*Journal of Materials Education Vol.38 (3-4): 161-174 (2016)*

# REFORMING AN UNDERGRADUATE MATERIALS SCIENCE CURRICULUM WITH COMPUTATIONAL MODULES

R. Mansbach[1], A. Ferguson[2,3], K. Kilian[2], J. Krogstad[2], C. Leal[2], A. Schleife[2], D. R. Trinkle[2], M. West[4], and G. L. Herman[5,*]

[1] Dept. of Physics, University of Illinois at Urbana-Champaign, 1110 W. Green St., Urbana, IL 61801, mansbach2@illinois.edu,
[2] Dept. of Materials Science and Engineering, University of Illinois at Urbana-Champaign, 1304 W. Green St., Urbana, IL 61801, alf@illinois.edu, kakilian@illinois.edu, jakrogst@illinois.edu, cecilial@illinois.edu, schleife@illinois.edu, dtrinkle@illinois.edu;
[3] Dept. of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, 114 Roger Adams Laboratory, 600 S. Mathews Ave., Urbana, IL 61801, alf@illinois.edu;
[4] Dept. of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, 1206 W. Green St., Urbana, IL 61801, mwest@illninois.edu,
[5] Illinois Foundry for Innovation in Engineering Education, University of Illinois at Urbana-Champaign, 1304 W. Green St., Urbana, IL 61801, glherman@illinois.edu *[*Corresponding Author]*

## ABSTRACT

Computational competencies such as the use of modeling and simulation tools are a new core literacy that students in Materials Science and Engineering must develop. To develop this literacy among our students, the Department of Materials Science and Engineering at the University of Illinois at Urbana-Champaign is synthesizing computational tools and skills across the core curriculum. In this paper, we describe the collaborative process for changing courses and curriculum, describe the outcome of these reforms, and provide evidence that these reforms have enhanced student learning.

**Keywords:** *Computation; course redesign; curriculum redesign, exams, student attitudes*

## INTRODUCTION AND BACKGROUND

As the use of simulations, big data, and numerical methods increases, the engineers of the future will increasingly be expected to possess computational competencies to not only perform well on the job, but to even understand the complex systems that govern the problems in their disciplines[1-4]. Computational competencies such as programming and the use of modeling and simulation tools are becoming core forms of literacy for most engineers on par with mathematics and the engineering sciences[1-3,5,6]. The 2011 White House Materials Genome Initiative has created a particular imperative for computational competencies in Materials Science and Engineering, creating a demand for students who can engage in the computer-aided design of materials[7].

Meeting this new demand for computational competencies is not straightforward—simply adding new skills and content independent of the traditional content is not viable in already packed curricula. To add these new competencies, we must either teach a smaller technical core to create space or find ways to synergize computational instruction with instruction in traditional content so that learning computational competencies accelerates learning of traditional content and vice versa[1]. Fortunately, other work suggests that integrating the use of modeling and simulation tools into instruction can foster deeper understanding of complex engineering concepts and problems[2,8,9]. In particular, such tools are useful for helping students understand microscopic or abstract phenomena, such as those found throughout a materials curriculum[10].

The Department of Materials Science and Engineering (MatSE) at the University of Illinois at Urbana-Champaign is synthesizing computational tools and skills across the curriculum. Over two years using a collaborative course-development approach, a team of six faculty (one tenured professor and five assistant professors) have integrated training in computational competencies across five courses (MSE 201 – Phases and Phase Relations, MSE 206 – Mechanics for MatSE, MSE 304 – Electronic Properties of Materials, MSE 406 – Thermal and Mechanical Behavior of Materials, MSE 498AF – Computational MatSE: course syllabi and other resources are publicly available at http://bit.ly/25OzbzG or may be requested from the authors). In this paper, we first describe the process for creating this curriculum revision and then describe the teaching methods and assignments of the revised courses. We conclude by presenting evidence for the effectiveness of this reform effort by presenting both examination data and student survey data.

## APPROACH TO COURSE AND CURRICULAR REFORM

The College of Engineering's Strategic Instructional Initiatives Program (SIIP) was created to transform and revitalize core engineering courses[11-13]. Over the past three years, the program has catalyzed innovation in most departments and large-enrollment, core courses in the college. Inspired by the work of Henderson et al.[14-17], SIIP was designed to focus on creating collaborative teaching environments that enabled faculty to iteratively and sustainably innovate instruction[11]. This environment was created by organizing faculty into Communities of Practice (CoPs) that would choose what innovations to pursue and evaluate their efforts to create those innovations[18]. A CoP creates a community to collaboratively explore a domain of knowledge to support the development of practice[19,20]. In this case, the MatSE faculty are the community and the implementation and evaluation of computational competency curriculum is the practice. A CoP is an organizational structure that effectively spreads knowledge, decreases the learning curve for novices, minimizes reenactments of failures, and promotes creativity[19,20]. The use of a CoP is promoting buy-in for course reforms, facilitating the continued use of developed resources when instructors change. (Table 1 shows how multiple course instructors were used across semesters in some courses.).

Table 1. List of which faculty taught which courses each semester

| | **Fall 2013** | **Spring 2014** | **Fall 2014** | **Spring 2015** |
|---|---|---|---|---|
| MSE 201 | Leal | Kilian | Leal | Kilian |
| MSE 206 | | Trinkle | | Krogstad |
| MSE 304 | | (Weaver) | | Schleife |
| MSE 406 | Trinkle | | Trinkle | |
| MSE 498 | Ferguson | | Ferguson | |

The MatSE CoP is composed of one tenured and five tenure-track faculty who meet on a weekly basis to discuss course administration, data collection, and future plans. The goal of these meetings is to develop a common set of resources, policies, teaching methods, and learning objectives (described in “Pedagogical Reforms in Courses”) across the courses to facilitate students’ computational competencies and technical content knowledge across the targeted course sequence. The use of common learning objectives, exam questions, and teaching methods additionally enables longitudinal studies of whether reforms are improving student learning.

The revisions to the MatSE undergraduate curriculum were guided by two curriculum and course reform aims: (1) integrating computational materials modeling in sophomore and junior-level core courses and (2) developing a capstone senior materials modeling elective.

The integration of computational materials with technical content took place in MSE 201, MSE 206, MSE 304, and MSE 406, each of which has 60-100+ students enrolled each semester. Together, these courses span three broad areas of materials science: mechanics, thermodynamics, and electronic properties. The longitudinal integration of computational modules across the sophomore and junior years was intended to reinforce student awareness of computation, build confidence in using computational tools, and cement the idea of computation as the "third pillar" of science alongside experiment and theory. Accordingly, we expected that this integration would (a) make abstract theoretical concepts more accessible, (b) promote active learning and hands-on engagement, and (c) develop student competency in computational materials science software tools.

The second aim of this effort was to develop a new senior-year computational materials science elective MSE 498. The course was conceived as an integrated computational materials science and engineering capstone design course to tie together students’ experiences in the other courses. In this course, students solve a materials engineering design problem at multiple length and time scales using a diversity of software packages and computational tools, gaining broad experience and confidence in industrially relevant MatSE software packages and a first-hand appreciation for the power and limitations of computational methods.

Team members have committed to recording and hosting all computational modules, lectures, and course forums online to facilitate access and dissemination of these materials.

## PEDAGOGICAL REFORMS IN COURSES

Aside from incorporating computation across the curriculum, course reforms also focused on integrating evidence-based instructional practices into the courses[21-23]. Pedagogical reforms focused on integrating classroom response systems (i>clickers), tablets for presenting content, online homework for rapid feedback, and discussion to promote deeper thinking and learning. Table 2 shows which courses implemented which instructional practices.

Table 2. List of which reforms were implemented in each course.

| | **i>clickers** | **Tablets** | **Computation** | **Online Homework** | **Discussion Sections** |
|---|---|---|---|---|---|
| MSE 201 | X | X | X | | |
| MSE 206 | X | X | X | X | X |
| MSE 304 | X | X | X | X | |
| MSE 406 | X | X | X | X | X |
| MSE 498 | | | X | | |

All of the required courses (i.e., 201, 206, 304, 406) used i>clicker for electronic polling of students during lectures to facilitate problem solving and active learning[21,22]. In the case of 201, i>clickers were used during lecture for active learning, and as a replacement for paper quizzes at the end of class. In 206 and 406, i>clickers replaced out-of-class reading quizzes with in-class problem solving and active learning using “think-pair-share.” In 304, i>clickers were used during lecture for active learning, for testing students’ understanding of the material that was presented in the previous class, and, on occasion, for in-class student problem solving. 304 also used pre-lecture questions before each class to evaluate student knowledge of the topic and to encourage students to study the reading material. In 206 and 406, some i>clicker questions were repurposed for randomized multiple-choice exam questions. As all course lectures were both recorded and slides posted online after, the content was available for student review following lectures.

All of the required courses used tablet devices to project lecture slides along with handwritten live problem solving. The lecture slides were prepared with different amounts of typed content: 206 and 406 had the largest amount of computer typed content, with handwritten content comprising live problem solving based on i>clicker questions. At the other end, 201 used more handwritten notes à la an electronic white board. 304 was in between with slides that mixed computer-typed with hand-written content, and the tablet was used to develop/explain graphs and derive equations. In all cases, final annotated slides were posted for student access, as well as full lecture capture: video of projected slides and lecture audio.

Half of the homework sets in MSE 304 and all of the homework sets in 206 and 406 were replaced by online homework, that allow submission of numerical answers to questions that were based on randomized input numbers for each student. Students had multiple tries to answer each question correctly and received feedback immediately upon entering an answer. Students were encouraged to solve each problem symbolically first and only insert the input numbers in the last step to obtain the final (numerical) answer, along with the correct units. MSE 206 used material from Pearson “Mastering Engineering” while both 304 and 406 relied on faculty generated homework assignments.

Two of the courses (206 and 406) used discussion sections for group learning. In both classes, multiple 50-inute discussion sections were created to ensure class sizes of 40 or fewer, with at least two teaching assistants in the room for a group/TA ratio of 5. Each class used groups of four, with random group creation each week for 206, and persistent groups created using Comprehensive Assessment of Team Member Effectiveness (CATME)[24,25] for 406. Each discussion section was given a single long “Engineering Design” or “Engineering Analysis” problem based on the topics from the past week to apply to a real-world problem in a group setting. The TAs assisted with questions about how to approach problems, and to ensure that groups were working together. Grading for the discussions was done on a group basis by randomly selecting one group solution for each group. The grade for that solution was given to the entire group; the grading is based primarily on “effort.” Since the questions were complex and multipart, as long as students were able to show reasoning towards the answer and application of engineering principles, they received full credit, even in cases where their solution was only partial. As the TAs engaged with groups in their efforts, most groups received full credit for their participation.

## DESCRIPTION OF COMPUTATIONAL TOOLS AND MODULES

The computational modules we have developed target four prime areas of computational materials science at different length scales using popular software packages: (i) density functional theory (DFT) with Quantum Espresso[26], (ii) molecular dynamics (MD) with LAMMPS[27] and Gromacs[28], (iii) finite element method (FEM) modeling with OOF2[29], and (iv) thermodynamic calculation of phase diagrams (CALPHAD) using Thermo-Calc[30]. By longitudinal integration of the modules into the core undergraduate curriculum, students will be repeatedly exposed to computational content over their academic trajectory at increasing levels of difficulty and complexity, ultimately preparing them for a capstone senior integrated computational materials engineering experience.

Each class has 2-3 computational modules associated with it. The current basic structure of a module is as follows: First, the subject, background, and tools of the module are introduced during a class lecture. Then, the module is given as a homework assignment, which students are expected to complete over the course of 1-2 weeks, with the aid of a dedicated computational TA, who holds 2-4 sessions of office hours in a computer lab that is accessible 24/7, in which the required software has been installed.

In Table 3 and the sections below, we briefly describe the particular modules developed, and their deployment in the target courses. The modules are described in order of increasing complexity as the students would experience them progressing through the curriculum.

**Matlab**

Beam Design. Students in MSE 206 used Matlab to numerically determine the bending moment of differently-shaped beams in order to predict the most appropriate geometry with the goal of minimizing the stress a beam experienced under load.

**Density Functional Theory (DFT)**

Si crystal. Using the Quantum Espresso software with a GUI provided by nanohub.org[31], students in MSE 201 were asked to compute the equilibrium lattice constants of silicon for three different crystal structures using plane wave self-consistent field (PWSCF) calculations. Building on this module, students in MSE 304 and MSE 498 were asked to calculate the bulk modulus of silicon from pressure perturbations to the lattice constant and to calculate and visualize the band structure of silicon and compare the computed band gap property with experiment. As another extension, students in MSE 498 were asked to perform geometry relaxation and energy convergence with respect to the plane wave cutoff and k-point sampling, and explore the effect of different exchange correlation functionals and bound electron pseudopotentials.

**Molecular Dynamics (MD)**

Properties of Al. Students in MSE 406 used the LAMMPS software package to investigate the movement of a dislocation through a solid block of aluminum. They used the stress-strain curve to predict the Peierls stress of a dislocation, and the Ovito software package to visualize the movement and the change in stress-strain over the course of the simulation. As an extension, students in MSE 498 also predicted the Young's Modulus, and used both pieces of information to parameterize a finite element simulation, demonstrating the construction of an Integrated Computational Materials and Engineering (ICME) bridge from one level of simulation to the next.

Table 3. List of which computational modules were deployed in each course.

| | DFT | MD | FEM | CALPHAD | Matlab |
|---|---|---|---|---|---|
| MSE 201 | X | | | X | |
| MSE 206 | | | X | | X |
| MSE 304 | X | | | | |
| MSE 406 | | X | X | | |
| MSE 498 | X | X | X | X | X |

Nonequilibrium Folding. Students in MSE 498 used the Gromacs software package to perform a nonequilibrium pulling simulation of the unfolding of a ß-hairpin protein and to estimate the work required for the unfolding to occur.

**Finite Element Method (FEM)**

Temperature effects on strain. Students in MSE 206 used the OOF2 software package on nanohub.org to investigate the effects of geometry on a system of steel pins holding a dog-bone-shaped aluminum sample. Students solved the coupled heat flux and force balance equations over a finite element mesh to compute the temperature and stress fields over the strip and predict its deflection. They then compared the stress patterns in systems with differently-shaped pins. As an extension, students in MSE 498 used Matlab to develop their own implementation of finite element software to solve the one-dimensional heat equation.

Nanocomposites. Students in MSE 406 used the OOF2 software package to explore the effects of fibers on strain and bulk modulus in a composite. They solved the force balance equations over a finite element mesh, in which applied strains were perpendicular and parallel to the direction of fibers along a composite. They investigated the effects of changing the Young's modulus of the fibers and of the matrix and visualized the resulting stress distribution.

Stress Field of a Crack. Students in MSE 406 used the OOF2 software package to explore the stress distribution around a crack tip. They solved the force balance equations over a finite element mesh for systems of a narrow and blunt crack and visualized the results. As a first extension, students in MSE 406 compared the results of the OOF2 simulation with the results obtained from performing a LAMMPS molecular dynamics simulation of crack propagation in aluminum and visualizing the dynamic stress distribution using Ovito. This module demonstrated the strengths and weaknesses of the two different software packages to the students. As a second extension, students in MSE 498 used the stress field at the tip of the crack to determine whether or not crack propagation would occur.

**Calculation of Phase Diagrams (CALPHAD)**

Ag-Sn-Cu phase diagram. Students in MSE 201 used the Thermo-Calc software package to compute the $T - x$ phase diagram for each of the binary alloys and then computed the ternary phase diagram qualitatively by hand, as a demonstration of the design of an alloy for soldering applications.

Steel phase diagram and design. Students in MSE 498 used the Thermo-Calc software package to compute the $T - x$ phase diagram for a Fe-C carbon steel, and used this diagram to design an equilibrium microstructure with desired materials properties, computed the maximum operating temperature of their steel as a function of composition, and predicted the equilibrium fractions of pearlite and pro-eutectoid α-ferrite / cementite for eutectoid, hypoeutectoid, and hypereutectoid steels. Secondly, students in MSE 498 computed the ternary phase diagram for a Fe-C-Cr martensitic stainless steel, and determined an appropriate level of case hardening by surface carburization to trade-off competing constraints of hardness, toughness, and melting point to design a case hardened steel optimized for a particular application.

**Capstone Integrated Computational Materials Engineering (ICME) Elective**

MSE 498 AF: Computational Materials Science and Engineering is a new course introduced into the MatSE curriculum in Fall 2013 as the department's first dedicated Computational Materials Science and Engineering (CMSE) course offering. This course was revamped as a capstone Integrated Computational Materials Engineering (ICME) course that ties together in a single senior year elective the range of computational tools embedded into the sophomore and junior core courses, and stressing the idea of linking together computational predictive tools at a hierarchy of length and time scales. The course meets twice a week for 90 minutes in a computer lab to provide students with hands-on experience with the software, and is organized into five units: (a) bash/Matlab, (b) density functional theory (DFT) using Quantum Espresso, (c) molecular dynamics (MD) using LAMMPS, (d) finite element method (FEM) using OOF2, and (e) calculation of phase diagrams (CALPHAD) using Thermo-Calc. Each unit teaches students the theoretical underpinnings of the methodology, trains them to operate the software through a worked example, and finally tests their ability to apply their skills to use the software to solve a materials analysis or design project. Assessment is based on multiple-choice quizzes on the theory and algorithms, and individual projects to display competence with the software itself. This structure provides students with an immersive computational experience that equips both experimentalists and theorists with the skills to deploy computational software packages as practical tools to tackle materials science problems. This model also enhances intrinsic motivation by providing autonomy, purpose, and competency as computational materials engineers, and was very positively received by the students, with high reported satisfaction with the course. Video captured lectures and course materials were hosted online, and made available for free public viewing and download at nanohub.org, (*https://nanohub.org/resources/22124*).

## EVALUATION OF IMPACT OF CURRICULUM CHANGES

The impact of the curriculum changes is being evaluated by tracking student performance on exams and students' perceptions of the usefulness of computational tools and their comfort using them.

### Student performance on examinations

In this section, we present data on the impact of the curriculum changes on students' exam scores. We focus only on MSE 201 and 206 because these two courses were the only ones taught by at least two different members of the CoP and had similar enough exams between semesters to facilitate valid comparisons of student performance across semesters. Performance on the examinations between semesters was compared using a Classical Test Theory approach[32,33].

From Classical Test Theory, we can estimate the quality of the examination by measuring its reliability with Cronbach's α[34] and measuring the difficulty and discrimination of each item in the examination[35,36]. Cronbach's α for dichotomously scored test items (i.e., items are scored 1 for correct, 0 for incorrect) is given in Eq. 1, where K is the number of exam items, $P_i$ is the proportion of students that scored 1 on item $i$, $Q_i = 1 - P_i$, and $\sigma_x^2$ is the variance of all test scores.

$$\alpha = \frac{K}{K-1}\left(1 - \frac{\sum_{i=1}^{K} P_i Q_i}{\sigma_x^2}\right) \tag{1}$$

Cronbach's α provides an estimate for the likelihood that a student would receive the same score if they took the same examination twice. Coefficients of α range from 0 to 1 with 0 being a perfectly unreliable test and 1 being a perfectly reliable test. A Cronbach α of 0.6 is considered acceptable for classroom assessments such as examinations and a Cronbach above 0.8 is acceptable for rigorous research studies[33,37]. If an item is removed from the examination, then Cronbach's α should always decrease. If removing an item causes Cronbach's α to increase, then there is evidence that the item does not measure the same construct as the rest of the examination and should be removed from the examination.

To determine whether an examination provides a valid assessment of student knowledge, we should see a range of difficulty in examination items and all items should positively discriminate between strong and weak students[37]. Difficulty is measured as the percentage of students who answered an item correctly (0 is an impossible item and 1 a trivial item)[33]. Discrimination is the point-biserial (Pearson) correlation coefficient between students answering an individual item correctly and their overall score on the examination[33]. All discrimination coefficients should be positive, with values greater than 0.2 being particularly desirable[37].

*MSE 206 Final Examinations*

For MSE 206, final examination data was collected from the Spring 2014 and Spring 2015 semesters. Both examinations had 45 multiple-choice items (questions), of which 31 items (68%) were identical between semesters except for changes in the numbers used for calculations (i.e., the same figures and calculations could be used to solve the problem). Of these 31 items, 25 items (56%) were perfectly identical between semesters. For this analysis, all items were scored dichotomously (assigned a 0 for wrong, 1 for correct) so that a maximum score is 45 points.

Descriptive statistics (N for sample size, $\mu$ for mean, $\sigma$ for standard deviation, and $\alpha$ for Cronbach's $\alpha$) for both examinations are presented in Table 4. During Spring 2014, 118 students took the final examination and 102 students took the final examination during Spring 2015. Both examinations had excellent reliability with each Cronbach $\alpha$ above 0.85, making the results reliable enough for research purposes. Both examinations had questions that spanned a wide range of difficulties with an average difficulty of 0.75. All items had acceptable (positive) discrimination with at least 40 items per examination having discrimination above 0.2.

For all comparisons of performance between semesters, we used a 2-tailed t-test with a p-value of 0.05 as the threshold for significance and rejecting the null hypothesis. If a difference between course offerings is described as significant, it should be interpreted as $p < 0.05$. Effect sizes were measured using Cohen's d $\left(d = \frac{\mu_1 - \mu_0}{\sigma}\right)$ as is appropriate for comparing mean performance between two populations. We use the following thresholds for effect sizes: Cohen's d less than 0.4 is a small effect size, Cohen's d between 0.4 and 0.6 is a moderate effect size, Cohen's d above 0.6 is a large effect size.

Using all examination items, we found that students performed significantly better ($p<0.01$) in Spring 2015 than in Spring 2014 with a moderate effect size ($d = 0.44$).

To make sure that the difference in performance was not an artifact of differences in performance on the non-identical items, we repeated the above analysis on only the 25 perfectly identical items between semesters. Note that Cronbach's $\alpha$ and discrimination coefficients are test dependent, changing when the selection of items is changed. Descriptive statistics of this subtest are presented in Table 5. With only these 25 items, the Cronbach $\alpha$ of both exams remained excellent (>0.80), making the results reliable enough for research purposes. Both examinations had questions that spanned a wide range of difficulties with an average difficulty of 0.77. All items had acceptable (positive) discrimination with at least 23 items per examination having discrimination above 0.2.

Using only the perfectly identical examination items, we found that students still performed significantly better ($p<0.01$) in Spring 2015 than in Spring 2014 with a moderate effect size ($d = 0.50$).

*MSE 201 Final Examinations*

For MSE 201, final examination data was collected from the Fall 2013 and Fall 2014 semesters. Both examinations had 17 items (questions), of which 14 items were written to test the same concepts. For this analysis, all items were scored with a minimum score of 0 and maximum score of 1. We present only an analysis of those items that were intended to test the same conceptual content, so a maximum score is 14 points.

Descriptive statistics for both examinations are presented in Table 6. During Fall 2013, 48 students took the final examination and 57 students took the final examination during Fall 2014. Both examinations had excellent

Table 4. MSE 206 students' scores on all exam questions by semester

| | Population | Mean | Standard Deviation | Cronbach α | Difficulty Range | Discrimination Range |
|---|---|---|---|---|---|---|
| SP14 | 118 | 32.3 | 0.63 | 0.88 | 0.18-0.98 | 0.08-0.65 |
| SP15 | 102 | 35.3 | 0.62 | 0.86 | 0.49-0.99 | 0.05-0.62 |

Table 5. MSE 206 students' scores on all exam questions that were identical between semesters

| | Population | Mean | Standard Deviation | Cronbach α | Difficulty Range | Discrimination Range |
|---|---|---|---|---|---|---|
| SP14 | 118 | 18.3 | 0.37 | 0.80 | 0.23-0.98 | 0.10-0.69 |
| SP15 | 102 | 20.3 | 0.37 | 0.80 | 0.49-0.99 | 0.14-0.65 |

Table 6. MSE 201 students' scores on all exam questions by semester

| | Population | Mean | Standard Deviation | Cronbach α | Difficulty Range | Discrimination Range |
|---|---|---|---|---|---|---|
| FA13 | 48 | 12.07 | 0.22 | 0.80 | 0.55-1.00 | 0.43-0.78 |
| FA14 | 57 | 12.95 | 0.23 | 0.89 | 0.85-0.97 | 0.04-0.88 |

Table 7. MSE 206 students' scores on all computationally oriented exam questions by semester

| | Population | Mean | Standard Deviation | Cronbach α | Difficulty Range | Discrimination Range |
|---|---|---|---|---|---|---|
| SP14 | 118 | 3.6 | 0.11 | 0.59 | 0.44-0.92 | 0.54-0.76 |
| SP15 | 102 | 3.9 | 0.11 | 0.54 | 0.54-0.99 | 0.18-0.76 |

Table 7. MSE 201 students' scores on all computationally oriented exam questions by semester

| | Population | Mean | Standard Deviation | Cronbach α | Difficulty Range | Discrimination Range |
|---|---|---|---|---|---|---|
| SP14 | 48 | 2.7 | 0.06 | 0.53 | 0.77-0.90 | 0.60-0.90 |
| SP15 | 57 | 2.8 | 0.04 | 0.26 | 0.86-0.88 | 0.34-0.89 |

reliability with each Cronbach α above 0.8, making the results reliable enough for research purposes. Both examinations had questions that spanned a wide range of difficulties, but students were more likely to get each question right than wrong. All items had acceptable (positive) discrimination with all but 1 item having discrimination above 0.2.

We found that students performed significantly better ($p = 0.02$) in Fall 2014 than in Fall 2013 with a moderate effect size ($d = 0.55$).

**Student performance on computationally related questions**

A core goal of the evaluation was to determine whether adding computational modules improved students' understanding of core content. To measure this effect, we repeated the above analysis examining only those items that assess students' knowledge of content covered by the computational modules in each of the courses.

In MSE 206, five items pertained directly to the content covered by the computational modules. Descriptive statistics of this subtest are presented in Table 7. With only these 5 items, the Cronbach α of both subtests were close to the level needed for classroom assessments (~0.60), making the results less reliable than desired. Both subtests had questions that spanned a moderate range of difficulties, ranging from moderate to easy. All items had acceptable (positive) discrimination with all but one item having discrimination above 0.2.

Using only the items that covered content related to the computation modules, we found that students performed significantly better ($p=0.03$) in Spring 2015 than in Spring 2014 with a small effect size ($d = 0.30$).

In MSE 201, three items pertained directly to the content covered by the computational modules. Descriptive statistics of this subtest are presented in Table 7. With only these three items, the Cronbach α of these exams were lower than the level needed for classroom assessments (~0.60), making the results less reliable than desired. Both subtests had questions that spanned only a small range of difficulties, ranging from moderate to easy. All items had acceptable (positive) discrimination with all items having discrimination above 0.2.

Using only the items that covered content related to the computation modules, we found no significant difference ($p=0.14$) in performance between semesters.

### Discussion of student learning outcomes

Results from MSE 201 and 206 suggest that the combination of pedagogical changes and addition of computational modules has improved students' learning outcomes in MSE 201 and 206. The reform efforts revealed significant improvements in exam scores with moderate effect sizes. According to Classical Test Theory analysis, the exams in these courses were excellent exams that provided reliable and valid measurements of students' learning, strengthening the claim that course reforms improved student learning. These improvements are robust across courses, minimizing the likelihood that the changes are dependent on changes in instructors[38]. The improvements in student performance cannot be explained by students' access to previous exam questions either as the improvements in students' learning is robust across identical and non-identical exam items.

It is not clear from the data whether the improvement in students' learning was caused primarily by the pedagogical changes or the addition of computation modules. The subtests of computation questions were insufficiently reliable to draw strong conclusions. The data suggests that the computation modules may play a role in improving student learning, but the results are not robust across courses. At minimum, though, the addition of the computation modules did not undermine or hinder students' learning of the core disciplinary content. Critically, students were able to learn more content (i.e., students learned both computation modules and the traditional content) because of the reforms.

## Student survey data

To study the impact of our curriculum change on students' attitudes toward computation, we administered surveys to measure students' attitudes and competency beliefs before and after the curriculum change.

### Students' perceptions of the value of computational tools

To measure students' perceptions of the value of computational tools, we administered the same survey question in MSE 201, 206, 304, and 406. In total, 397 students took the survey before the curriculum change and 486 students took the survey after the curriculum change. Students were asked two, 5-point Likert scale items.

> "I think computational materials science skills are important for my post-graduation career"
> (5 - Strongly Agree to 1 - Strongly Disagree)
>
> "I would like to use computation in my MatSE classes…"
> (5- Much More to 1 - Much Less)

Table 8. Students' responses to survey questions "I think computational materials science skills are important for my post-graduation career" (5 - Strongly Agree to 1 - Strongly Disagree), "I would like to use computation in my MatSE classes…" (5- Much More to 1 - Much Less).

| | Important Skill | More Computation |
|---|---|---|
| Mean rating before | 4.15 | 3.93 |
| Mean rating after | 4.20 | 3.83 |
| t-test p-value | 0.46 | 0.13 |

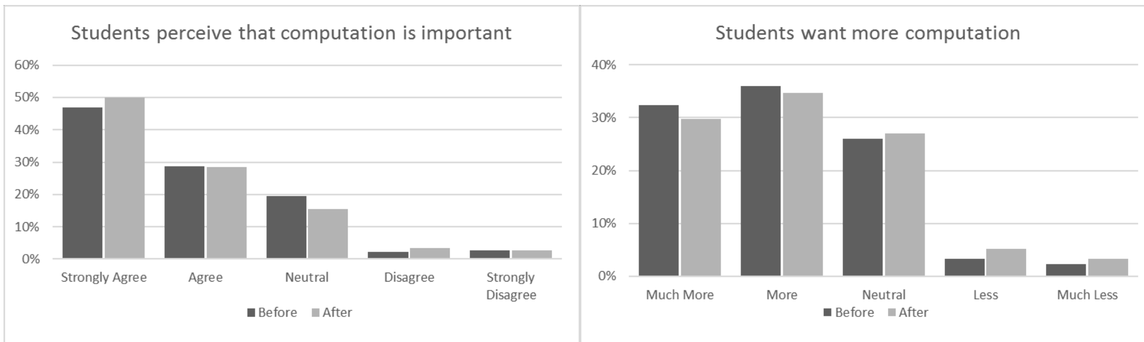


Figure 1. Distribution of students' responses to survey questions in Table 8

Students' responses to these two survey questions are shown in Table 8 and Figure 1. Comparisons between students' Likert scale ratings were performed using a t-test. These results show that the curriculum revisions had no significant impact on students' perception of the importance of computational skills or their desire for more computation in their MatSE classes.

**Students' sense of ability with computational abilities**

To measure students' sense of ability or competence with computational abilities, we asked students to rate their level of comfort with performing specific calculations using specific computation tools. Each calculation and tool was chosen to reflect the calculations and tools of each course. For MSE 201, we asked students (N=100 before the change and 196 after the change) to rate their comfort with "calculating a eutectic point with CALPHAD." For MSE 206, we asked students (N=94 before the change, 96 after the change), to rate their comfort with "calculating beam-bending with FEM." For MSE 304, we asked students (N=101 before the change, 99 after the change) to rate their comfort with "calculating band structure with DFT." Students rated their comfort using a 5-point Likert scale (5 – Very comfortable, 1 – Very Uncomfortable).

Students' responses to these survey questions are shown in Table 9 and Figure 2. Comparisons between students' Likert scale ratings were performed using a t-test. These results show that the curriculum revisions had a significant, positive impact on students' beliefs that they could use the computation tools related to their course content.

**Discussion of student attitudes**

The surveys reveal that adding instruction on computational tools does not by itself improve students' perceptions of the usefulness of computation or their desire to learn more about computational tools. The lack of change has two possible explanations. First, students' perceptions may not have changed because students frequently perceive that what they do in

Table 9. Students' ratings of their comfort with various computational tools used in MSE 201 (CALPHAD), MSE 206 (FEM), MSE 304 (DFT).

| | Comfort with CALPHAD | Comfort with FEM | Comfort with DFT |
|---|---|---|---|
| Mean rating before | 1.03 | 1.19 | 1.35 |
| Mean rating after | 2.86 | 1.96 | 3.69 |
| t-test p-value | <0.01 | <0.01 | <0.01 |

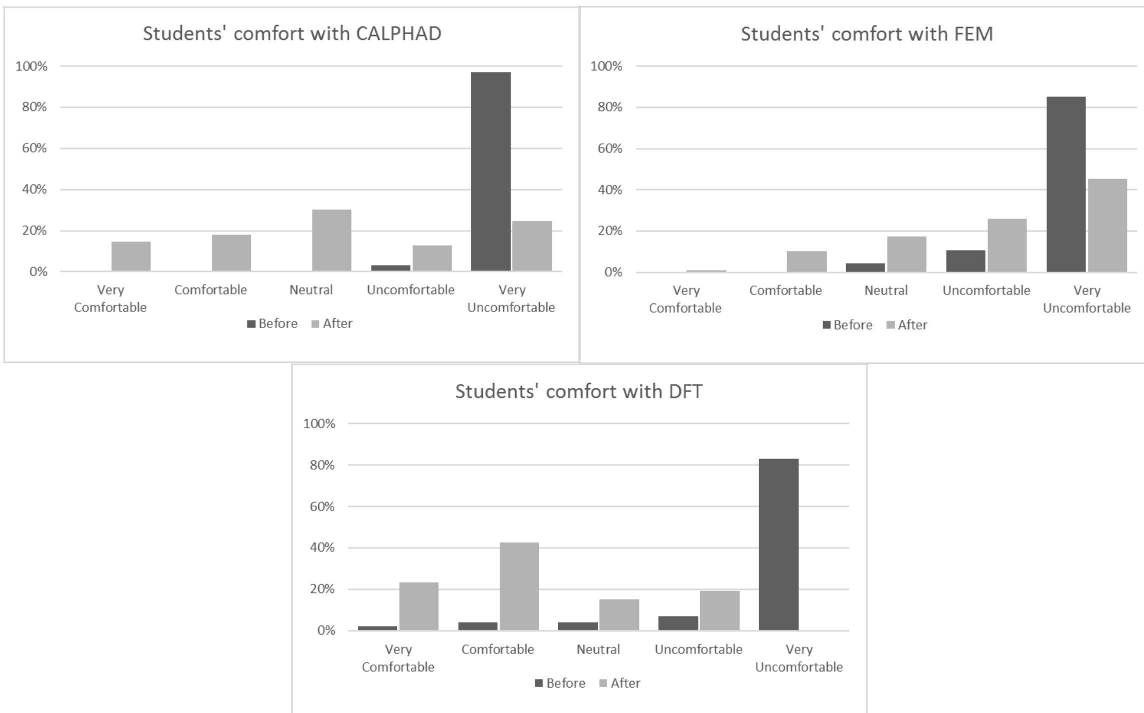


Figure 2. Distribution of students' comfort with various computational tools

the classroom and in school is somewhat disconnected from what they will do in their careers. For example, while the faculty have first-hand experience with the importance of computational tools in Materials Science, students, lacking this experience, must simply trust what their instructors say. Consequently, students' responses to these questions may simply reflect their general attitudes about the relevance and importance of what they are learning in the classroom. Alternatively, students' responses to these survey questions may simply reflect affirmation bias. Without a strong pre-existing belief about the importance of computation, the students' may have answered with affirming responses simply to make the survey writer happy.

While there is not any evidence of a change in students' attitudes, there is strong evidence that adding instruction on computational tools positively changes students' beliefs that they can use computational tools.

## CONCLUSIONS

The integration of computational modules across a Materials Science and Engineering curriculum provides a compelling avenue for increasing students' learning of core concepts and improving their comfort with using those computational tools. The use of a Community of Practice in particular facilitates communication about how to implement this new curriculum and support students' learning. This model is sustainable as new instructors buy into teaching courses using the tools and resources that other instructors have developed.

## ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DUE 1347722 and the CAREER Award to A.L.F. (Grant No. DMR-1350008). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

1. A. J. Magana, L. M. Falk, J. M. Reese, *ACM Trans. on Comp. Ed.* **13**, 16:11-16:22 (2013).
2. U. Shaikh, A. Magana, C. Vieira, E. Garcia, *ASEE Annual Conf. & Exp., 2015.*
3. K. Thornton, M. Asta, *Model. Simul. Mater. Sc.* **13**, R53 (2005).
4. K. Thornton, S. Nola, R. E. Garcia, M. Asta, G. Olson, *JOM - J. Min. Met. Mat.* **61**, 12-17 (2009).
5. National Research Council of the National Academies, *Report of a workshop on the pedagogical aspects of computational thinking*
6. C. E. Vergara *et al.*, *Comp. in Ed. J.* **18**, 84-98 (2009).
7. http://www.whitehouse.gov/mgi
8. W. Brostow, V. M. Castaño, *J. Mater. Ed.* **21**, 297-304 (1999).
9. W. Brostow, R. Simões, *J. Mater. Ed.* **27**, 19-28 (2005).
10. D. Yang *et al.*, *Int. J. Eng. Ed.* **28**, 686 (2012).
11. G. L. Herman, L. Hahn, M. West, *2015 Int. Mech. Eng. Congress & Exp.* (2015).
12. M. West, G. L. Herman, *ASEE Annual Conf. & Exp.*, 2015.
13. M. West, M. S. Sohn, G. L. Herman, *ASEE Annual Conf. & Exp.*, 2015.
14. C. Henderson, M. Dancy, *Phys. Rev. Phys. Ed. Res.* **3**, (2007).
15. C. Henderson, A. Beach, N. Finkelstein, *J. Res. Sci. Teach.* **48**, 952-984 (2011).
16. C. Henderson, M. Dancy, M. Niewiadomska-Bugaj, *Phys. Rev. Phys. Ed. Res.* **8**, 020104 (2012).
17. M. Borrego, C. Henderson, *J. Eng. Ed.* **103**, 220-252 (2014).
18. G. L. Herman et al., *ASEE Annual Conf. & Exp.*, 2015.
19. E. Wenger, *Communities of Practice: Learning, Meaning, and Identity*, Cambridge University Press, Cambridge, 1998.
20. E. Wenger, R. McDermott, W. M. Snyder, *Cultivating Communities of Practice,* Harvard Business Press, Cambridge, MA, 2002.
21. E. Mazur, *Peer Instruction: A User's Manual,* Prentice Hall, Upper Saddle River, NJ, 1997.
22. C. H. Crouch, E. Mazur, *Am. J. Phys.* **69**, 970-977 (2001).
23. P. Heller, M. Hollanbaugh, *Am. J. Phys.* **60**, 637-644 (1992).
24. R. A. Layton, M. L. Loughry, M. W. Ohland, G. D. Ricco, *Adv. Eng. Ed.* **2**, 1-28 (2010).
25. M. W. Ohland, *et al. Acad. of Mgmt. Learn. & Ed.* **11**, 609-630 (2012).
26. P. Giannozzi *et al.*, *J. Phys. Condens. Matter* **21**, 395502 (2009).
27. S. Plimpton, *J. Comp. Phys.* **117**, 1-19 (1995).
28. H. J. C. Berendsen, D. V. D. Spoel, R. V.

Drunen, *Comp. Phys. Comm.* **91**, 43-56 (1995).

29. A. C. E. Reid, R. C. Lua, R. E. Garcia, V. R. Coffman, S. A. Langer, *Int. J. Mat. Prod. Tech.* **35**, 361-373 (2009).
30. J. O. Andersson, T. Helander, L. Höglund, P. F. Shi, B. Sundman, *CALPHAD* **26**, 273-312 (2002).
31. G. Klimeck, M. McLennan, S. P. Brophy, G. B. Adams, M. S. Lundstrom, *Comp. in Sci. Eng.* **10**, 17-23 (2008).
32. L. Crocker, J. Algina, *Introduction to classical and modern test theory,* Holt, Rinehart and Winston, Orlando, FL, 1986.
33. T. M. Bechger, G. Maris, H. Verstralen, A. A. Beguin, *App. Psych. Meas.* **27**, 319 (2003).
34. F. M. Lord, M. R. Novick, *Statistical theories of mental test scores,* Addison-Wesley, Reading, MA, 1968.
35. J. Pellegrino, N. Chedowsky, R. Glaser, *Knowing what students know: The science and design of educational assessment,* National Academy Press, Washington, DC, 2001.
36. J. Pellegrino, L. V. Dibello, S. Brophy, *The Science and Design of Assessment in Engineering Education,* Chapter 29 in *Cambridge Handbook of Engineering Education Research*, ed. A. Johri, B. M. Olds, Cambridge University Press, Cambridge, 2014, pp. 571-600.
37. N. Jorion *et al.*, *J. Eng. Ed.* **104**, 454-496 (2015).