Motivation on Programming Assignments in Materials Science and Engineering

Dr. Susan P. Gentry, University of California, Davis

Dr. Susan P. Gentry is an Assistant Professor of Teaching Materials Science and Engineering at the University of California, Davis. In her current position at UC Davis, she is integrating computational modules into the undergraduate and graduate materials curriculum. She is specifically interested in students' computational literacy and life-long learning of computational materials science tools.

Motivation on Programming Assignments in Materials Science and Engineering

Abstract

The curriculum for many materials science and engineering (MSE) programs includes computational topics such as writing programming code or using software packages to simulate materials phenomena. Faculty agree that computational materials science and engineering (CMSE) is an essential topic for students to learn, but students' attitudes are more varied. This paper investigates students' CMSE motivation using the framework of expectancy-value theory. This model suggests that a persons' motivations are influenced by both their expectancy beliefs ("Can I succeed?") and value beliefs ("Is it worth it?"). Students' value-based motivations were measured for programming and other course assignments such as homework. The survey was distributed in the same junior-year materials kinetics class in subsequent years, with differing results. In study 1, students were significantly less motivated to complete programming assignments than other course assignments. However, the following year, Study 2 found that there was no significant difference; students were equally motivated to complete programming and other course assignments. Furthermore, students articulated that programming skills are essential for engineers to be efficient at their jobs. The contrasting results are discussed, presenting several hypotheses for the dissimilar attitudes.

Introduction

Computational materials science and engineering (CMSE) is vital to the field of materials science and engineering (MSE) as simulations can aid the development of advanced materials and processing routes. CMSE includes the use of commercial software packages as well as developing computer code that simulates materials phenomena. Specifically, simulations can be used to predict phase stability, microstructural evolution, or materials properties. Furthermore, multiple simulations can be linked through multiscale materials modeling so that physical properties can be related to microstructural and atomistic simulations [1]. For instance, Pratt and Whitney, a manufacturer of jet engine turbine rotors, combined solid mechanics codes with material property models to improve the materials and design of their rotors [2]. Advances in CMSE are allowing materials researchers to design innovative materials using simulations and computer tools.

Within the materials education community, there is currently a thrust to integrate CMSE into the undergraduate curriculum. A 2018 study by Enrique, Asta, and Thornton investigated the use of CMSE in the undergraduate MSE curriculum [3]. *All surveyed faculty and department chairs* agreed that CMSE should be integrated into the core curriculum, with 90% of chairs and 72% of CMSE faculty noting that this was a "very important" topic. Furthermore, 84% of MSE departments offer at least one CMSE course for undergraduate students. The authors compare the 2018 results to those from a 2009 study [4], concluding that "CMSE is no longer viewed as a special interest topic, but rather a core part of the curriculum" [3]. There are different methods of teaching CMSE at colleges and universities. For instance, CMSE topics are distributed amongst the disciplinary classes at Boise State University [5] and the University of Illinois at Urbana-Champaign [6]. In contrast, Ohio State University teaches a three-semester CMSE lab course within the required curriculum [7]. Within MSE, faculty value CMSE and are increasingly incorporating these topics into the undergraduate curriculum.

In contrast, students have mixed perceptions of CMSE. For instance, one institution with a stand-alone CMSE course sequence found that students responded positively to survey questions regarding the importance of computational tools [7]. Nonetheless, 24% of those students ranked the CMSE course as the *least* valuable course in the curriculum! Studies at other institutions [6, 8] have similarly found students respond positively towards computational skills when asked questions such as "I feel computation (data visualization, modeling, and simulation algorithm design) will be useful in my career" [9]. Both studies found no significant change in these attitudes before and after completing CMSE modules. However, these studies do not provide a basis for comparison to other topics; as Ref. [6] notes, "the students may have answered with affirming responses simply to make the survey writer happy." These three studies indicate that further research is needed to understand students' perceptions and motivations as they relate to CMSE.

Study Overview

This work further investigates students' motivation on programming assignments within MSE. Prior to the study, three research questions (RQs) were defined:

> RQ1: Do students perceive programming assignments as less valuable than other types of assignments in their courses (such as homework)?
> RQ2: What are students' motivations on programming assignments in technical engineering courses, as they relate to task value, intrinsic goal orientation, and extrinsic goal orientation?
> RQ3: Do students see the value of programming skills for both computer-based simulations and data analysis?

This paper further investigates students' motivations towards CMSE using expectancy-value theory as the motivational framework. For educational activities, Wigfield and Eccles have posited that "individuals' choice, persistence, and performance can be explained by their beliefs about how well they will do on the activity and the extent to which they value the activity" [10]. This is the foundation of their expectancy-value theory of motivated learning [10, 11], in which students' expectancies (believing that they can succeed) are comprised of efficacy and ability beliefs and their values are comprised of intrinsic motivation, extrinsic motivation, task value, and cost. When students do not see the value of an educational activity and/or their self-efficacy is low, they may have "evading" or "rejecting" behavior, even in a supportive learning environment [12]. Thus, it is vital that instructors understand students' motivations for learning CMSE topics.

Methods

Two studies were conducted to assess students' motivation on programming assignments. Both studies quantified students' motivation using the Motivated Strategies for Learning Questionnaire (MSLQ) [13]. The MSLQ is a validated instrument [14] which uses statements with a Likert scale to investigate students' expectancy-, value-, and attitude-based motivations at the course level. Three sets of questions related to value were utilized: intrinsic goal orientation, extrinsic goal orientation, and task value. The MSLQ was modified to specifically query students' value beliefs regarding programming assignments. Sample statements for each of these value beliefs are as follows:

- *Intrinsic Goal Orientation:* "In a class like [the target course], I prefer programming assignments that really challenge me so I can learn new things."
- *Extrinsic Goal Orientation:* "Getting a good grade on programming assignments is the most satisfying thing for me right now."
- *Task Value:* "It is important for me to learn the material for the [target course] programming assignments."

As a comparison dataset, the statements were repeated replacing the language of "programming assignments" with "course assignments (such as homework)." The full surveys are included in Appendices A and B.

The subsections below describe two studies that were conducted in subsequent years, after review by the Institutional Research Board at the institution. Students' responses were collected anonymously, providing them with a $5 Starbucks gift card as an incentive for completion.

Study 1

This survey was distributed to students at the conclusion of the term of the Winter 2018 MSE kinetics course. The survey had three sections:

1) MSLQ-based survey of programming assignments
2) MSLQ-based survey of other course assignments (such as homework)
3) Questions about prior knowledge of programming

Appendix A provides a summary of the survey items.

Study 2

Study 2 was designed to expand upon the information gathered in Study 1. Study 2 was distributed to students at the beginning of the Winter 2019 MSE kinetics course. This study focuses on students' task value beliefs related to programming, omitting the Study 1 survey items related to intrinsic and extrinsic goal orientation. The study added questions about perceived future use and the benefit of programming skills. Writing skills were used as a comparison set since Pulford previously investigated engineering students' perceptions of writing skills [15]. The survey had four sections:

1) MSLQ-based survey of task value on programming assignments
2) MSLQ-based survey of task value on other course assignments (such as homework)
3) Predicted future use of writing and programming skills
4) Short answer questions about how it benefits an engineer to program or write well. These questions are adapted from Ref. [15].

Appendix B summarizes the survey for Study 2.

Overview of Studies 1 and 2

Study participants were recruited from students enrolled in a junior-year materials kinetics course at a large research institution. The required MSE course integrates CMSE topics using a series of four MATLAB programming assignments throughout the term [16]. Students first complete two introductory modules that prepare them to complete two comprehensive modules related to course content. A comparison of the instructor and course enrollments are given in Table 1. The two terms represent the third and fourth times that the instructor had taught the class, each time incorporating MATLAB programming assignments. The enrollment by major and self-reported gender identity are also included. Grade information is unavailable at this time.

A summary of the surveys is also provided in Table 1. The survey was completed anonymously by students in the course, so specific demographics of survey participants are unknown.

Table 1. Comparison of the courses and surveys for studies 1 and 2.

	Study 1 (Winter 2018)	**Study 2 (Winter 2019)**
Course information		
Course topic	Materials kinetics (junior-year undergraduate)	
Instructor	Instructor X	Instructor X
Course Enrollment	45 students	38 students
Enrollment by Major	MSE majors: 40 (89%) Non-majors: 5 (11%)	MSE majors: 33 (87%) Non-majors: 5 (13%)
Gender Identity	33% female 67% male/other	26% female 74% male/other
Number of Required Programming Assignments	4 total (2 preparatory, 2 comprehensive)	4 total (2 preparatory, 2 comprehensive)
Survey Information		
Number of Participants (Percentage of Class)	21 (47%)	16 (42%)
Survey Timing	After completion of term	Start of term
Description of Survey Items	- Comparison of intrinsic motivation, extrinsic motivation, and task value of programming and other course assignments - Prior knowledge of programming	- Comparison of task value of programming and other course assignments - Comparison of programming and writing skills - Prior knowledge of programming

Study 1 Results

The survey was completed by 21 students, representing 47% of the class enrollment. The survey indicated that prior to the course, students should have had sufficient programming knowledge to complete the MATLAB assignments. All respondents had taken an introductory programming course and had taken a course requiring the use of MATLAB as a programing language. (Not all introductory courses teach MATLAB as a programming language). 62% of respondents had taken one prior course, while the remaining 40% had taken two or more courses.

The participants completed 30 survey items regarding motivation, which are presented in Appendix A. Students' responses were compiled into a paired 3x2 analysis, separating the three types of motivation (intrinsic motivation, extrinsic motivation, and task value) and the two types of assignments (programming and other course assignments). A repeated measures analysis of variance (repeated measures ANOVA) test was used to determine the statistical significance of these results. A summary of the results is given below in Figure 1 where the error bars indicate 95% confidence intervals; responses for each category spanned the entire range of the 7-point Likert scale.

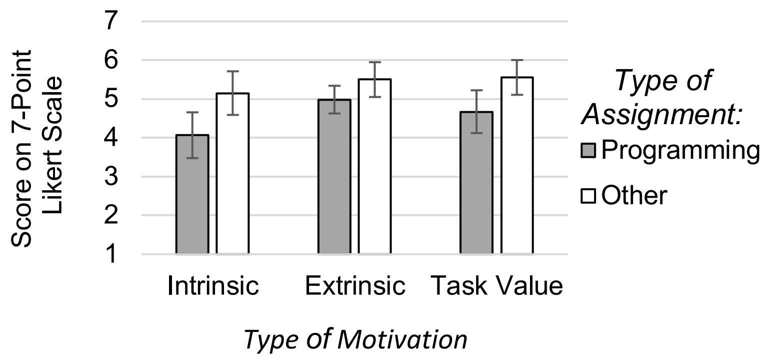

Figure 1. Summary of student motivation on programming and other course assignments from Study 1, divided by intrinsic motivation, extrinsic motivation, and task value. The error bars indicate a 95% confidence interval.

Table 2. Statistical results from Study 1. The statistical analysis method is given in parentheses for each comparison and asterisks (*) indicate results that are statistically significant ($p \leq 0.05$).

Effect of Motivation and Assignment Types (ANOVA: within subjects effects)	
Type of motivation	$p = 0.019$*
Programming/other course assignments	$p = 0.002$*
Comparison of Programming and Other Course Assignments (Paired student's t-test)	
Intrinsic motivation	$p = 0.003$*
Extrinsic motivation	$p = 0.073$
Task value	$p = 0.001$*

Students were less motivated on programming assignments than on other course assignments, with mean responses of 4.6 and 5.4, respectively; this result is statistically significant (p=0.002 from the ANOVA test). Furthermore, paired t-tests for each of the types of motivation (see Table 2) found that for both intrinsic motivation and task value, motivation was significantly lower for programming versus other course assignments; the difference for extrinsic motivation was not statistically significant. Finally, there was a significant difference between the three types of programming (p=0.019), as expected from the MSLQ validation study [14].

These results support the stated research questions. As a whole, students were less motivated on programming assignments as compared to other course assignments (like homework), in support of RQ1. This is surprising since the course homework emphasized derivations and calculations, whereas the programming assignments used hypothetical scenarios to relate students' analyses to their future careers. These began: "Your supervisor at work needs the following MATLAB analysis to be completed to benchmark some future studies…". The MATLAB scenarios were designed to enhance students' task value, whereas the homework questions were standard engineering coursework.

Additionally, paired comparisons of programming and other course assignments for the three types of motivation were conducted (RQ2). Students' external motivation was similar for programming and other course assignments. Since universities emphasize grades, it is not surprising that students would equally rate statements emphasizing good grades or showing ones' abilities to family, friends, and employers. In contrast, the intrinsic motivation statements centered around students' inner desire to learn, which could be different for the types of assignments. Similarly, the task value statements investigated whether students' perceived future use and interest in the content, which can also change.

Study 2 Results

Study 2 was completed the year after Study 1, surveying students in the same upper-division materials kinetics course. As described in the Study 2 methods, the survey was modified to only include statements about task value, removing the 18 statements regarding intrinsic and extrinsic motivation. Questions relating to students' prior programming experience were also removed. Several new questions investigated the perceived importance of writing and programming to an engineer. These survey changes were made to limit the length of the survey while also providing more specific information about students' thought processes. This second study had 16 respondents (42% response rate), which is similar to the response rate for the first survey.

Similar to Study 1, Study 2 compared students' task value of programming and other course assignments as articulated in RQ1. On a 7-point Likert scale, the average score was 5.3 and 5.5 for programming and other course assignments, respectively (see Figure 2). A paired t-test (Table 3) found that there was no significant difference between the two types of assignments ($p = 0.427$). This result is surprising since it does not match the results from Study 1, which found a significant difference ($p = 0.001$) between programming and other course assignments with scores of 4.7 and 5.6, respectively. The discussion section will address this discrepancy.

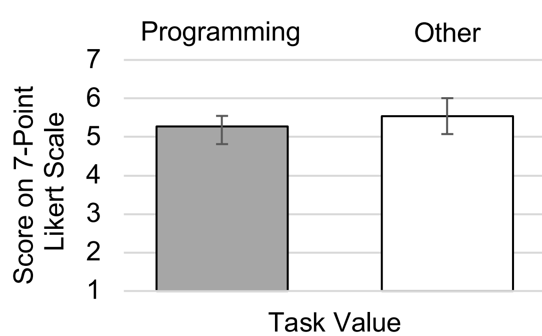

Figure 2. Comparison of task value for programming and other course assignments in Study 2. Error bars indicate a 95% confidence interval.

Table 3. Statistical results from Study 2. The statistical analysis method is given in parentheses for each comparison and asterisks (*) indicate results that are statistically significant ($p \leq 0.05$).

Comparison of Task Value (Paired student's t-test)	
Programming/other course assignments	$p = 0.427$
Effect of Skill Type and Future Use (ANOVA: within subjects effects)	
Writing/programming skill	$p = 0.397$
Future use (classes/career/non-professional areas)	$p < 0.001$*
Comparisons of Future Use of Academic Skills (ANOVA: post hoc tests with Bonferroni correction)	
Courses/career	$p = 1.000$
Courses/non-professional areas	$p = 0.002$*
Career/non-professional areas	$p = 0.001$*

Study 2 also compared students' perceived future use of "computer programming skills" and "writing skills," which are two crucial professional skills for engineers. Students rated their likelihood of using the skill in future classes, their career, or non-professional areas such as hobbies; no guidance was provided as to the definitions of "computer programming skills" or "writing skills" (see Appendix B for full prompts). The results are summarized in Figure 3, with the 2x3 ANOVA analysis given in Table 3. Overall, there was no difference between the future use of programming and writing skills ($p = 0.397$); students thought that they were equally likely to use these in the future. However, there were significant differences between the types of future use ($p < 0.001$). Post hoc tests with Bonferroni corrections were conducted to determine which of the three usages were independent of the others. Students are equally likely to use these skills in their future classes and their careers ($p = 1.000$) but are less likely to use these skills in non-professional areas of their lives ($p = 0.002$ and $p = 0.001$ for comparisons to courses and careers, respectively).

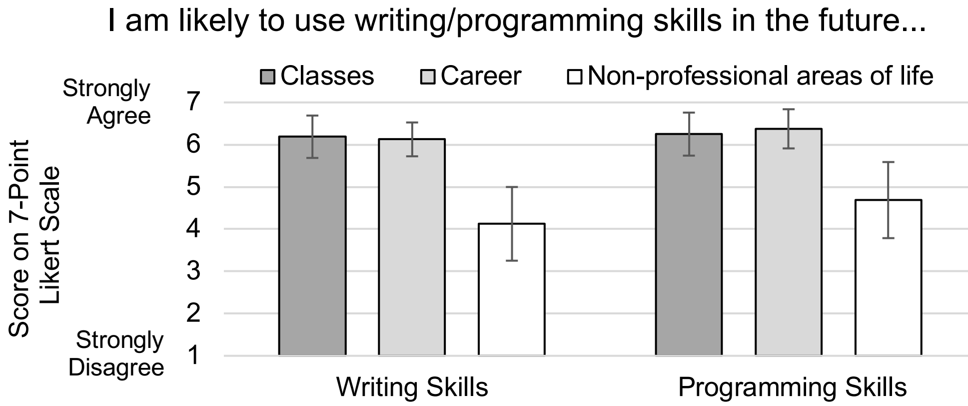

Figure 3. Students' perceptions of their future use of writing and programming skills in their future classes, career, or non-professional areas of life (such as hobbies). Error bars indicate 95% confidence intervals.

The final part of the survey investigated the perceived importance of programming skills. Specifically, students responded to the following question: "How do you think it benefits an engineer to develop computer code or programming well?" Many discussed the ubiquitousness of computers and programming for technical engineering careers (dubbed the "*technological era*" by one student). They also noted that computer programming is "*efficient*" and can be used to speed up calculations and analysis compared to methods by hand. To address RQ3, two additional themes were included: *experimental data* and *simulations*. Representative quotes for these four themes are given below:

- *Technological era:* "Literally every engineering field needs some amount of coding or programming now."
- *Efficiency:* "An engineer can whip up a program to do something in ten minutes that would have taken 5 hours manually."
- *Experimental Data:* "This is a technological era, everything depends on programming to quickly and accurately process and analyze data."
- *Simulations:* "To model anything with any accuracy, programming is necessary and powerful."

The student responses were coded for these four themes, with most respondents addressing multiple themes. As shown in Figure 4, "technological era" and "efficient" were each mentioned by 50% of respondents. Finally, more students noted the applicability of computer programming to experimental data analysis than for simulations, 38% and 19%, respectively. In response to RQ3, this data shows that MSE students are more likely to attribute programming skills to data analysis than to simulations.

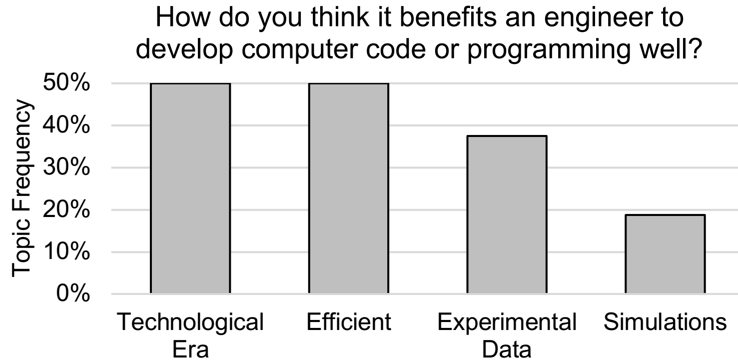

Figure 4. Fraction of students that addressed four thematic areas (technological age, efficient, experimental data, and simulations) when answering the question "How do you think it benefits an engineer to develop computer code or programming well?" Some student responses contained multiple themes, so the numbers do not add to 100%.

Overall, the results of Study 2 suggest that students see value in learning programming skills and completing programming assignments. Students are equally motivated to complete programming assignments and other course assignments in their MSE courses. Furthermore, they understand that programming skills are essential for the modern computer era and are applicable to both experimental data analysis and computer simulations. This study suggests that students are motivated to complete CMSE modules and courses.

Discussion

This paper found that students' task value on programming assignments increased from Study 1 to Study 2, which was surprising. Prior to analyzing the results from Study 2, the author was not expecting changes in students' responses. Comparison of the courses, enrollments, and surveys (see Table 1) indicates that many variables remained the same between the two studies: instructor, class size, and distribution of majors. Notably, there were no substantial changes to the curriculum during this time, either on the program or course level. The demographics of the study participants are unknown since the surveys were completed anonymously.

The author notes several differences between Studies 1 and 2. First, the survey questions were changed to provide qualitative information on students' motivations without increasing the time needed to complete the survey. The positive or negative effects of this change are unknown but expected to be small. A second difference is a shift in the gender distribution from 33% to 26% female for the courses corresponding to studies 1 and 2, respectively; the significance of this drop is unknown due to the small course sizes. Finally, the third difference was the timing of the surveys: Study 1 was conducted at the *completion* of the term whereas Study 2 was at the *beginning* of the term. Of the noted changes, the survey timing is likely to be the most significant factor accounting for the change to students' motivation. Two hypotheses have been generated to explain the contrasting results.

One hypothesis is that the assessments of students' programming motivation are affected by the timing of the survey relative to the term. Students may enter a course with high expectations about their learning but must work hard throughout the term to develop mastery. The disparity between expectations and reality can cause students' perceptions to become less favorable. For instance, Galloway and Bretz studied students' cognitive perceptions of learning across a two year chemistry lab sequence [17]. Tracking a cohort of students found that their cognitive perceptions were highest at the beginning of the first year lab and decreased over the first academic year. However, students' cognitive perceptions were "reset" at the start of the second academic year before declining again. Furthermore, a multi-institution study found that these decreases in perceived cognitive learning were common across institutions [18]. Students' changing attitudes are relevant since Studies 1 and 2 were conducted at the *end* and *beginning* of the terms, respectively. Thus, it is likely that the study results were influenced by conducting the surveys at different points in the academic year.

An alternate theory is that the survey results show the effect of evolving attitudes toward computer programming. Computer science skills are increasingly relevant to diverse disciplines such as biology and statistics. The demand for computer science classes is multiplying, as discussed in the 2019 *New York Times* article titled "Hard Part of Computer Science Class? Getting In" [19]. Studies 1 and 2 were conducted nine months apart, so it is possible that campus-wide changes to students' attitudes towards programming are affecting the motivation of MSE students.

Conclusions and Future Work

Students' motivation on programming assignments was studied for two cohorts of MSE students. However, there were conflicting results as to whether there was a difference between students' motivation on programming and other course assignments (such as homework). Study 1 found that students' programming motivation was significantly lower than their motivation for other course assignments, while Study 2 found these to be the same. It is hypothesized that these changes could be due to fluctuations in student's perceptions across a term; alternatively, the differences could be attributed to the growth of computer science for many disciplines and careers.

These mixed results highlight the need for follow-on studies to understand the evolution of students' motivation. Most notably, the author plans to conduct a longitudinal study that repeats the survey for Study 2 at the conclusion of the Winter 2019 term so that the same cohort of students is evaluated at two times relative to the term. Additionally, the repeated study will allow for a closer comparison between the two groups of students (those enrolled during the Winter 2018 and Winter 2019 terms), to see if the student responses are still different at the same point in time. Alternatively, more qualitative information can be collected through short answer questions and focus groups to elucidate students' thought processes with regards to CMSE assignments. As CMSE topics are added to the undergraduate curriculum, it is important that faculty understand our students' experiences and mindset so that our courses provide supportive learning environments.

Acknowledgments

This work was supported as a part of the Scholarship of Teaching and Learning Community at the Center for Educational Effectiveness at the University of California, Davis.

References

[1] "Modeling Across Scales: A Roadmapping Study for Connecting Materials Models and Simulations Across Length and Time Scales," The Minerals, Metals, Materials, Society, Warrendale, PA, 2015. [Online]. Available: www.tms.org/multiscalestudy

[2] "Integrated Computational Materials Engineering: A Transformational Discipline for Improved Competitiveness and National Security," National Research Council, Washington, DC, 978-0-309-11999-3, 2008. [Online]. Available: http://www.nap.edu/catalog/12199/integrated-computational-materials-engineering-a-transformational-discipline-for-improved-competitiveness

[3] R. A. Enrique, M. Asta, and K. Thornton, "Computational Materials Science and Engineering Education: An Updated Survey of Trends and Needs," *JOM,* vol. 70, no. 9, pp. 1644-1651, September 01 2018.

[4] K. Thornton, S. Nola, R. E. Garcia, M. Asta, and G. B. Olson, "Computational materials science and engineering education: A survey of trends and needs," *JOM,* vol. 61, no. 10, pp. 12-17, October 1 2009.

[5] L. Li. (June 2016). Integrating Computational Modeling Modules into Undergraduate Materials Science and Engineering Education. Presented at American Society for Engineering Education Annu. Conf. [Online]. Available: https://peer.asee.org/25791.

[6] R. Mansbach *et al.*, "Reforming an undergraduate materials science curriculum with computational modules," *Journal of Materials Education,* vol. 38, pp. 161-174, 2016.

[7] A. K. Polasik. (June 2017). Successes and Lessons Learned in an Undergraduate Computational Lab Sequence for Materials Science and Engineering. Presented at American Society for Engineering Education Annu. Conf. [Online]. Available: https://peer.asee.org/28877.

[8] C. Vieira, A. J. Magana, R. E. García, A. Jana, and M. Krafcik, "Integrating Computational Science Tools into a Thermodynamics Course," *Journal of Science Education and Technology,* vol. 27, no. 4, pp. 322-333, August 1 2018.

[9] A. J. Magana, M. L. Falk, C. Vieira, and M. J. Reese, "A case study of undergraduate engineering students' computational literacy and self-beliefs about computing in the context of authentic practices," *Computers in Human Behavior,* vol. 61, pp. 427-442, August 1 2016.

[10] A. Wigfield and J. S. Eccles, "Expectancy–Value Theory of Achievement Motivation," *Contemporary Educational Psychology,* vol. 25, no. 1, pp. 68-81, 2000.

[11] J. S. Eccles and A. Wigfield, "Motivational Beliefs, Values, and Goals," *Annual Review of Psychology,* vol. 53, no. 1, pp. 109-132, 2002.

[12] S. A. Ambrose, M. W. Bridges, M. DiPietro, M. C. Lovett, and M. K. Norman, *How Learning Works: Seven Research-Based Principles for Smart Teaching*. San Francisco, CA: Jossey-Bass, 2010.

[13] P. R. Pintrich, "A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)," 1991.

[14] P. R. Pintrich, D. A. F. Smith, T. Garcia, and W. J. Mckeachie, "Reliability and Predictive Validity of the Motivated Strategies for Learning Questionnaire (MSLQ)," *Educational and Psychological Measurement,* vol. 53, no. 3, pp. 801-813, 1993.

[15] S. Pulford. (June 2016). Useful but Not Interesting: Illuminating Student Task Values Surrounding Engineering Writing Classes. Presented at American Society for Engineering Education Annu. Conf. [Online]. Available: https://peer.asee.org/27117.

[16] S. P. Gentry. (2018). *Scaffolding Simulations in a Rate Processes of Materials Course*. Available: https://nanohub.org/resources/28793

[17] K. R. Galloway and S. L. Bretz, "Measuring Meaningful Learning in the Undergraduate General Chemistry and Organic Chemistry Laboratories: A Longitudinal Study," *Journal of Chemical Education,* vol. 92, no. 12, pp. 2019-2030, Dec. 8 2015.

[18] K. R. Galloway and S. L. Bretz, "Measuring Meaningful Learning in the Undergraduate Chemistry Laboratory: A National, Cross-Sectional Study," *Journal of Chemical Education,* vol. 92, no. 12, pp. 2006-2018, Dec. 8 2015.

[19] N. Singer, "Hard Part of Computer Science Class? Getting In," *The New York Times*, Jan. 23, 2019. Available: https://www.nytimes.com/2019/01/24/technology/computer-science-courses-college.html.

Appendix A: Survey 1

<u>*Motivation Questions*</u>

For this section, students rated the following statements for both programming and other course assignments in the target course. Students first rated the statements about programming assignments, then repeated the task for statements that used the phrase "course assignments (such as homework)" in place of "programming assignments." The order of the questions was randomized amongst a given type of assignment.

Below is a list of statements about programming assignments in engineering courses such as [the target course]. Please indicate how representative each statement is of yourself, on a scale from 1 (not at all true of me) to 7 (Very true of me). For each question, you may also select "Decline to answer."

Intrinsic Goal Orientation

- In a class like [the target course], I prefer programming assignments that really challenge me so I can learn new things.
- In a class like [the target course], I prefer programming assignments that arouse my curiosity, even if it is difficult to learn.
- The most satisfying thing for me on programming assignments is trying to understand the materials-science content as thoroughly as possible.
- The most satisfying thing for me on programming assignments is trying to understand the programming content as thoroughly as possible.
- When I have the opportunity in classes like [the target course], I choose programming assignments that I can learn from even if they don't guarantee a good grade.

Extrinsic Goal Orientation

- Getting a good grade on programming assignments is the most satisfying thing for me right now.
- The most important thing for me right now is improving my overall grade point average, so my main concern on programming assignments is getting a good grade.
- If I can, I want to get better grades on programming assignments than most of the other students.
- I want to do well on programming assignments because it is important to show my ability to my family, friends, employer, or others.

Task Value

- I think I will be able to use what I learned on programming assignments in [the target course] in other courses.
- It is important for me to learn the material for the [target course] programming assignments.
- I am very interested in the content area of the [the target course] programming assignments.
- I think the material for the [the target course] programming assignments is useful for me to learn.
- I like the subject matter of the [the target course] programming assignments.
- Understanding the subject matter of the [the target course] programming assignments is very important to me.

Student Background:

- Prior to [the target course], had you taken an introductory programming course (such as [list of introductory programming courses], or equivalent)?
    - Yes
    - No
    - Decline to answer
- Prior to [the target course], how many college-level courses had you taken which required programming (either on assignments or as the focus of the class)?
    - 1
    - 2
    - 3+
    - Decline to answer
- Prior to [the target course], had you taken a course which required the use of MATLAB as a programming language?
    - Yes
    - No
    - Decline to answer
- Prior to taking [the target course], which of the following programming languages had you used with sufficient proficiency that you could set up arrays, use loops and logic (if/then) statements, and perform calculations with numbers? Select all that apply.
    - C
    - C++
    - MATLAB
    - Python
    - Java
    - Other language(s)
    - Decline to answer

Appendix B: Survey 2

<u>*Motivation Questions*</u>

For this section, students rated the following statements for both programming and other course assignments in the target course. Students first rated the statements about programming assignments, then repeated the task for statements that used the phrase "course assignments (such as homework)" in place of "programming assignments." The order of the questions was randomized amongst a given type of assignment.

Below is a list of statements about programming assignments in engineering courses such as EMS 164. Please indicate how representative each statement is of yourself, on a scale from 1 (not at all true of me) to 7 (Very true of me). For each question, you may also select "Decline to answer."

Task Value

- I think I will be able to use what I learned on programming assignments in EMS 164 in other courses.
- It is important for me to learn the material for the EMS 164 programming assignments.
- I am very interested in the content area of the EMS 164 programming assignments.
- I think the material for the EMS 164 programming assignments is useful for me to learn.
- I like the subject matter of the EMS 164 programming assignments.
- Understanding the subject matter of the EMS 164 programming assignments is very important to me.

<u>*Future Use of Skills*</u>

On a scale from 1 (Strongly Disagree) to 7 (Strongly Agree), rate how likely you are to use computer programming or writing skills in the future.

- I am likely to use computer programming skills in future classes.
- I am likely to use computer programming skills in my future career.
- I am likely to use computer programming skills in the future in non-professional areas of my life (such as hobbies).
- I am likely to use writing skills in future classes.
- I am likely to use writing skills in my future career.
- I am likely to use writing skills in the future in non-professional areas of my life (such as hobbies).

<u>*Short Answer*</u>

- How do you think it benefits an engineer to write well?
- How do you think it benefits an engineer to develop computer code or programming well?