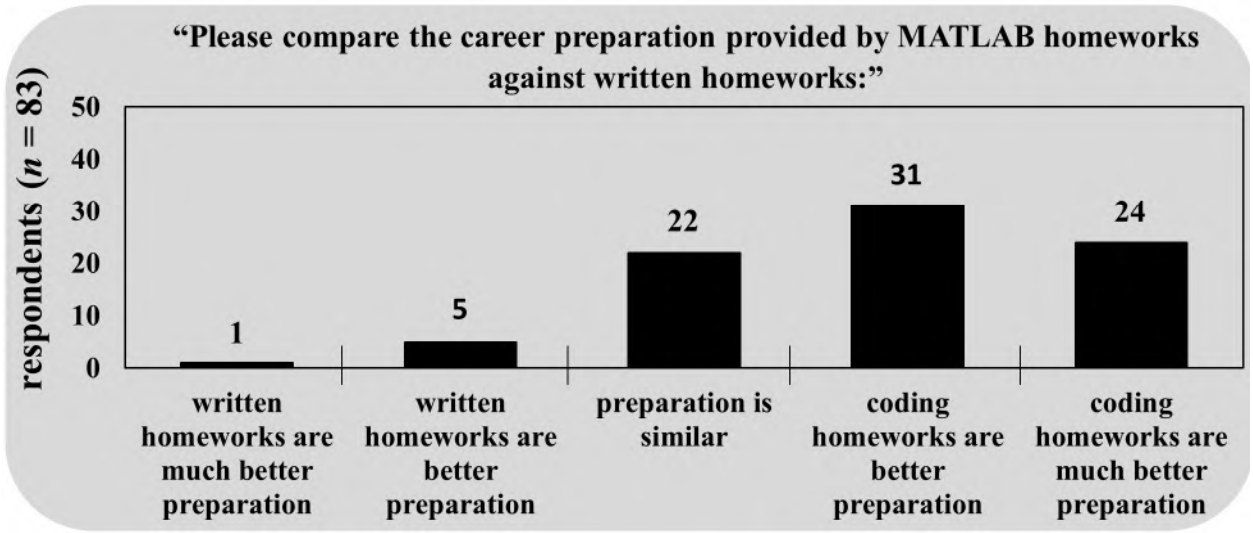


Figure 9: results of the alumni survey submitted by 83 voluntary participants from 2020:







Discussion

The fully programmatic approach to homework was used as the exclusive framework for homework execution in fluid dynamics, circuits, controls and separations classes. The method is described generically in Figures 2 through Figure 7, which outline a framework for content delivery via MATLAB. A retrospective questionnaire showed that alumni found the MATLAB programmatic homework method more modern, more impactful and more workplace relevant than written homework. Of the 249 questions levied only 11 responses showed a preference for written homeworks over programmatic homework.

To inhibit academic dishonesty students are never provided an explicit answer key to programmatic homeworks. Instead an encrypted program is distributed that students download to validate the correctness of their code and homework submissions. Thus, answers are always available, but never actually disclosed. However, unscrupulous students can still cheat once a given problem is solved by peer(s). Indeed, one student noted the ease of digital plagiarism:

"I think it's harder for people to cheat on written homework because it's not possible to copy and paste. Either way that's an error with the human, not an issue with the value of coding homework."

A databank of problems is often used in these instances, which is sampled randomly to produce custom homework sets for individual students.[26] Thereafter the variety of problems distributed to students can wholly impede plagiarism. Such an approach could be applied to programmatic homeworks to combat cheating. Alternatively, text-based plagiarism detection is applicable to programmatic homework.[27] Unlike written homework where isolated numeric answers are usually graded, code itself is searchable for copied patterns and verbatim cheating.

Necessity of the `tester.p` program (Figure 4 and Figure 5) is debatable. Instructors could simply provide sample program inputs and valid outputs within assignment directives to reveal the correctness of student code. However, within the programmatic homework workflow `tester.p` development naturally arises from `HWgrader.m` construction. Use of the `tester.p` program then allows students to challenge their work within the context of eventual grading.

Some students noted the exacting nature of programming as an evaluation medium:

"Coding problems were great because it forced you to think outside the box and there were often multiple ways to get to the end result which I think nurtured creativity. ***However, I will say that coding problems can be very very frustrating because sometimes it's not working because of a small error and it can be disheartening."***

Students vocalized similar complaints throughout class. Instructors responded with the same rationale: engineering requires exquisite awareness of particulars. Like engineering debugging demands attention to detail and complexity management.

Although alumni feedback widely lauded the programmatic approach to class assignments, many students sought Python homeworks instead of MATLAB tasks. Of the 67 comments, 12% of respondents would have preferred to program in Python. Currently Python is the most queried and

popular computer language worldwide [24,25]. Notably, the Tiobe Index lists Python as 1,400% more popular than MATLAB (https://www.tiobe.com/tiobe-index/). The programmatic homework framework could be adapted to Python, which is likely worthwhile given Python's prevalence.

Survey results (Figure 9) showed that alumni found programmatic homework superior to written homework. However, whether these self-reported perceptions translate into higher grades and enhanced content comprehension is unclear. Followup studies that compare the effect of written homework to programmatic homework on exam scores are planned.

References

1. Waldrop, M. M. Online learning: campus 2.0. *Nature News* **495**, 160 (2013).
2. Selingo, J. Colleges can still save themselves. Here's how. *The Chronicle of Higher Education* **60**, 14 (2013).
3. Hodge, B. K. & Steele, W. G. A survey of computational paradigms in undergraduate mechanical engineering education. *Journal of Engineering Education* **91**, 415–417 (2002).
4. Dishman, L. Why coding is still the most important job skill of the future. *Fast Company* **14**, (2016).
5. Assi, A. H. *Engineering education and research using MATLAB*. (2011).
6. Castillo, E., Conejo, A. J., Pedregal, P., Garcia, R. & Alguacil, N. *Building and solving mathematical programming models in engineering and science*. vol. 62 (John Wiley & Sons, 2011).
7. Al-Masoud, N. Integrating MATLAB Graphical User Interface in Statics Course. in *2006 Annual Conference & Exposition* 11.788. 1-11.788. 12 (2006).
8. Brownjohn, J. M. W. & Pavic, A. NDOF;: a MATLAB GUI for teaching and simulating structural dynamics. *IMACXXVI, Orlando, Florida, USA* 1–8 (2008).
9. ÇETİN, B., ÖZTÜRK, Y. & TAZE, S. VISCOUS FLOW CALCULATIONS FOR UNDERGRADUATE FLUID MECHANICS EDUCATION USING MATLAB.

10. Espinosa, H. G. & Thiel, D. V. MATLAB-Based interactive tool for teaching electromagnetics [Education corner]. *IEEE Antennas and Propagation Magazine* **59**, 140–146 (2017).
11. Ghosh, A. & Pantaleón, C. Teaching Computational Fluid Dynamics Using MATLAB. in *ASME International Mechanical Engineering Congress and Exposition* vol. 56277 V005T05A024 (American Society of Mechanical Engineers, 2013).
12. Khader, S., Hadad, A. & Abu-Aisheh, A. A. The application of psim & matlab/simulink in power electronics courses. in *2011 IEEE Global Engineering Education Conference (EDUCON)* 118–121 (IEEE, 2011).
13. Ming, Y. Application of Matlab in Teaching of Statics in Theoretical Mechanics Course. (2015).
14. Radin, V. P., Poznyak, E. V., Novikova, O. V. & Khromatov, V. E. The Use of MATLAB for Academic Course Mechanics of Materials and Structures. in *2018 IV International Conference on Information Technologies in Engineering Education (Inforino)* 1–4 (IEEE, 2018).
15. Shine, S. J. & Sebusang, S. E. M. Using MATLAB as a teaching and learning tool for beam bending problems in mechanics. (2004).
16. Williams, J. *et al.* RPI-MATLAB-Simulator: A Tool for Efficient Research and Practical Teaching in Multibody Dynamics. in *VRIPHYS* 71–80 (2013).
17. Rhudy, M. & Nathan, R. Integrated development of programming skills using MATLAB within an undergraduate dynamics course. in *2016 ASEE Annual Conference & Exposition* (2016).

18. Monks, J. & Ehrenberg, R. G. US News & World Report's college rankings: Why they do matter. *Change: The Magazine of Higher Learning* **31**, 42–51 (1999).
19. Kelley, R. & Dooley, B. The technology of cheating. in *2014 IEEE International Symposium on Ethics in Science, Technology and Engineering* 1–4 (IEEE, 2014).
20. Carpenter, D. D., Harding, T. S., Finelli, C. J., Montgomery, S. M. & Passow, H. J. Engineering students' perceptions of and attitudes towards cheating. *Journal of Engineering Education* **95**, 181–194 (2006).
21. Gallant, T. B., Van Den Einde, L., Ouellette, S. & Lee, S. A systemic analysis of cheating in an undergraduate engineering mechanics course. *Science and engineering ethics* **20**, 277–298 (2014).
22. Srikanth, M. & Asmatulu, R. Modern cheating techniques, their adverse effects on engineering education and preventions. *International Journal of Mechanical Engineering Education* **42**, 129–140 (2014).
23. Rowe, S. C., Samson, C. I. & Clough, D. E. A Framework to Guide the Instruction of Industrial Programmable Logic Controllers in Undergraduate Engineering Education. *Education for Chemical Engineers* **31**, 76–84 (2020).
24. Kumar, K. & Dahiya, S. Programming languages: A survey. *International Journal on Recent and Innovation Trends in Computing and Communication* **5**, 307–313 (2017).
25. Robinson, D. *The Incredible Growth of Python*. https://stackoverflow.blog/2017/09/06/incredible-growth-python/ (2017).
26. Evrard, A. *et al.* Problem roulette: Studying introductory physics in the cloud. *American Jounal of Physics* **83**, 76 (2013).
27. Cortinhas, C. *Detection and Prevention of Plagiarism in Higher Education*. (2017).