Affordances and Challenges of Computational Tools for Supporting Modeling and Simulation Practices

ALEJANDRA J. MAGANA,[1] MICHAEL L. FALK,[2] CAMILO VIEIRA,[3] MICHAEL J. REESE JR.,[4] OLUWATOSIN ALABI,[3] SYLVAIN PATINET[2]

[1]Computer and Information Technology and Engineering Education, Purdue University, West Lafayette, Indiana

[2]Materials Science and Engineering, Johns Hopkins University, Baltimore, Maryland

[3]Computer and Information Technology, Purdue University, West Lafayette, Indiana

[4]Center for Educational Resources, Johns Hopkins University, Baltimore, Maryland

Received 24 October 2016; accepted 15 January 2017

ABSTRACT: This mixed-methods sequential explanatory design investigates disciplinary learning gains when engaging in modeling and simulation processes following a programming or a configuring approach. It also investigates the affordances and challenges that students encountered when engaged in these two approaches to modeling and simulation. © 2017 Wiley Periodicals, Inc. Comput Appl Eng Educ; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.21804

Keywords: computational tools; modeling and simulation; mixed-methods; domainspecific software

INTRODUCTION

In the context of problem-solving in science and engineering, the manipulation and creation of computing artifacts is an increasingly crucial step for understanding and designing systems [1]. Facility with computational modeling, simulation, and domain-specific software has become a new form of literacy in science and engineering domains. Thus, undergraduate students need to effectively combine modeling and simulation skills with engineering science knowledge for effectively designing solutions [2]. Specifically in engineering domains, to effectively use conceptual knowledge to solve problems, individuals need to be able to abstract physical phenomena into some form of representational model (e.g., mental, mathematical, computational) and be able to connect them and effectively adapt them during problem solving episodes [3–5].

Correspondence to A. J. Magana (admagana@purdue.edu).

However, science and engineering education may approach modeling and simulation fundamentally differently. While science education, specifically at the K-12 level, may approach modeling and simulation using pedagogical methods and instructional simulation tools, engineering education may approach modeling and simulation in an expert-like approach using practitioners' modeling and simulation tools [6]. Effective modeling and simulation in engineering also requires (1) knowing when, why and how computational tools and methods work, and (2) applying or configuring existing tools or methodologies to successfully solve problems or design solutions [7]. Thus, we aimed to identify how engineering students experience the use of different affordances of computational tools when they engage in modeling and simulation practices, and how that impacts their conceptual learning.

An initial step toward the realization of the effective use of technology integration for learning purposes is the identification of their benefits and challenges; that is their affordances. Gibson [8,9], described "affordance" as the functional properties that define how such things could potentially be used or manipulated. Therefore, identifying the affordances of different computational tools for engaging students in modeling and simulation practices is a critical process for their proper integration as learning tools. The implications for this study relate to the investigation of the effects of different computational tools on helping students to (1) externalize their thinking by visualizing, testing, and refining the components of their engineering designs, (2) identify and internalize the governing fundamental physical principles or behaviors of phenomena, and (3) develop creative ways in which systems can be created or altered to design new devices, materials and other artifacts.

The guiding research questions are: *QUAN: What are student disciplinary learning gains when engaging in modeling and simulation processes following a programming or a configuring approach? QUAL: How undergraduate engineering students experience modeling and simulation processes when following a programming or a configuring approach? MIXED: What are affordances and challenges that students encounter when engaging in modeling and simulation processes following a programming or a configuring approach?*

To this end, we explore two different computational approaches for supporting modeling and simulation practices: A programming approach and a configuring approach. The major difference between these practices resides in the level of transparency of the underlying mechanism and the way learners interacted with the computational tool. To start with, we explore the literature on modeling and simulation in science education. We describe how these practices take place in the field of engineering and provide an overview of educational research studies that have focused on understanding the distinct way modeling and simulation arise in engineering education. We then proceed with the methods and results, and discuss our findings identifying the implications for teaching and learning.

MODELING AND SIMULATION IN EDUCATION

This section briefly describes an overview of the role of modeling and simulation in mathematics, science, and engineering education. We believe that there is an important distinction between the context and goals of educational research in modeling and simulation in mathematics and science education vis-a-vis engineering education. Without loss of generality, while most educational research in modeling and simulation in mathematics and science education centers on practices and tools that have been specifically designed for learning purposes. In contrast in engineering education, these practices and tools are imported from professional and research contexts and adapted for educational deployment. As a result instructor practices, student perceptions and the relevant educational questions are not quite commensurate in these closely related domains.

Modeling and Simulation in Mathematics and Science Education

Computer modeling [10] and computer simulation [11] have been widely studied in science education contexts, and their benefits in student learning and engagement have been thoroughly documented. One of the primary uses of computer modeling and simulation in science education has been to perform scientific experimentation so individuals connect observed phenomena with their underlying causal processes [12]. In this context, science educators have made a broad distinction between “building” simulations and “using” simulations [12,13]. Learners “build” simulations when they interact with the simulation and also build a model and program it through the user interface [14]. When learners build simulations they are able to modify the attributes of variables, change the agents that are part of the system, design different subsystems and design different functionalities of that subsystem [13]. Learners “use” simulations to explore and develop an understanding of the underlying models [13]. Model exploration is conducted when learners test an input–output relationship [15].

Although this differentiation between building and using simulations has been broadly defined, learning with simulations can actually support different forms of inquiry learning. For instance, building or using simulations can support exploration and observation of natural and man-made systems, testing of theories to explain or validate design decisions, supporting problem-solving tasks, building devices based on desired specifications, calibrating instruments, collecting experimental data, and constructing mathematical models [16].

Modeling and Simulation in Engineering and Engineering Education

In engineering domains, modeling and simulation refer to a combination of processes in which a system’s behavior is predicted by a reductive computational representation. Modeling consists of producing a model to represent the inner workings of a system. Simulation refers to the operation of a model that can be reconfigured and explored [17]. In the engineering context, these two processes are often undertaken so as to be combined in a series of steps that allow a system under study to be altered for a specific purpose. These series of steps, as shown in Figure 1, can be summarized as “model development, experiment design, output analysis, conclusion formulation, and making decisions to alter the system under study” [17]. According to the figure, modeling and simulation are often deployed in an iterative cycle, where conclusions derived from each simulation experiment can feed back into the system under study until it results in the desired altered system.

Research studies investigating the use of computer simulations for learning concepts in engineering have identified the

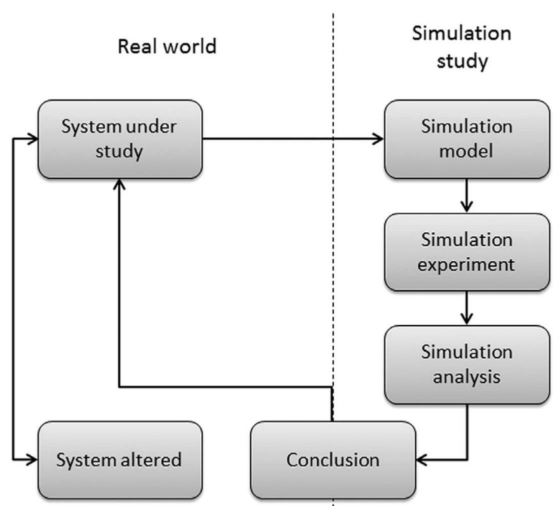


Figure 1 Modeling and simulation processes (adapted from Ref. [17]).

value of these tools for conceptual understanding [18–21], representational fluency [22,23], laboratory experiments and inquiry learning [24–26], and problem solving and design [4,27,28]. In addition, engineering educators have started to identify the breadth and depth of modeling and simulation skills needed by current STEM workforce [29]. Specifically, a study conducted with diverse industry engineering sectors revealed that employers consider of high importance students' abilities, "to understand engineering principles and computational principles that allow them to use computational tools to solve engineering problems by moving between physical systems and abstractions in software" [30]. Responding to industry shifts, engineering professors who use computational simulations for research have started to integrate those as learning tools. For instance, Magana and colleagues [6] conducted open-ended interviews with 14 instructors teaching undergraduate and graduate courses in science and engineering who integrated expert-grade computational tools in their undergraduate and graduate courses. Their phenomenographic analysis revealed an outcome space consisting of eight qualitatively different categories that detailed ways in which instructors conceptualized the incorporation of simulation tools as learning activities into courses they were then teaching. These eight categories are summarized into two major learning goals: (1) using simulations to identify and describe the governing fundamental physical principles or behaviors of devices, materials and other artifacts, and (2) building simulations to apply modeling and computational techniques to approach engineering design tasks. The eight categories identified in this previous study were also aligned with several of the learning objectives for engineering instructional laboratories presented by Feisel and Rosa [31].

Educational researchers have also started to identify the benefits and challenges of expert computational tools into undergraduate classrooms [32,33]. For example, a study in the materials engineering domain identified that students who were exposed to a research-grade molecular dynamics simulation enhanced their abilities to predict how an unfamiliar material would behave at the molecular level, suggesting they understood the atomic processes governing the plastic deformation of materials [34]. In another study, Magana and colleagues also conducted studies that investigated student perceptions and experiences with computation and computational simulations [35]. The quantitative and qualitative analyses showed that graduate and undergraduate students reported overall positive experiences with computational simulation tools and their use. Differences were observed in the way undergraduate students reacted to the computational simulations as compared with graduate students; undergraduate students showed a moderately positive attitude toward their ability to interpret the outputs of the simulation tools. To gain deeper insight into student experiences, the same authors conducted qualitative research studies aiming to identify the potential aspects that may inhibit student learning processes with computation and computational simulation tools [36]. A major theme that emerged from these studies was related to students' need to have more access to the underlying mechanism. This access included aspects related to the underlying equations, the assumptions of the model, the numerical method, and so forth. A second theme related to students' struggles when they were given access to the underlying model. Students, when given access to the underlying model, experienced severe difficulties in conducting mappings between the physical model, the mathematical representation and the computational representations [35,36]. This dichotomy was referred to as the transparency paradox [37].

The studies described above highlight the benefits for using expert computational tools to support researchers in their discovery and innovation processes. Other studies conducted in educational settings also highlight the benefits of pedagogical tools in supporting students' development of conceptual learning. However, only a handful of studies have evaluated the effectiveness of expert computational tools for educational purposes and they have reported mixed results. We therefore considered it valuable to conduct in-depth investigation into the effect of different levels of transparency of computational tools and into the affordances and challenges for supporting modeling and simulation practices.

LEARNING DESIGN, MATERIALS, AND CONTEXT

Our own preliminary work has explored the effect of different levels of transparency, and the way learners interacted with a certain computational tool, on students' conceptual learning (authors, 2016). Our previous work identified that when students interacted with computational tools by just modifying the input parameters, the average gains (i.e., the average increment from pretest to posttest) in their conceptual learning were of 14.11% with a standard deviation of 27.96%. On the other hand, when students interacted with the computational tools by having access to the underlying model and actually having the capability to alter it, the average gain was 8.8% with a standard deviation of 23.61% (authors, 2016). These findings prompted us to investigate further these differences and the effect of different computational tools in affording or hindering the integration of modeling and simulation practices. In this section, we describe the instructional tools that afforded modeling and simulation practices within specific disciplinary contexts. Specifically, we explored affordances and challenges students encountered when using MATLAB in a programming approach, and affordances and challenges students encountered when using COMSOL in a configuring approach. We provide details of the computational tools and the way students interacted with them in the following paragraphs.

The study is contextualized within a curricular innovation aimed at introducing modeling and simulation practices and the use of computational tools across the core courses of an undergraduate materials science and engineering program (authors, 2013). The curricular innovation consists of a new discipline-based computing course entitled "Computation and Programming for Materials Scientists and Engineers" (CPMSE), coupled with the integration of computational learning modules (i.e., one-week long modules) in the major's six core courses offered in the following academic year (three courses per semester). The modules were designed to reinforce computational materials science and engineering skills and to support the acquisition of foundational disciplinary conceptual understanding. For each of the six core courses students were exposed from one to three different computational modules depending on the course. Twelve modules in total were introduced throughout the six courses. This study describes students' experiences with four learning modules implemented in four different courses: "Structure of Materials" and "Physical Chemistry of Materials I: Thermodynamics" delivered in the Fall of 2012 and "Physical Chemistry of Materials II: Kinetics and Phase Transformations" and "Mechanical Properties" delivered in the Spring of 2013 (see Table 1 for details of the modules).

Table 1 Description Learning Modules and Objectives for Each of the Four Areas of Application

Module	Learning goals	Description
Structures: Monte Carlo simulation of alloys and compounds (MATLAB®—required programming)		
	Disciplinary goal	Identify the role of order parameters in relation to phase transitions
	Modeling and simulation goal	Apply basic Monte Carlo algorithm, calculate order parameter in simple simulated system, compare to theoretical prediction
Thermodynamics: Statistical mechanics of magnetic response (Ising model) (MATLAB®—required programming)		
	Disciplinary goal	Understanding the Ising model and how statistical mechanics can be used to perform thermodynamic averages
	Modeling and simulation goal	Apply basic Monte Carlo algorithm, calculate order parameter in simple simulated system, compare to theoretical prediction
Kinetics and phase transformations: Diffusion in 1D and 3D (COMSOL®—no programming required)		
	Disciplinary goal	Understand mass transport in 1D and 3D, effects of boundary conditions
	Modeling and simulation goal	Parameterize and set boundary conditions for a finite element model, analyze, and interpret results, be aware of numerical error and the potential for inaccuracies in numerical results
Mechanical properties: Necking in bars (COMSOL®—no programming required)		
	Disciplinary goal	Observe the difference between elastic and plastic deformation, understand conditions for formation of a neck, analyze stress and strain in 3D
	Modeling and simulation goal	Parameterize and set boundary conditions for a finite element model, analyze, and interpret results

Learning Materials and Computational Tools

Learning modules were designed to: (1) give students extended exposure to computational tools that can help them solve disciplinary problems, (2) provide them with modeling and simulation skills, and (3) reinforce conceptual understanding of disciplinary foundational concepts. Table 1 provides details of how disciplinary learning goals from the materials science and engineering discipline were coupled with modeling and simulation goals for each of the four areas of application. Two domain specific software tools, MATLAB® and COMSOL®, were used as part of this study. MATLAB® is a high-level language for numerical computation using built-in mathematical functions, a data analysis and visualization tool to create models, and a programming and algorithm development tool to develop applications [38]. COMSOL Multiphysics®, herein called COMSOL®, is a general-purpose software platform for modeling and simulating physics-based problems. COMSOL® is based on advanced numerical methods for the analysis and design of coupled or multiphysics phenomena in electrical, mechanical, fluid flow, and chemical applications [39]. These two tools are similar insofar as they provide learners with analytic and visualization tools that integrate numerical computation.

There are two important ways in which these tools are distinct from the student learning perspective. MATLAB® is a general purpose computing environment well suited for numerical analysis. Consequently while MATLAB® is of potential utility in an almost unlimited range of application areas, for most applications (and all those instantiated in MATLAB® here) algorithms must be coded in the MATLAB® language and deployed. In contrast, COMSOL® is a special purpose tool for undertaking the solution of coupled ordinary and partial differential equations. These can be time-dependent or time-independent and may be posed as initial value and/or boundary value problems. For most applications (and all those instantiated in COMSOL® here) the finite element analysis algorithms for solving these equations are already instantiated in the software and the inner workings are opaque to the end user. The other way these two environments differ is in the way the user interacts with the tool, that is, the user interface provided to the user (see Fig. 2). Although MATLAB® can be used to produce graphical user interfaces, in this study MATLAB® is operated through a high-level programming language that includes mathematical functions that provide execution of vector and matrix calculations as well as graphical representation of numerical data. COMSOL® is operated through a graphical user interface (GUI), a series of menus and forms through which the user configures the model and visually represents and analyzes the results by manipulating input parameters.

The process of designing and validating the learning modules consisted of having a disciplinary and computational expert, Author 2, develop modules in consultation with each of the instructors of the four courses. During the first one-on-one meeting between Author 2 and the professor of the course, they jointly identified a topic which could serve as a good candidate for teaching with a computational module. Author 2 also communicated with the course instructors to create an implementation schedule. Author 2 then independently created a first version of the learning module and an accompanying four question multiple choice conceptual pre/post assessment that was shared with the course instructor for further feedback. Details of the validity of the assessment are discussed in the methods section below.

The basic structure of the modules differed significantly between the MATLAB®-based modules and the COMSOL-based modules, particularly in how they required students to engage and build their models. Building on previous work on "building" simulations and "using" simulations [12,13], we made a similar

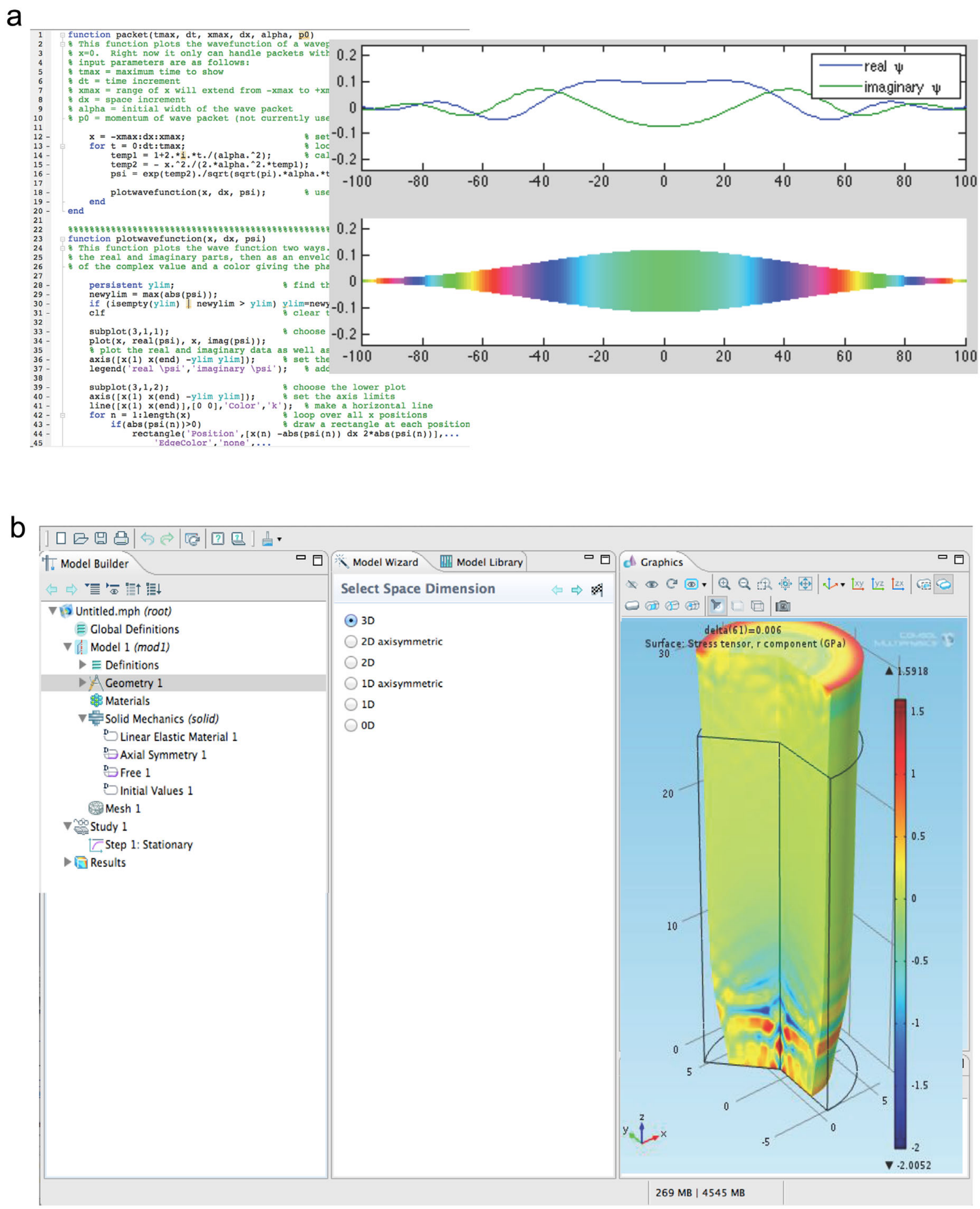


Figure 2 Interfaces for MATLAB® and COMSOL®. (a) MATLAB command prompt and sample of output, (b) COMSOL menus, inputs and sample of output.

distinction for this work. This distinction is highlighted by referring to (1) learning experiences with programming and executing models in MATLAB®, and (2) learning experiences configuring and running simulations with COMSOL®.

Both MATLAB®-based modules involved providing students with the main elements of a Monte Carlo (MC) simulation, an algorithm used to calculate thermodynamic averages for equilibrium systems. Students then had to read the code, understand it in the context of the algorithm for MC and alter the code to correctly implement the MC algorithm. Once a correct code was generated students used the code to investigate the system in question. Here the systems in question were a simple two-dimensional lattice model of a binary alloy in Structures and a similar model of a magnetic material known as the Ising model in Thermodynamics. An MC algorithm in these contexts is a series of steps in which random transformations are proposed for a system. The energetic change to the system that would be caused by each transformation is computed and the transformation is accepted or rejected depending on the energy change and the temperature of interest. Running averages are collected over many such transformations to produce "expectation values" for measurable quantities thereby computing approximations of their thermodynamic equilibrium values. These simulations can be carried out under multiple conditions at different temperatures, concentrations, applied fields, etc., depending on the system.

In the COMSOL® modules, students were provided with a description of a set of steps to carry out in order to build a representation of a series of boundary value problems. In the case of mechanical properties, these involved rate-independent solution of a plasticity problem. In the case of kinetics, these involved time-dependent diffusion and reaction-diffusion equations describing a phase field model. Instructions guided students through selecting the appropriate physics to implement, creating the geometry, setting boundary conditions, and specifying the constitutive equations for the model. COMSOL® was then able to solve the model and students again used the COMSOL® interface to specify plots and other graphical representations of the solution for analysis. In this sense, the COMSOL® exercises represent more of a "cookbook" approach in which students were not privy to the internal workings of the solution procedure unlike the MATLAB® exercises where students were explicitly required to read, analyze, and alter the code to produce a result.

Classroom Context and Delivery

All modules were introduced in a similar fashion implemented in a two-class session format during the course of one week. The day the module was introduced, class started with a brief introduction of the guest instructors (Author 2 and 4) followed by an introduction to the computational module (Author 6 undertook classroom observations). The computational module was briefly introduced in a four-step procedure that typically took no more than 10 min: (1) the guest instructor and the course instructor introduced the context of the module to the class, (2) the instructors explained how the module related to course content and to specific concepts previously taught, (3) the instructors briefly explained the underlying mathematical and computational representations. Students were then provided with the computational module document that introduced in detail the computation or algorithmic aspects of the module in MATLAB®, or how to operate the graphical user interface of COMSOL®. The rest of class time was used for students to work on the modules. Instructors helped students individually as they worked on the modules during this class, clarifying confusing aspects as needed. Author 2 and 4 held office hours during that week so students could receive additional, personalized assistance during the assignment. During the second class of the same week, students continued working on module solutions during class time. In this second class period, students asked questions to the instructors discussed their questions and problem-solving approaches with their peers. Teaching assistants from each class also responded to questions from students. These typically involved clarifying the instructions, answering questions about the structure of the code (MATLAB®), providing guidance regarding interaction with the user interface (COMSOL®), verifying that steps were properly configured and executed, and discussing the computational results in light of the engineering problem being addressed. Each assignment was typically due 1 week after the class in which it was introduced. Students could work on all exercises outside of class using on-campus computer laboratories or using their personal computers. When students opted to use their personal computers, they could install the software via a campus-wide educational licensing agreement or run the software via remote desktop software.

Besides the different software employed for the modules, the only difference between each implementation was the level of engagement of the course instructor, which we believe was due to their varying level of comfort with computation. While some instructors participated more actively during the delivery of the modules, others chose to only participate at the very beginning introducing the guest instructors, the module and the relationship between the module and the course topics. However, and as mentioned earlier, all professors were fully engaged in the conceptual design of the modules and pretest and posttest assessments.

METHODS

A mixed-methods sequential explanatory design [40] was employed to plan, analyze, and report the findings of this investigation. Accordingly, the design started with collecting and analyzing quantitative and then qualitative data in two consecutive phases. The quantitative phase identified the effect of integrating modeling and simulation practices to increase students' understanding of the subject matter. Then, qualitative data was collected and analyzed from a structured interview to help explain, differences in students' learning gains (i.e., to explain the quantitative results from the first phase) [41]. Priority was given to the qualitative data based on the purpose of the study and our research questions. Specifically, in the second phase our goal was: (1) to characterize how students engaged with different computational tools, (2) to identify benefits and challenges that students encountered when engaging in modeling and simulation processes, and (3) to identify how such affordances and challenges may have contributed to differences in conceptual learning gains when students engaged in modeling and simulation processes following a programming or a configuring approach. Figure 3 provides an overview of our research design.

The quantitative component of our research aimed to identify: What are student disciplinary learning gains when engaging in modeling and simulation processes following a programming or a configuring approach? To measure learning gains we conducted pretest and posttest assessments aimed to measure student conceptual understanding. The pretest assessment measured what students learned from the lecture and the posttest assessment measured what students learned by solving the computational challenge. The conceptual assessments were directly aligned with the learning objectives of the lesson (see Table 4 for details). The qualitative portion of our study aimed to identify: How undergraduate engineering students experience modeling and simulation processes when following a programming or a configuring approach? To perform the qualitative analysis, we implemented a case study approach. Combining case study with mixed-methods was originally proposed by Ivankova, Creswell, and Stick (2006). We chose the same approach because

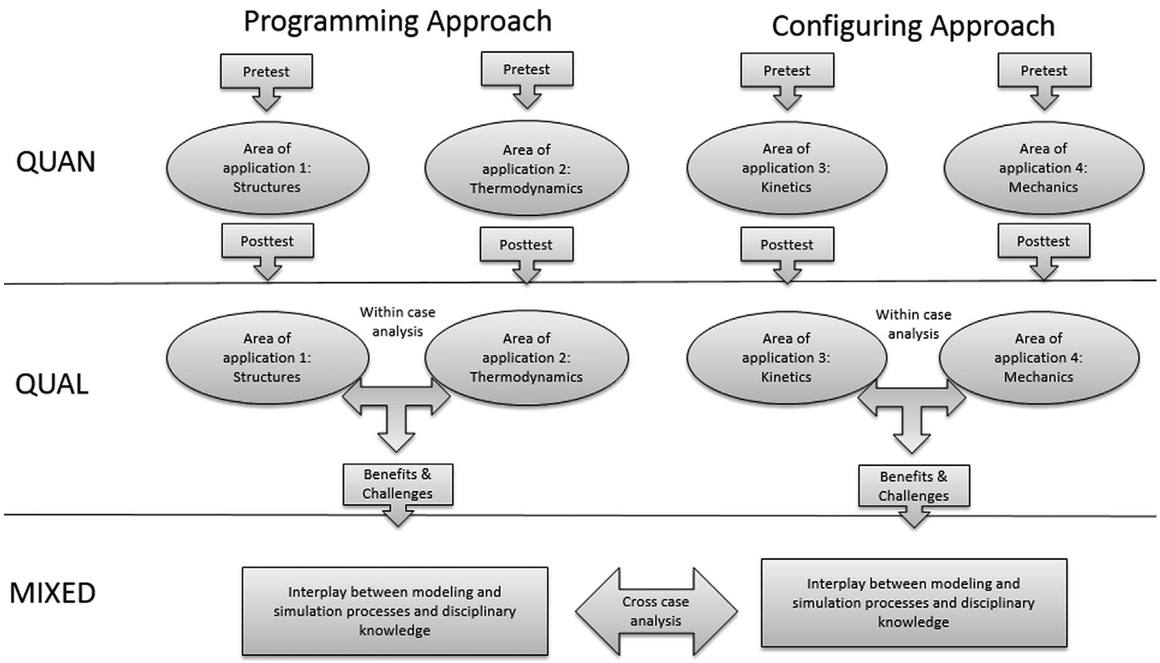


Figure 3 Alignment between the research questions and data analysis procedures.

doing so would help us explain how students' ways of interacting with the computational tools, tested in the first phase, resulted in significant or not significant learning gains. For this, we identified a subset of six students from each course. To capture as much variation as possible from students' experiences, a combination of high and low performers, according to their grade on the module, were invited to participate in the think aloud. We then performed within case analysis for each of the cases where we aimed at identifying how students engaged in different stages of the modeling and simulation process while solving the computational challenge. Finally, the integration of the quantitative and qualitative portions of our study were guided by the research question: What are affordances and challenges that students encounter when engaging in modeling and simulation processes following a programming or a configuring approach? This integration was performed in the cross case analysis of the two case studies. This last portion of the analysis is reported along with the discussion section.

Participants and Procedures

The participants for this study were 82 undergraduate students from four different Materials Science and Engineering courses at a Mid-Atlantic University. The courses were Structure of Materials ($N = 23$) and Physical Chemistry of Materials I: Thermodynamics ($N = 27$) offered in the Fall of 2012; and Physical Chemistry of Materials II: Kinetics and Phase Transitions ($N = 16$) and Mechanical Properties of Materials ($N = 16$) offered in the Spring of 2013. From the 82 students, 71 completed the procedures for at least one of the courses. Because of the naturalistic nature of this study, there was some overlap between participants and courses. Four students were simultaneously enrolled in all four courses; one student participated in the Thermodynamics, Structures of Materials and Kinetics courses; eight students were enrolled in the Kinetics and Thermodynamics courses; six students were enrolled in the Structure of Materials and Mechanical Properties courses; and three students were enrolled in the Thermodynamics and Structures of Materials courses. Table 2 describes the number of participants enrolled in each of the courses. Some of the students who were absent the day of data collection, only presented the pretest or the posttest, therefore both values are provided.

The procedures for data collection consisted of having students engage with the computational modules at different times in the semester. Before implementing each of the computational modules a pretest assessment was applied to identify students' current understanding of disciplinary concepts in the selected area of application. Students then worked on the computational modules during class time and for a one week-long period. During that week, students were able to seek assistance from the professor or the teaching assistant during specific office hours. Students completed the posttest assessment during the class in which they submitted the module.

Results from the quantitative strand were used to identify the cases for the qualitative strand. Thus, a purposeful subsample of six students from each of the four courses was drawn to conduct in-depth structured interviews to characterize students' processes for solving the projects and to identify possible benefits and challenges of the computational tools following a programming or a configuring approach. The three highest and three lowest performing students, based on the module grade, were then asked to participate in a structured interview at the end of the corresponding semester. Since the module grades focused on student ability to use

Table 2 Number of Participants Per Course

	Number of participants		
Area of application	Pre and post test	Pretest	Posttest
Structure of materials	19	23	19
Thermodynamics	23	27	23
Kinetics	16	16	16
Mechanical properties	13	16	13

expert computational tools for modeling and simulation, we argue that these scores are comparable across the different courses. This purposeful sampling was done as one way to characterize how students engaged in meaning making based on the computational tools the used. The goal was also to explore the affordances and challenges students encountered when engaged in modeling and simulation processes following a programming or a configuring approach. Students were not told that they were recruited based on their module grade. Whenever students declined to participate because of scheduling conflicts, the next student in the ranking was recruited into the study (this happened twice). The subsample of students who participated in the structured interview, their pseudonym, the semester they participated, and the area of application they were exposed to, are described in Table 3.

All participating students in the Spring semester interviews, but (L)River_M, were exposed at least to one computational module the previous semester. Five of these students participated in the structured interviews in both semesters. These participants are identified with an asterisk on Table 3 and the pseudonyms were kept the same. For instance, (L)Justice_S, who participated on the interview for Structures of Materials during Fall 2012, is the same (L)Justice_K, and participated on the interview for Kinetics during Spring 2013. That is, students who were exposed to computational modules in the Fall and Spring semesters were exposed to the same practices, but within different disciplinary contexts, tools, and even computational approaches. Interviews were designed to last about one hour, but students were given unlimited time. These interviews lasted between 40 and 80 min.

Data Collection Method

The data collection methods consisted of multiple-choice conceptual assessments (quantitative measures and an interview protocol (qualitative measures).

Conceptual Assessments. In order to obtain a quantitative measure of student learning gains in their conceptual understanding over the course of the computational modules, a unique four-question multiple choice assessment was designed for each module and were scored either correct or incorrect (1 or 0). Each question on the assessment was tied to a learning objective common to the class and the computational module (see Table 4).

The questions were chosen to test common misunderstandings that students have related to the foundational disciplinary learning goals in these four classes. In order to describe how these learning objectives were transformed into test questions, consider specific objective 2 within the subject area of Thermodynamics. Because it is typically impossible to enumerate all states in most systems of practical interest, Monte Carlo methods compute an approximation of the equilibrium average by visiting states in the system with the same probability that they would occur in equilibrium. (This is known as "importance sampling"). Though not all states are visited, the computed value will be a good approximation of the equilibrium expectation value because the states sampled are representative of the equilibrium system. The question posed to the students was to properly complete the sentence: "When computing an expectation value it is most important that____." Several confounding answers were presented: "every state is included in the calculation"; "the lowest energy states are included in the calculation"; "the states with the highest multiplicity are included in the calculation," and "a representative sample of states was included in the calculation." Students needed to find the correct answer, which was that "a representative sample of states was included in the calculation."

Interview Protocol. Four interview protocols were designed by two computational materials science and engineering experts, Author 2 and 4. The interview protocols were purposefully designed to be closely related based on their computational approach (i.e., programming or configuring), and each focused on concepts and skills already introduced as part of the course disciplinary modules. Appendix B describes how the interview protocols were structured and provides sample probing questions. The computational modules on the application of structures and thermodynamics focused on inferring macroscopic response from microscopic behavior. To this end, statistical mechanics was applied to discrete systems.

The interview protocols following a programming approach for both the structure of materials and thermodynamics area of application were focused on the consequences of interactions between magnetic domains on a nanotube structure at different temperatures. The only difference between these two interview protocols was that Structures' students were asked questions related to the "order parameter," the concept introduced in the Structures'

Table 3 Participant Pseudonyms Per Semester and Area of Application

Area of application	Experiences programming models with MATLAB® software (Fall 2012)	Area of application	Experiences configuring simulations with GUI-based COMSOL® software (Spring 2013)
Structure of materials	(H)Dakota_S	Kinetics	(H)Charlie_K*
	(H)Sidney_S		(H)Phoenix_K*
	(H)Jaylin_S		**(H)Peyton_K**
	(L)Justice_S		(L)Justice_K*
	(L)Jessie_S		**(L)Heyden_K**
	(L)Landry_S		**(L)Tatum_K**
Thermodynamics	(H)Charlie_T	Mechanical properties	(H)Sidney_M*
	(H)Skyler_T		(H)Emerson_M*
	(H)Casey_T		**(H)Rowan_M**
	(L)Emory_T		**(L)Finley_M**
	(L)Phoenix_T		(L)River_M
	(L)Emerson_T		**(L)Parker_M**

Bold values represent the students from the Spring semester who were exposed to computational modules the previous semester.
*Students who participated in the interviews in both semesters.

Table 4 Learning Modules and Assessments of Conceptual Understanding

Learning modules	Assessment of conceptual understanding
Structures: Monte Carlo simulation of alloys and compounds (MATLAB®—programming)	
1	In a system with a positive enthalpy of mixing phase separation will occur below a critical temperature. This results in coexistence of two solid solutions of different compositions that become more pure as the temperature decreases. A prediction of these compositions can be obtained using a regular solution model
2	In a system that forms an ordered compound the extent of ordering can be described by the long-range order parameter
3	In some systems with a negative enthalpy of mixing ordered compounds emerge below a critical temperature. This results in increased ordering of atoms on distinct sublattices with the degree of order increasing as temperature decreases. A prediction of this increase in ordering can be obtained using a Bragg–Williams model
4	The nature of the bonding influences whether a material will exhibit phase separation or ordering; a propensity for bonds between like atoms favors phase separation, while a propensity for unlike bonds favors ordered compounds
Thermodynamics: Statistical mechanics of magnetic response (Ising model) (MATLAB®—programming)	
1	Statistical mechanics of the canonical (NVT) system allows us to calculate equilibrium properties by taking an average over all the microstates in the system weighted by the probability that each microstate occurs at that temperature
2	Because it is impossible to enumerate all states in most systems Monte Carlo provides an approximation by visiting states in the system with the same probability that they would occur in equilibrium. Though not all states are visited, the average value will be a good approximation of the equilibrium expectation value if states are representative of the equilibrium system
3	Paramagnets do not have strong interactions between spins and therefore exhibit magnetization that is proportional to the applied field at low field strength. At high field strength the magnetization saturates
4	In a ferromagnet spins align in the same direction, and this causes the system to have a magnetization at zero applied field at low temperature. Above a critical temperature this spontaneous magnetization disappears
5	In an anti-ferromagnet spins align in opposite directions, and this causes the system to have very low magnetization at low temperature even under an applied magnetic field. Above a critical temperature anti-ferromagnets behave like paramagnets
Kinetics and phase transformations: Diffusion in 1D and 3D (COMSOL®—configuring)	
1	A well-posed mass transport problem requires specifying the diffusivity of the species, the initial value of the concentrations of the species and the boundary conditions
2	Mass conservation requires that the time rate of change of a species in the absence of local sources and sinks is given by the difference between the flux into the region and the flux out of the region
3	The geometry of a mass transport problem has an influence on the way one expresses the gradient in the problem. This has an effect on the rate of mass transport
Mechanical properties: Necking in bars (COMSOL®—configuring)	
1	Elasticity problems can be solved using Hooke's law making sure to impose the correct boundary conditions using the known elastic constants, in this case the Young's Modulus and Poisson's ratio
2	Plasticity is, in general, volume conserving and this can be used to estimate contractions during uniform extension or compression
3	Many structural materials exhibit hardening whereby the yield stress increases with plastic strain
4	Under large tensile deformation materials can undergo an instability called necking. Necking arises because the rise in the yield stress due to hardening is insufficient to counteract the increase in stress due to the contraction of the cross-section of the member due to volume conservation

module, while Thermodynamics' students were asked about the "magnetization," the concept introduced in the Thermodynamics' module. Table 5 describes the design similarities of each of the interview protocols highlighting the modeling and simulation goal, the mathematical model represented and the computational technique used to approach a specific disciplinary problem. The interview protocols following a configuring approach on the application of kinetics and mechanical properties focused on relating continuum constitutive behavior to system behavior. For this purpose, partial differential equation (PDE) models were applied to a continuum system with appropriate boundary conditions to model systems under specific conditions. These problems are generically denoted as boundary value problems. The interview protocol for the mechanical properties area of application

Table 5 Context and Design Features of the Interview Protocols

Area of application	Modeling and simulation goal	Disciplinary problem	Mathematical model	Computation method	Tool
Structures	Studying a system at the microscopic level to be able to understand and predict macroscopic behaviors	Calculate expectation value for the order parameter of a magnetic nanotube	Statistical mechanics	Monte Carlo algorithm	MATLAB®
Thermo-dynamics		Calculate expectation value for magnetization of a magnetic nanotube	Statistical mechanics		
Kinetics	Application of continuum theory to be able to predict system behavior	Characterize the diffusion of a drug in a delivery system to determine the drug uptake	Fick's second law	Finite element analysis	COMSOL®
Mechanical properties		Evaluate a cylindrical bar for necking and stability as a tensile member of structural system	Elastic/plastic constitutive relations		

consisted of evaluating a structural system for necking and stability, while the protocol for the kinetics area of application consisted of characterizing the mass transport in a drug delivery system.

Each of the interview questions of the protocol corresponded to some specific steps of the modeling and simulation cycle. Table 6 describes an approximate alignment with the steps associated with modeling and simulation processes as proposed by Maria [17] and how those are aligned with the two interview protocols. As described by Maria [17], not all the steps may be required or may be possible to perform. More steps may be needed as well as multiple iterations between sub-steps. Thus, the steps followed in each of the protocols do not map neatly. Within the programming protocols with MATLAB®, we have placed an emphasis on problem framing; model formulation and implementation along with the corresponding documentation; selection of experimental design; and interpretation and presentation of results. Within the configuring protocols with COMSOL®, additional steps were implemented including: problem identification; selection experimental conditions for runs; executing the simulation runs; and recommending future course of action. These additional steps were possible to be implemented due to the affordances of the COMSOL software.

Data Analysis Methods

Quantitative Data Analysis. Each of the questions for the pre and posttest assessments was scored with either 0 or 1 and the sum of the correct responses was normalized in a score that ranged from 0 to 100%. Descriptive statistics as well as inferential statistics were used to evaluate the effect of the computational modules on disciplinary learning gains. One sample paired t-test was used to compare the pretest scores to the posttest scores. The null hypothesis for each test is that the mean score on the pretest for a group of students is equal to the mean score on the posttest for the same group of students. The alternative hypothesis for each test is that the mean score on the pretest is significantly different to the score on the posttest. Finally, a comparison between the two groups of modules (i.e., programming MATLAB® vs. configuring simulations in COMSOL®) was performed.

Qualitative Data Analysis. As Stick and Ivankova did in their sequential explanatory design [42], we decided to enhance the depth of the qualitative analysis by using a multiple case study design [43]. A case study approach facilitated in-depth investigations of student experiences with modeling and simulation. The use of multiple cases or units of analysis is a common strategy for identifying contextual variations [44]. By comparing cases, "one can establish the range of generality of a finding or explanation, and at the same time, pin down the conditions under which that finding will occur" [45]. We individually studied how the six students in each of the four areas of application of modeling and simulation approached the solution of the engineering challenge [46].

The case study was then structured to optimize the opportunity to learn as much as possible about the experiences of the participants [47]. Two specific cases were built around the four engineering undergraduate courses whose students were exposed to computational learning modules. We chose to group the four courses into two cases because of two reasons (1) the major elements of modeling and simulation practices were very similar for the structures and the thermodynamics course, and for the kinetics and mechanical properties course, and (2) because the quantitative results were similar between the modules following a programming approach and the modules following a configuring approach, suggesting similar levels of difficulty between the two implementations for each approach. The typology, summarized in Table 5, highlights the two cases that were observed as part of this research design. Similar elements include the modeling and simulation goal, the computational method and the modeling tool. Differences were only centered in the area of application.

Each case included observation of 12 participants engaging in discipline-based computational problem solving tasks. In case one, the computational activities are related to students' experiences altering and executing programs using MATLAB® and in case two the activities engaged students in configuring simulations using of a GUI-based simulation tool called COMSOL®. Once the interviews were transcribed by a third party, two researchers performed the data analysis. Thematic analysis [48] at two levels was used to analyze data for identifying challenges and benefits. Within-case analysis was performed between the two areas of application exposed to similar modeling and simulation tasks and cross-case analysis was performed across the two major cases. Figure 3 depicts how data analysis procedures were aligned with the three research questions.

The data analysis started by reading through the interview transcripts several times and conducting open coding [44]. During open coding categories were freely generated [49,50]. To this end, each question was analyzed independently gathering insights from the students' answers and, in some cases, comments by the interviewer. After the students' actions and responses for each question were identified, common ideas/approaches between them were highlighted. A title, a description and some quotes where assigned to these commonalities and they were named as categories. Also, their number of instances was identified. After

Table 6 Alignment Between the Modeling Process and the Interview Protocols

Processes	Steps	Programming models with MATLAB®	Executing GUI-based simulations with COMSOL
Simulation model	Step 1: Identify the problem Step 2: Formulate the problem	The students were presented with a situation where a new magnetic nanotube with four domains was designed. Students should describe which of two instances of this nanotube would be more common at low temperatures. This would lead them to describe their understanding of the situation and how temperature affects magnetization in the context of statistical mechanics	The students were presented with a finite element analysis of diffusion problem or mechanical response problem. They had already been exposed to the problem during the computational module and they were required to apply it in a particular context: a drug delivery device or a structural member. The first part of the interview focused on understanding the problem and the boundary conditions
Simulation development	Step 3: Collect and process real system data Step 4: Formulate and develop a model Step 5: Validate the model Step 6: Document the model for future use	The second section included the definition of a mathematical representation for the energy and the magnetization or order parameter of the system. The mathematical modeling process would provide the students with the tools to measure the macroscopic phenomena	The students were provided with the developed solution during the computational module. The second section was focused on understanding why and under which conditions the previous solution worked well. They used a COMSOL® model to describe their analysis
Simulation experiment	Step 7: Select appropriate experimental design Step 8: Establish experimental conditions for runs Step 9: Perform simulation runs	Students were asked to describe their understanding of the Monte Carlo Algorithm as well as completing its implementation for the given situation. They did not only have to explain it step-by-step but to describe its purpose	Students were asked to build a solution for the new requirements. They started from the previous COMSOL® solution and adapted it to the new conditions
Simulation analysis	Step 10: Interpret and present results Step 11: Recommend further course of action	Students describe how they would evaluate macroscopic phenomena at a higher scale based on the changes of microscopic properties	Students examine the COMSOL® simulation results and discuss the results in the context of Fick's second law for Kinetics and the Considère criterion for mechanical properties

the experts who designed the task conducted their review, the categories were refined and regrouped using axial coding [50]. Axial coding was used to compare and contrast the emerging categories and reorganize them into themes. Graphing techniques were also used, such as 2×2 or other cell design to compare several categories at once. Those with a very limited number of instances were either discarded or assigned to another category.

The next step was to write individual case studies for each of the courses. In doing so, we grouped the quotes that belonged to the same category. We reread each of the quotes and, if necessary, regrouped them. Once the quotes were categorized, we proceeded to identify the most representative quotes for each category and followed by documenting each individual case. Finally, we examined all the cases together to inspect similarities and differences in order to establish consistent patterns across multiple cases.

Role of the Researchers

The research team consists of an interdisciplinary group of researchers who are engineering educators or educational researchers. Three investigators worked together in all aspects of this study by providing their own disciplinary expertise while simultaneously learning from each other. Author 2, with expertise in research and teaching of computational materials science and engineering, developed the computational learning modules, the assessments and the structured interviews. Author 1, with expertise in engineering education research in the area of modeling and simulation in science and engineering contributed to the design of the study and analysis of learning outcomes. Author 6, with expertise in educational assessment and project evaluation, assisted with data collection and analysis. In addition, one postdoctoral student, Author 4, with expertise in computational materials science participated in the study assisting Author 2 with the design and validation of the learning materials and data collection methods. Author 2, Author 4 and the course instructor delivered the computational modules during class time. Author 2 and 4 offered additional office hours to help students with their assignments. Two doctoral students, Author 3 and 5, with training in engineering education and computing education, participated in the data analysis under the supervision of Author 1. All members of the team worked together in the interpretation of the findings, implication of the study and conclusions.

Validity and Trustworthiness

Validity and reliability for the quantitative portion of the study was considered throughout the entire process. During the design stage, data collection instruments, including the conceptual assessments and the interview protocols, were jointly created by two disciplinary experts and revised by a third one. These instruments were then revised by two social scientists who provided feedback on the structure, length, wording, and presentation of the materials. The interview protocols were pilot tested with at least one student and then revised based on the student feedback.

During the data collection stage, we employed multiple sources of evidence to verify our data sources. Sources included the conceptual assessments, think-aloud transcripts, interview videos, and students' hand-written notes [43]. During the qualitative data analysis, we followed the internal validity guidelines for case study analysis as proposed by Eisenhardt [51]. We attempted to discover the underlying theoretical reasons for why a relationship existed. This was done in order to provide a good understanding of the dynamics of such relationships. We also measured constructs and verified relationships by judging the strength and consistency of the relationship within and across cases. We displayed in detail the evidence and procedures when reporting the findings so that readers may apply their own judgment.

The trustworthiness of the qualitative data analysis was supported by two strategies; inter-rater reliability combined with peer-debriefing to revise and refine the initial categories. In the first iteration, the two researchers individually categorized 16% of the data (i.e., two transcripts out of twelve from each case). Then, the two researchers discussed their separate categories and worked collectively to achieve a consensus on categories describing similar experiences. Once an agreement was reached, the transcripts were coded separately using the identified codes. In this process, new categories were identified where different experiences of modeling and simulation were found. Finally, to validate the categories, a third researcher expert in the technical background conducted a separate peer debriefing [52]. This process offered the researchers a different perspective on the data that helped them to refine and sharpen the findings. This analysis allowed the educational researchers to validate the accuracy of the student responses, thus improving the credibility of the findings.

During the data analysis stage, we also sought multiple ways to verify the data collection and analysis by means of coder and dialogic reliability checks with all six research group members. As mentioned above, 16% of the transcripts were initially coded by two educational researchers. For all cases, the group met to discuss the coding schemes until we reached consensus. First, the two educational researchers presented the emerging categories and the sample quotes supporting them. Then, two disciplinary experts revised the categories and quotes to better explain what affordances and challenges students encountered. These explanations were incorporated into the analysis framework as well as into the explanation of the results. We dutifully followed a process for documenting a chain of evidence where we continually checked the links between our research questions, protocols, data, and claims [43].

RESULTS

This section presents the quantitative results of the disciplinary learning and qualitative results of student experiences with modeling and simulation. We first analyze these results separately in depth and then we performed a cross-case comparison relating and contrasting each of them.

Learning Performance

This section aims at responding the research question: What are student disciplinary learning gains when engaging in modeling and simulation processes following a programming or a configuring approach? Two approaches, building models by altering and executing a program created using MATLAB®, or configuring and executing models within COMSOL®, were used in the modules to integrate computation into disciplinary learning. There were 71 observations from all modules, because each participant could encounter multiple modules. Among the observations, 42 were based on modules of the first type that had a programming component in MATLAB® while 29 were based on modules of the second type in which the underlying algorithmic complexity was hidden and students instead relied on constructing models via the

COMSOL® GUI interface. The descriptive statistics of learning performance, in terms of conceptual understanding for each module, is summarized in Table 7.

Pre- and post-test learning scores of each module were compared to identify student learning performance. We reject the null hypothesis for both "configuring models," suggesting that learning gains were statistically significant for the Mechanical Properties module 1, $t = -3.45$, $p = 0.0047$ and the Kinetics module 1, $t = -4.34$, $p = 0.0006$. This indicates that students performed significantly better in understanding associated disciplinary concepts after being exposed to the two computational learning modules that did not require them to simultaneously engage in disciplinary learning, configuring, and programming models. The results of the paired t-test for the programming approach were non-significant and therefore, we fail to reject the null hypothesis.

Affordances and Challenges in Different Stages of the Modeling and Simulation Process

This section aims to respond the research question: How undergraduate engineering students experience modeling and simulation processes when following a programming or a configuring approach? For this, we then qualitatively analyzed student experiences with modeling and simulation to identify (1) how students engaged with different representations, (2) how did students engage in modeling and simulation processes following a programming or a configuring approach, and (3) benefits and challenges that students may have encountered when engaging in modeling and simulation processes. We first describe general categories found when students programmed models with MATLAB®, and then identify patterns of student experiences with configuring and executing models in COMSOL®.

Within-Case Analysis 1: Programming and Executing Models With MATLAB®. Twelve students were interviewed to study their experiences with modeling and simulation using MATLAB®. The participants in this study were given gender-neutral pseudo names per IRB guidelines. Six of the participants were enrolled in the Structures of Materials course: (H)Dakota_S, (L)Justice_S, (H)Jaylin_S, (L)Jessie_S, (L)Landry_S, and (H) Sidney_S. For these students, the protocol was focused on a model for ordering atoms in a lattice to understand order parameter and the Monte Carlo algorithm. The other six students were part of the Thermodynamics course: (H)Charlie_T, (H)Skyler_T, (L) Emory_T, (L)Phoenix_T, (H)Casey_T, and (L)Emerson_T. Their protocol was focused on the concept of magnetization and the use of the Monte Carlo algorithm to understand statistical mechanics. This section describes how students approached the different stages of the modeling and simulation process along with affordances and challenges students encountered.

Simulation Model Process. The participants started by analyzing and describing a situation with different temperatures for the given system. When asked about how magnetization/order parameter should behave at different temperatures, students drew figures representing the order of domains to explain its effects. (H) Charlie_T, (H)Dakota_S, (L)Emory_T, (L)Jessie_S, (L) Landry_S, (L)Phoenix_T, (H)Casey_T, (L)Emerson_T, and (H) Sidney_S accompanied the representations by responses such as the one from (H)Charlie_T as identified on Appendix A as Q1. Note that although both students used graphical representations to support their explanation of the model, these are contrasting examples. (H)Charlie_T provided an explanation that demonstrated a good understanding of the underlying statistical mechanics concept. He recognized systems at low temperatures favor states with low energies while systems at high temperatures explore their states democratically and therefore more often visit states with higher multiplicity and less order. In contrast (L) Emerson_T (as shown on Appendix A as Q2), demonstrated a lack of clarity in conveying this concept. The response did not clearly indicate whether the student believed that high energy states have higher probability than low energy states at high temperature, which would be false, or if he or she understood that high energy states are more common because they constitute a larger number of states, which would be correct. Furthermore, at the end of the activity, students were asked to identify the most difficult concepts in solving the problem. Four Structures students [(H)Dakota_S, (L)Jessie_S, (L)Justice_S, and (L)Landry_S] and two Thermodynamics students [(L)Emerson_T and (L)Emory_T] highlighted that the order parameter/magnetization was the most difficult part.

Simulation Development Process. Most of the participants (11 out of 12) were not able to build either a mathematical or a computational external representation of the phenomenon. Some of them had problems with foundational mathematical representation such as (L)Emory_T who said, "Is there any mathematical thing that specifies adding a number to a consecutive one?" (L) Emory_T, also conflated the expression for magnetization with the energy equation as shown in a representative quote (Q3) on Appendix A. Some other times, students exhibited a lack of conceptual understanding when building the mathematical representation. For example, when asked for the mathematical expression of order parameter, all Structures students struggled to

Table 7 Descriptive Statistics and t-test on Student Learning Scores of Each Module

Module	Approach	Pretest Mean	Pretest Std. dev.	Pretest n	Posttest Mean	Posttest Std. dev.	Posttest n	t	DF	P-value
Structure of materials										
1	Programming	48.68	22.78	19	51.32	32.78	19	−0.38	18	0.7061
Thermodynamics										
2	Programming	35.87	24.80	23	44.57	30.11	23	−1.50	22	0.1479
Mechanical properties										
1	Configuring	25.00	25.00	13	59.62	24.02	13	−3.45	12	0.0047*
Kinetics and phase transformations										
1	Configuring	53.13	20.16	16	75.00	20.41	16	−4.34	15	0.0006*

*Statistically significant at with $P < 0.05$.

recall a definition for this concept. (H)Dakota_S explicitly said, "I somewhat forget what an order parameter is." (L)Landry_S said that the most confusing part of the exercise was, "The piece about order parameter was really the most confusing. Because I didn't know what they wanted, or what they meant by order parameter." And (L)Justice_S: "I think mainly the order parameter one just because I didn't really understand it."

Students proposed an external mathematical representation for the two variables: energy and either magnetization (Thermodynamics) or order parameter (Structures). During this process, all the students built upon previous answers and representations. However, six students made fallacious connections between related but incongruent representations that reinforced conceptual misunderstandings. For instance, when students were asked to write a mathematical expression for the energy of the system in terms of the four domains, (H)Charlie_T, (H)Dakota_S, (L) Emory_T, (L)Jessie_S, (L)Landry_S, (L)Phoenix_T, (H) Casey_T, (L)Emerson_T, and (H)Sidney_S used a graphical representation that depicted the alignment of domains to build the mathematical representation for the energy as shown in the corresponding quote (Q4) on Appendix A.

(L)Landry_S, (H)Sidney_S, and (H)Charlie_T also extracted the mathematical expression for the magnetization or order parameter from the MATLAB® code provided. This inverted the expected sequence of mappings between models anticipated by the exercise, as it was anticipated that students would use the mathematical expression to analyze the algorithm, not vice-versa (see (H)Charlie_T's quote (Q5) on Appendix A for a sample). Finally, when asked about the most difficult part of the assignment, the required math to solve the problem was noted as difficult by three students from the Thermodynamics course and three from the Structures course.

Simulation Experiment Process. During this step, students worked with a MATLAB® implementation of the MC algorithm. Regarding the MATLAB® representation, (L)Phoenix_T viewed the algorithm as an aid to their thinking process: "I read the algorithm . . . to see what—if there's anything here that might help me" while (L)Emerson_T felt it was a "convenient way" to solve these kinds of problems: "So having an algorithm, that's just fool proof as long as you follow the steps the same way each time or retrieve the same results."

However, when the participants were trying to understand and complete the Monte Carlo algorithm, some of them presented misconceptions about the disciplinary knowledge or the algorithm. (H)Casey_T: Right now I'm just confused about what "energysum" is. Or (H)Charlie_T: I don't know why we have the running total actually.

Participants kept saying that the aim of the algorithm was either to converge to the most stable state [(L)Justice_S, (H)Dakota_S, (L)Emory_T, (H)Sidney_S] or to find a minimum [(L)Phoenix_T, (H)Casey_T, (L)Emerson_T, (L)Jessie_S, (H)Skyler_T, (H)Charlie_T]. This is an important distinction between thermodynamics, which asserts that the equilibrium of the system is associated with an extremum of an appropriately chosen thermodynamic potential function and statistical mechanics, which asserts that the equilibrium is associated with a properly weighted average over microstates. These two assertions lead to identical results in an ergodic equilibrium system in the thermodynamic limit, but the distinction is important for appreciating the difference between the two theoretical frameworks. Hence, students struggled to actually understand the actual purpose of the algorithm, and connect it with the disciplinary knowledge related to the module.

Although students like (H)Sidney_S struggled with this representation because of their lack of experience with MATLAB ®: "I'd never done MATLAB® before," others like (L) Emerson_T or Skiler_T highlighted the benefit they got out of programming MATLAB® to solve the exercise (see corresponding samples of quotes Q6 and Q7 on Appendix A).

To the final question related to the most difficult steps in solving the problem, four students from Structures and three students from Thermodynamics noted the implementation of Monte Carlo algorithm as the most challenging one. Three Structures students that talked about the Monte Carlo algorithm said they were confused with some specific parts of the algorithm such as Boltzmann's constant or the quantity "Delta" in the code. (H)Jaylin_S said, "Why exactly E to the negative delta? Or why k^BT? But I remember thinking that out once, by reading the book and Wikipedia, but I just forgot now. That would be something very nice to know," (L)Jessie_S: "I guess just Monte Carlo, and more generally, like, actually understand the math behind it and stuff, instead of just grasp at it."

Simulation Analysis Process. When the participants were trying to understand and complete the MC algorithm, some of them presented misconceptions about magnetization, order parameter, statistical mechanics, or the algorithm. For example, (H)Casey_T was struggling to calculate the magnetization and its relationship to the energy "So I guess I got confused with the energy and the magnetization. . .so I guess I'll just add a quantity to calculate the magnetization." While (H)Casey_T identified part of the confusion as arising from an inability to distinguish between the energy and the order parameter (magnetization), the other likely source of confusion was a lack of understanding of the statistical mechanics concept of an "expectation value." An expectation value is a properly weighted average of a measureable quantity, for example, energy or magnetization, over the ensemble of states accessible to the system that can be used to predict the value of this quantity for the system at a given temperature. This scenario depicts that sometimes students struggle with the computational representation (MATLAB® code) not because a lack of ability to deal with it from the programming point of view, but because of a disciplinary misconception.

Ten students considered their prior experience with the computational module in class helpful for approaching this problem. This group thought the visual aids provided during computational module helped them to understand the concepts (see Q8 for a sample quote on Appendix A). It is not clear to what extent the perception of understanding that arises here from interacting with the visual output represents a deep form of learning, that is, an improvement in the students' internal representational model via a kind of visual intuition or a reinforcement of tentative internal representations that jibe with the visual feedback, or to what extent it represents a fragmented or fallacious sense of understanding through visual familiarity.

At the end of the interview, students were asked what additional information they would desire regarding the module topic and solution. (L)Justice_S, (L)Landry_S, (H)Sidney_S, (L) Phoenix_T, (H)Charlie_T, and (L)Emory_T mentioned that they would like to be able to understand how to apply this model to other areas. For example, (L)Justice_S said: "How else can the Monte Carlo be used other than just for nanotube domains or for a lattice in how ordered, disordered is it? Can it be used to simulate something else?" Student comments suggested that they would like to be able understand the model from a higher level of abstraction. They don't want to only solve a specific problem but to be able to extrapolate the model to other situations.

Within-Case Analysis 2: Configuring and Running Simulations With COMSOL®. We present here student computational modeling and simulation process, their challenges and benefits when using COMSOL®. This characterization was identified as part of the experiences of twelve participants interviewed to describe their experiences using the COMSOL® tool to complete the open-ended design problem. The participants in this study were given gender-neutral pseudo names per IRB guidelines. Module topic number three related to Kinetics and Phase Transformations where six of the participants [(H)Peyton_K, Charlie _K, (L)Tatum_K, (H)Phoenix_K, (L)Justice_K, (L) Heyden_K] performed a finite element analysis to characterize the diffusion in a spherically symmetric drug delivery system. In the fourth module topic, six of the participants [(H)Sidney_M, (H) Emerson_M, (H)Rowan_M, (L)Parker_M, (L)River_M, (L) Finley_M] completed an activity related to mechanical properties in which they were asked to use a finite element analysis to evaluate plastic necking of a cylindrical bar. The description of students' experiences using COMSOL® are summarized below.

Simulation Model Process. While describing the system behavior, (L)Tatum_K used prior experience to discuss the expected outcome, although his or her phrasing indicates a misconception regarding the distinct contributions of the geometry (surface-to-volume ratio) and material properties (diffusivity). For a sample of a quote see Q9 on Appendix A.

In the case of the Kinetics activities, four out of six students recalled aspects of building similar models as part of the course module exercises; these recollections helped them think through how to decompose modeling task. In the highlighted example, (L) Justice_K's prior experiences helped guide him through the process of setting up the model and identifying the challenge of setting up the model in COMSOL (see Q10 on Appendix A for a quote).

During the interview process, students explained some related concepts improperly. In the mechanics subject area, students' misconceptions were apparent when students discussed their analysis of the mechanical structure. Students struggled understanding the concepts of mechanical loading [(H)Rowan_M, (H)Emerson_M, (L)River_M, (L)Finley_M, (L)Parker_M, (H) Sidney_M]. For instance, when (H)Rowan_M was asked to discuss the implications of the analysis of the bar just modeled, the interviewer suggested "not confusing the conditions of plasticity and necking." (H)Rowan_M responded with the excerpt provided on Appendix A as Q11.

Simulation Development Process. A common challenge that students faced during the modeling activities was their limited knowledge of the COMSOL® features and tools that were useful in building or analyzing their design. It is important to note that students in most instances had no prior exposure to COMSOL®, and during the initial modules students used a "cookbook" approach to configure the problem for solution by COMSOL®. As shown in Appendix B, the structured interview asked students to explain work without being provided with step-by-step instructions. Most participants lacked the expertise regarding how to appropriately employ COMSOL® tools to achieve sub-task goals.

In the instances that related to Kinetics, students who were able to understand aspects of behavior characteristics that determined the system response still struggled to recall the configuration techniques needed to model the phenomena using COMSOL® (i.e., external computational representation) despite their ability to recall the conceptual facts related to the system behavior [(L)Justice_K, Charlie _K, (L)Heyden_K, (H)Peyton_K, (L)Tatum_K, (H)Phoenix_K]. In one instance (H)Charlie_K, was able to summarize his thoughts regarding how the suggested adjustments to the model would impact the diffusion profile of the drug delivery system. Although, (H)Charlie_K was able to express his expectations and was aware that COMSOL® could help analyze the changes, he relied on the interviewer to help him program his suggested changes in COMSOL®. The participants in the Mechanics activity experienced their own challenges with recalling knowledge related to how to set up and implement COMSOL®. In most cases students' actions and thoughts revealed their challenge with properly using the COMSOL® tool [(L)River_M, (L)Parker_M, (L)Finley_M, Armari_K].

(L)Finley_M advised the interviewer of his trial and error approach to changing his design in COMSOL® (see Q12 on Appendix A). Most students referred to different aspects of their experiences using COMSOL®; these past experiences then became a resource for reasoning, making decisions and setting their expectation during the computational design activity provided for them as part of the structured interview [(L)Justice_K, (L)Tatum_K, (L)Heyden_K, (H)Charlie_K, (L)River_M, (L)Finley_M, (L)Parker_M, (H)Rowan_M].

Simulation Experiment Process. In the case of the Kinetics activities, students recalled aspects of building similar external representations as part of the course module exercises; these recollections helped them think through how to decompose modeling task. As highlighted in the simulation model process, (L) Justice_K prior experiences help guide him through the process of setting up the model and identifying the challenge of setting up the model in COMSOL® (see Q13 on Appendix A for a representative quote). For (H)Rowan_M, his value in the experience of learning the COMSOL® tool for professional practice motivated him to acquire knowledge that was useful during the interview. He was able to complete most of the model set-up with very little assistance from the interviewer. This is highlighted when he refers back to his prior experiences with COMSOL® when asked what he learned for the computational module provided during the interview (see Q14 on Appendix A). From the quote Q14, it is not clear to what extent the perception of understanding that arises here from interacting with the visual output represents a deep form of learning, or to what extent it represents a fallacious or fragmented sense of understanding through visual familiarity.

Simulation Analysis Process. All the students in Mechanical Properties referred to the computational outputs of COMSOL® when formulating their judgments regarding the usefulness of their designs given the findings. These descriptions refer to the mechanical properties of the modeled structure in general and the qualitative aspects of the computational observations that supported their design judgments [(H)Rowan_M, (H)Sidney_M, (H)Emerson_M, (L)River_M, (L)Finley_M, (L)Parker_M]. For instance, four students [(L)Parker_M, (H)Rowan_M, (L)Finley_M, and (H)Emerson_M] relied on graphical output from the COMSOL® model to characterize the failure mode. Here (L)Parker_M refers to the 3D visualization produced by the model: "You can see that instead of the necking occurring down here—since we strengthened it, necking won't occur placed just in the middle. Between this fortified place and the top now yields because there's still that critical stress being applied."

When students were asked to reflect on what they learned by completing the computational modules they highlighted the value as a visual aid. The COMSOL® external representation were perceived as helpful in building conceptual understanding of the scientific concepts taught in the class lectures [(L)Heyden_K, (H) Phoenix_K, (H)Peyton_K, (L)Tatum_K, (L)Justice_K]. Examples of students' comments from the Kinetics course provide insight as to how students valued COMSOL® as a visualization learning tool. In the first example, (H)Phoenix_K highlighted the value of reinforcing what was learned during the class: "it allows you to visualize things . . . Cause you talk about it theoretically in class, be like, "Oh, it phase separates," but then when you do the computational modules you actually see how that happens and how it changes."

A student from the Kinetics activity highlighted the usefulness of the tool in helping her build deeper conceptual understanding beyond memorizing the material (see Q15 on Appendix A for a sample quote). Students in the mechanics subject area also noted the tool as a useful visualization aid for building their conceptual understanding of the subject [(H)Emerson_M, (H)Rowan_M, (L)Parker_M, Finely_M, (L)River_M]. (H)Emerson_M noted the usefulness of COMSOL® as a visual aid in his reflection regarding learning with the computational modules: "This is the first simulation for a visual aid we have in the classroom other than the book, but those are stagnant 2D drawings, so this is color and actually changes with time." Some other students perceived that the steps provided during the modules were limited in context regarding how the knowledge could be applied beyond the task goals. (L)River_M advised that the module be redesigned to help guide inquiry learning and metacognitive skills (see Q16 on Appendix A). Although students were aware of what they needed to set-up in COMSOL®, their lack of experience and fluency operating COMSOL® was articulated at various stages of their design process as they worked through the interview design activity [(H)Sidney_M, (H)Rowan_M, (L)Finley_M, (L)River_M, (L)Parker_M, and (H)Emerson_M].

CROSS-CASE ANALYSIS BETWEEN PROGRAMMING APPROACH AND CONFIGURING APPROACH AND DISCUSSION

This section aims to integrate and make sense of the overall findings of the study and to try to explain differences identified in quantitative portion of the study by going deeper into describing students' experiences during the interviews. Thus this section addresses the research question: What are affordances and challenges that students encountered when engaging in modeling and simulation processes following a programming or a configuring approach? Here, we relate and explain the main findings from the two previous sections, which were identified after comparing and contrasting students' modeling and simulation processes with their disciplinary learning gains. When we compared pretest and posttest assessments from all four computational modules, it was identified that students in the configuring approach on average reported statistical significant learning gains, as opposed to students exposed to the programming approach. We also identified no clear pattern on relationship (1) between high and low performing students who participated in the think aloud and the outcome of their experiences following during the modeling and simulation processes nor (2) between high and low performing students and their performance in the conceptual pretest and posttest assessments. However, we did identify commonalities and differences between instances in which students were asked to engage in tasks that required them to configure and execute models vis-à-vis instances in which student were also asked to engage in programming models. We were particularly interested in how each supported or hindered students' ability to make-meaning from different representations and to draw inferences on how students engaged in specific stages of the modeling and simulation process.

Challenges When Generating and Interacting-With Computational Tools

Students faced difficulties with the algorithmic representations when implementing the model or when operating it to establish the experimental conditions by means of its graphical user interface. Students' interactions with the software posed challenges that interfered with their ability to leverage their prior experiences to approach related problems. For MATLAB® learning modules, most of the students (eleven out of twelve), faced difficulties implementing the algorithmic representation due lack of understanding of: (1) The connection between the overall goal of the MC algorithm and the disciplinary concepts, (2) how certain variables represented or related to the mathematical model, and (3) the MATLAB® syntax (Within-Case Analysis 1: Simulation Experiment Process). For the COMSOL® learning modules, some other students encountered difficulties navigating the interface for setting up the model (Within-Case Analysis 2: Simulation Development Process). For the latter case, the software itself guided students through a series of interfaces that assisted their design and subsequent problem solving processes (Within-Case Analysis 2: Simulation Experiment Process). This process was not always fully understood by the students who were often unable to meaningfully relate them to mathematical or conceptual aspects of the system. It is notable that quantitative conceptual learning gains were observed in the COMSOL® exercises but not in the MATLAB® exercises although students, for the most part, had prior exposure to MATLAB® (Quantitative Results: Learning Performance). This may indicate that the complexity of the task of parsing and conceptualizing code, even when that code is partially provided in a known language and environment, distracts from disciplinary learning in ways that building models in GUI environments does not.

Building or programming simulations and/or computational representations appears to be a challenging task. For instance, literature in computer science education has consistently identified over years that learning to program is difficult [53–55]. Some of the difficulties that students might have encountered in this study include: (1) orientation, where the learner identifies the purpose of the programming task, (2) notation, where the learner must master the syntax and semantics of the programming language; (3) structure, where the learner needs to deal with the difficulties of acquiring standard patterns or schemas that can be implemented to attain small-scale goals, and (4) pragmatics, where the learner develops the skills to be able to specify, develop, test, and debug programs using whatever tools are available [56,57]. This study was performed in the context of a broader curriculum revision that involved sequencing the computational modules after a systematic introduction to programming through MATLAB®, but challenges arise when students change majors, take courses out of sequence or substitute programming coursework in other languages such as Java or Python. While programming is a transferrable skill, the need to learn new syntax presents a barrier, and furthermore, it is not clear that skill in programming translates into facility using GUI based software. Therefore, a tension exists between what level of transparency needs to be provided to learners when working with expert computational tools [36]. These tensions also exist in engineering education where educators have questioned the use of research tools as learning tools, the integration of computation as a tool that can help students create or adapt models to solve engineering problems, and the deployment of transparent simulation tools that can allow learners to inspect and even modify the underlying model [37].

Affordances When Generating and Interacting-With Computational Tools

Once implemented or configured, students view the models as useful tools for testing and experimentation when performing the simulation runs, and for making abstract concepts visible during the interpretation of results (Within-Case Analysis 1: Simulation Experiment Process, Simulation Analysis Process; Within-Case Analysis 2: Simulation Experiment Process, Simulation Analysis Process). Students reported benefiting from running or executing the models once they overcame the programming, configuring stages or the simulation stages as identified in Figure 1. Students find them as useful tools for experimentation and as tools for meaning making to help them understand systems behavior or to enhance conceptual learning. However, deeper investigation is needed to identify what is the effect of these expert tools in supporting students' learning gains.

When considering students' perceptions, previous studies have reported the educational value expert computational tools can provide in helping students make meaning of difficult concepts in engineering [35]. Some other studies have identified significant differences between pretest and posttest assessments measuring student conceptual understanding, after being exposed to laboratory experiences with modeling and simulation tools [34]. However, there is still little to no research that supports student mastering learning goals after being exposed to integrated expert modeling and simulation tools. Other possible explanations for students' inability to master learning objectives after being exposed to simulation tools, and in particular as related to expert modeling and simulation tools, could be attributed to the complexity of the simulation task [13], as it was the case of the activities presented as part of this study. Another possible explanation could be attributed to the conceptual overload caused by the exploratory process [58], which in this case seemed to be caused by the MATLAB® interface in which students need to write the programming code, or the complex graphical user interface in COMSOL®.

Challenges for Meaning-Making With Computational Tools

Students seemed to have particular trouble mapping between their internal representations (i.e., conceptual model) and the external representations (mathematical or computational model) and from the mathematical representation to the computational instantiation when engaged in a programming approach (Within-Case Analysis 1: Simulation Development Process, Simulation Experiment Process). To describe the systems' components and behaviors, students used different forms of representations provided either in the protocol or additional ones students derived from prior activities or learning experiences. In the process students were required to make connections between these representations in order to be able to proceed with the problem solving task. Often these artifacts were used as tools to describe their understanding of the system and later on were used to set up their models. Students were observed to infer improper or misleading relationships between these representations that interfered with their solution of the problem at hand. This issue seemed more problematic for students using programming models with MATLAB® because more complex mappings from mathematical representations to algorithmic representations and programming codes were required (Within-Case Analysis 1: Simulation Experiment Process). At the same time, we did observe students make reverse mappings from codes provided to mathematical representations, and these mappings appeared to help them overcome hurdles that they could not surmount relying on their internal models alone to generate mathematical representations (Within-Case Analysis 1: Simulation Development Process). However, we suspect that the added complexity required to map from mathematical representations to algorithmic representations to codes added sufficient complexity to the problem that students' attention were diverted from the acquisition of disciplinary concepts.

We speculate on the possibility that learners find it difficult to create representations because of (1) lack of the required previous knowledge [59], particularly in terms of devising or interpreting the physical model, or lack of understanding of the underlying systems in which they are working [60], (2) a struggle in applying or mapping knowledge about graphical representations while simultaneously comprehending new domain knowledge [61,62], (3) the arbitrary nature of representations because learners do not continuously participate in these practices [63,64]; or the level of difficulty of the disciplinary concepts. Specifically, students' lack of prior knowledge, particularly as related to the physical model (not their conceptual understanding), was a strong limitation on their abilities to abstract the physical phenomena to represent the given system mathematically, and subsequently connecting that representation with the computational aspects of the simulation. Previous knowledge is an important component for students to be able to effectively use representations [59]. Studies that have compared novices and experts revealed that experts use representations more proficiently as tools together with domain knowledge [61,65,66].

Similar studies [67,68], found that experts recalled elements on representations in patterns based on principles rather than on surface features, as novices did [66]; that they used these devices as tools with which to think [62,66,69]; and that differences in searches between novices and experts are in dissimilar problem spaces [70]. The novice learners who participated in this study seemed to have difficulties recalling the conceptual aspects of the disciplinary problem, and therefore, were often unable to make connections between the mathematical and computational aspects of the problem solution.

Affordances for Meaning-Making With Computational Tools

It was particularly notable that students experienced strong affinity with visual and graphical representations, particularly those that translated associated concepts into time-dependent, 3D and/or color images (Within-Case Analysis 1: Simulation Analysis Process; Within-Case Analysis 2: Simulation Analysis Process). Students characterized these as being more intuitive and realistic than mathematical representations. A majority of students identified these aspects of visualization as the most useful catalyst for their learning. However, we are skeptical of student self-reported claims because students using both the configuring and programming approaches and then executing the models, made these claims; while only the students who configured models using COMSOL® showed significant learning gains with respect to disciplinary concepts (Quantitative Results: Learning Performance).

While graphical representations have been identified as ways to enhance engineering problem-solving and scientific-inquiry skills [71] by making the content accessible to students in a more learnable or concise way [72], research on the effectiveness of inducing the use of different forms of representation, such as learner-generated drawings, has demonstrated inconsistent results [73,74]. We believe that although these students used representations to explain their conceptual understanding and to walk through the problem-solving process, some of them were unable to use them effectively, and therefore, failed to successfully complete their assignments. By interacting with the computational representation in MATLAB® and COMSOL®, some students appear to have experienced cognitive conflict, confronting new information that may have contradicted their prior beliefs [75]. Although these students might have experienced an initial step in their learning process, some of them failed to experience learning gains in their disciplinary knowledge, particularly as identified for the programming approach (Within-Case Analysis 1: Simulation Analysis Process).

IMPLICATIONS FOR LEARNING

Our analysis suggests that for students' to successfully understand the problem and achieve a solution requires appropriate interactions between internal (e.g., mental models, thought experiments) or external representations (e.g., computational models) [4]. In this study, different forms of representations were identified. The internal representations that we refer to here as "conceptual" models may take the form of mental models or thought experiments, among others. To deploy their conceptual model for the purpose of understanding a system, students first have to translate their conceptual models into a "set of rules and structures that can be represented in mathematical terms using a programming or modeling language" [76]. Mathematical models use mathematical equations to represent the key relationships among system components, which then are translated into other forms of representations afforded by the computational tools.

> By executing the simulation students can ask "what if" questions about the system. Changes are made in the physical conditions or their mathematical representation and the model is run many times to "simulate" the impacts of the changes in the conditions. The model results are then compared to gain insight into the behavior of the system [76].

Once the computational tool depicts the output of the simulation, the learner must map this visual representation back onto their conceptual model in order to improve their understanding of the system and potentially to re-evaluate their own conceptual model of the underlying system or the scientific principles by which it is understood. However, affordances of different tools can influence the process just described. For the case of students programming models with MATLAB®, students must map their conceptual model to a mathematical representation, which is then mapped by the student to an algorithmic representation taking the form of a simulation that transforms the solution into a new visual representation that could be a data plot, diagram or animation. For students configuring and executing models in COMSOL®, they must map their conceptual model to a set of schematics or diagrams to identify the system to be modeled mathematically, and establish the relationships among variables. Then, students must translate these variables and manipulate them to configure the simulation software in order to identify the system's behavior via new representations created by the simulation software.

Figure 4 depicts observed differences on students' modeling and simulation processes when engaged in either a programming approach or a configuring approach. Our findings suggest that students, in general, experienced the highest challenges in the "simulation development" stages. Our findings also suggest that those students who might have been capable of overcoming those challenges, may have also benefited in acquiring disciplinary knowledge in the "simulation experiment and analysis" stages, particularly if engaged in a configuring approach. However, students in the programming approach seem to need additional supports to overcome such challenges, particularly as related to the mapping of the mathematical representations, and then to the algorithmic representations.

Previous research conducted with experts has identified that external representations provide specificity to the mental model, providing details and constraints, and at the same time off-load cognitive load [4]. However, our findings suggest that for novice learners, integrating modeling and simulation in undergraduate engineering education is a form of complex learning. For students to effectively use these tools, they must integrate disciplinary knowledge with mathematical representations and at the same time deal with aspects of scientific thinking, engineering thinking (i.e., inquiry processes, problem solving, design), and computational

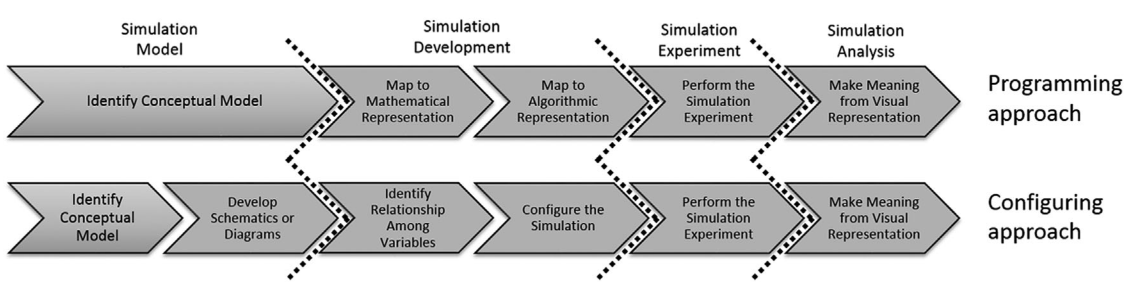


Figure 4 Observed differences of modeling and simulation processes between a programming approach and a configuring approach.

thinking. For these novice learners whose computational literacy may not be fully developed, this integration may have caused cognitive overload. In our introduction, we generally defined modeling as a process consisting of producing or setting-up a model to represent the inner workings of a system [17]. Our findings suggest that modeling is a very challenging process for novice learners that may be a source of confusion or difficulty. This difficulty may be at times caused by lack of prior knowledge [77], students' inability to map between representations [78], students' difficulties in applying algorithmic thinking [79], or complexities presented by a graphical user interface for operating, or specifically in this case, for configuring a model [80]. We also defined simulation as the process of operating a computer model that can be reconfigured by changing input parameters so that variations in its performance can be explored [17]. Our findings suggest that students may have benefited from engaging in simulation processes, specifically by analyzing the simulation output: (1) Eliciting cognitive conflict allowing them to identify their own misconceptions [75], and (2) making the content accessible in a more learnable way [72].

The implications for learning can be related to Cognitive Load Theory (CLT) [81] and complex learning [82]. CLT describes how learning occurs through a cognitive architecture composed by a limited working memory and a vast long-term memory. When students are exposed to many interacting information elements, students are overwhelmed given the limited working memory capacity. [83]. It is generally accepted that performance worsens at cognitive load extremes. That is, both excessive underload or excessive overload conditions can affect learning negatively [83]. Experts do not experience cognitive overload in the same situations as novices because they have previous knowledge that help them to make sense of the interacting elements, thus loading them as a unit in the working memory. In our study, novice learners who lacked the required prior knowledge may not have benefited from the representations and thus struggled to gain a deeper understanding of their disciplinary knowledge (see Table 7). We then hypothesize that these students may have experienced cognitive overload because they needed additional background knowledge to make sense of the disciplinary phenomenon they were simulating. This is consistent with prior research that has identified the critical role of prior knowledge in determining the impact of visual representations on learners' cognitive structures and processes [77]. The qualitative analysis also suggests that some students struggled to understand the MATLAB® representations, either because these were too abstract or because they were not familiar with MATLAB® programing. The result suggests that for modules that include programming components, additional scaffolding may be required to reduce the cognitive load.

Another source of cognitive load could have resulted from students' difficulties in handling the operational features of the computational tool due to their inexperience with the graphical user interface. These students seemed to have experienced extraneous load when making sense of elements that were not directly beneficial towards learning, thus depleting cognitive resources. These resources, that otherwise would have been devoted to intrinsic load (i.e., to create and automate schemata in the long term memory), were then devoted to extraneous load [83]. Specifically, these students may not have fully benefited from the learning experience because they were devoting their cognitive resources to secondary or unessential issues such as the functional or operational characteristics of the software, instead of focusing on the representations themselves.

SUMMARY, LIMITATIONS, AND CONCLUSION

This study presented a description of how undergraduate materials science and engineering students engaged with modeling and simulation practices using different computational approaches, and how different approaches related to their disciplinary learning. Our findings suggest that novice learners may encounter challenges, particularly in the simulation development stage, when they have to formulate and develop the model, or when they have to select and establish experimental conditions for the simulation runs by configuring a model via a graphical user interface. We hypothesize that these tasks seem particularly challenging in part because students may lack the appropriate prior knowledge or the required programming skills or in part because of the complexity of the software. On the other hand, once students completed the modeling stage, they seem to benefit from the simulation experiment and simulation analysis stages. Students, specifically in the configuring approach, seemed to benefit from testing their ideas by performing the simulation runs and benefit from the simulation output when engaging in the interpretation and presentation of results. Throughout the study we offered instances of how students explained their understanding of the problem to be solved and how they tried to draw upon their prior knowledge and make that explicit to us by means of the representations they used. We also offered instances in which students attempted to make connections between different forms of conceptual, mathematical, and computational representations, particularly for the programming approach. Through the different categories and overall findings identified we were able to recognize what were major challenges students encountered when experiencing modeling and simulation practices. We argue that integrating these practices at the undergraduate level represents a form of complex learning. From the explanations provided we can infer that the computational tools, at times, were supporting disciplinary learning, but at other times were interfering with it. We therefore emphasize the importance of the integration of direct instruction or delayed instruction methods to support the process. Such methods should be designed in ways to offer feedback to learners providing consolidation checkpoints where they can test the representational connections being formed so that fallacious or fragmented connections do not hinder learning.

The limitations of the study relate to the naturalistic nature of the educational innovation that resulted in the researchers' inability to control for the possible influence of previous exposure to computational learning experiences the preceding semester, or additional exposure to computational modules the same semester, but on other courses. However, because of (1) the very different nature of the modules and the computational tools, (2) the differences in disciplinary knowledge measured through the conceptual assessments, and (3) no clear patterns between students' performance in the module and the conceptual tests; we believe this possible interaction may have had a minor effect. Regardless of this limitation, we believe that this exploration provides in-depth descriptions of what benefits or struggles students experienced throughout different stages of the modeling and simulation process afforded by different computational tools, and therefore provides some insights into how we can help students overcome challenges during their learning processes.

In conclusion, it is important to couple these instructional strategies with research that is conducted in naturalistic classroom settings because many of the results from contrastive studies on expertise (i.e., experts versus novices) do not translate easily into instructional interventions about training for the acquisition of expertise [70]. Our work will continue to explore how expert tools and practices that are commonly used in engineering workplaces can be effectively integrated into engineering classrooms. To accomplish this goal, it is important to use design-based research procedures to identify students' struggles so that we can design and redesign learning experiences in response. Our ultimate goal is to identify design principles that can help us design learning experiences that effectively integrate computational modeling and simulation practices and tools, because these are increasingly critical in industry and research enterprises.

ACKNOWLEDGMENTS

Research reported in this paper was supported in part by the U.S. National Science Foundation under the awards #EEC1449238 and #EEC1137006. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Science Foundation.

REFERENCES

[1] S. Emmott and S. Rison, Towards 2020 science, Science in Parliament 65 (2008), 31–33.
[2] C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey, and L. J. Leifer, Engineering design thinking, teaching, and learning, J Eng Educ 94 (2005), 103–120.
[3] N. J. Nersessian, Model-based reasoning in conceptual change. In: Model-based reasoning in scientific discovery, L. Magnani, N. J. Nersessian, and P. Thagard (Eds.), Kluwer Academic/Plenum Publishers, New York, 1999, pp 5–57.
[4] J. Aurigemma, S. Chandrasekharan, N. J. Nersessian, and W. Newstetter, Turning experiments into objects: The cognitive processes involved in the design of a lab-on-a-chip device, J Eng Educ 102 (2013), 117–140.
[5] E. Hamilton, R. Lesh, F. Lester, and M. Brilleslyper, Model-eliciting activities (MEAs) as a bridge between engineering education research and mathematics education research, Adv Eng Educ 1 (2008), 1–25.
[6] A. J. Magana, S. P. Brophy, and G. M. Bodner, Instructors' intended learning outcomes for using computational simulations as learning tools, J Eng Educ 101 (2012), 220–243.
[7] C. Hu, Integrating modern research into numerical computation education, Comput Sci Eng 9 (2007), 78–81.
[8] J. Gibson, The ecological approach to visual perception, Lawrence Erlbaum Associates, Hillsdale, NJ, 1979.
[9] J. Gibson, The ecological approach to visual perception: Classic edition, Psychology Press, New York, 2014.
[10] L. T. Louca and Z. C. Zacharia, Modeling-based learning in science education: Cognitive, metacognitive, social, material and epistemological contributions, Educ Rev 64 (2012), 471–492.
[11] C. D'Angelo, D. Rutstein, C. Harris, R. Bernard, E. Borokhovski, and G. Haertel, Simulations for STEM learning: Systematic review and meta-analysis, SRI International, Menlo Park, 2014.
[12] W. Feurzeig and N. Roberts, Modeling and simulation in science and mathematics education, Vol. 1, Springer, New York, NY, 1999.
[13] R. Clariana and J. Strobel, AECT handbook of educational communications and technology, 3rd ed., J. M. Spector, M. D. Merril, J. J. G. Van Merrienboer, and M. P. Driscoll (Eds.), Lawrence Erlbaum Associates, Mahwah, NJ, 2007, pp 329–344.
[14] S. Alessi, Building versus using simulations. Integrated and holistic perspectives on learning, ins truction, and technology, Springer, Netherlands, 2002, pp 175–196.
[15] B. Du Boulay, T. O'Shea, and J. Monk, The black box inside the glass box: Presenting computing concepts to novices, Int J Hum-Comput St 51 (1999), 265–277.
[16] AAAS. Science for All Americans, American Association for the Advancement of Science, New York, 1989.
[17] A. Maria, Introduction to modeling and simulation, In: Proceedings of the 29th conference on Winter Simulation, Atlanta, GA, 1997, pp 7–13.
[18] D. Fraser, R. Pillay, L. Tjatindi, and J. Case, Enhancing the learning of fluid mechanics using computer simulations, J Eng Educ 96 (2007), 381–388.
[19] J. O. Campbell, J. R. Bourne, P. J. Mosterman, and A. J. Brodersen, The effectiveness of learning simulations for electronic laboratories, J Eng Educ 91 (2002), 81–87.
[20] J. Mackenzie, W. Earl, R. Allen, and I. Gilmour, Amoco computer simulation in chemical engineering education, J Eng Educ 90 (2001), 331–345.
[21] B. Vahidi and M. Taherkhani, Teaching short circuit breaking test on high-voltage circuit breakers to undergraduate students by using MATLAB-SIMULINK, Comput Appl Eng Educ 21 (2013), 459–466.
[22] R. B. Kozma and J. Russell, Students becoming chemists: Developing representationl competence. Visualization in science education, Springer, Netherlands, 2005, pp 121–145.
[23] M. Stieff, Improving representational competence using molecular simulations embedded in inquiry activities, J Res Sci Teach 48 (2011), 1137–1158.
[24] W. R. Van Joolingen, T. De Jong, and A. Dimitrakopoulou, Issues in computer supported inquiry learning in science, J Comput Assist Lear 23 (2007), 111–119.
[25] M. d. R. Uribe, A. J. Magana, J. H. Bahk, and A. Shakouri, Computational simulations as virtual laboratories for online engineering education: A case study in the field of thermoelectricity, Comput Appl Eng Educ 24 (2016), 428–442.
[26] S. S. Moor and P. R. Piergiovanni, Process control kits: A hardware and software resource, Comput Appl Eng Educ 21 (2013), 491–502.
[27] M. Shacham, An introductory course of modeling and computation for chemical engineers, Comput Appl Eng Educ 13 (2005), 137–145.
[28] X. L. Zhou and J. H. Wang, Interactive computer for teaching biot poroelasticity modeling in civil engineering, Comput Appl Eng Educ 24 (2016), 5–15.
[29] A. J. Magana and G. S. Coutinho, Modeling and simulation practices for a computational thinking-enabled engineering workforce, Comput Appl Eng Educ 25 (2017), 62–78.
[30] C. E. Vergara, M. Urban-Lurain, C. Dresen, T. Coxen, T. MacFarlane, K. Frazier, D. Briedis, N. Buch, A.-H. Esfahanian, L. Paquette, J. Sticklen, J. LaPrad, and T. F. Wolff, Aligning computing education with engineering workforce computational needs New curricular directions to improve computational thinking in engineering graduates, In: Frontiers in education anual conference, San Antonio, TX, 2009, pp 1–6.
[31] L. D. Feisel and A. J. Rosa, The role of laboratory in undergraduate engineering education, J Eng Educ 94 (2005), 121–130.
[32] O. Alabi, A. J. Magana, and R. E. Garcia, Gibbs, comptuational simulation as a teaching tool for students' understanding of thermodynamics of materials concepts, J Mater Educ 37 (2015), 239–260.
[33] R. Mansbach, A. Ferguson, K. Killian, J. Krogstadt, C. Leal, A. Schleife, and G. L. Herman, Reforming an undergraduate materials science curriculum with computational modules, J Mater Educ 38 (2016), 161–174.
[34] S. P. Brophy, A. J. Magana, and A. Strachan, Lectures and simulation laboratories to improve learners' conceptual understanding, Adv Eng Educ 3 (2013), 1–27.
[35] A. J. Magana, S. P. Brophy, and G. M. Bodner, An exploratory study of engineering and science students' perceptions of nanohub.org simulations, Int J Eng Educ 28 (2012), 1019–1032.
[36] A. J. Magana, S. P. Brophy, and G. M. Bodner, Student views of engineering professors technological pedagogical content knowledge for integrating computational simulation tools in nanoscale science and engineering, Int J Eng Educ 28 (2012), 1033–1045.
[37] A. J. Magana, S. P. Brophy, and G. M. Bodner, The transparency paradox: computational simulations as learning tools for engineering graduate education, Presented at the Annual AERA Meeting: Understanding complex Ecologies in a Changing World, Denver, Colorado, 2010.

[38] The MathWorks, April 18, 2014, MatWorks. Accelerating the pace of engineering and science. Available: http://www.mathworks.com/products/matlab/?s_tid=hp_fp_ml.
[39] COMSOL, April 15, 2014, COMSOL multiphysics. Available: http://www.comsol.com/comsol-multiphysics.
[40] N. V. Ivankova, J. W. Creswell, and S. L. Stick, Using mixed-methods sequential explanatory design: From theory to practice, Field Methods 18 (2006), 3–20.
[41] J. W. Creswell, V. L. Plano, Clark, M. L. Gutmann, and W. E. Hanson, Advanced mixed methods research designs. Handbook of mixed methods in social and behavioral research, Sage Publications, Thousand Oaks, CA, 2003, pp 209–240.
[42] S. Stick and N. Ivankova, A decade of innovation and success in virtual learning: A world-wide asynchronous graduate program in educational leadership and higher education, Online J Dist Lear Admin 7 (2004), 1–14.
[43] R. K. Yin, Case study research: Design and methods, Vol. 5, Sage Publications Inc, Thousand Oaks, CA, 2009.
[44] M. Q. Patton, Qualitative research and evaluation methods, 2nd ed., Sage Publications Inc, Thousand Oaks, CA, 2002.
[45] M. B. Miles and A. M. Huberman, Qualitative data analysis: An expanded sourcebook, Sage Publications Inc, Thousand Oaks, CA, 1994.
[46] S. B. Merriam, Case study research in education. A qualitative approach, Jossey-Bass, San Francisco, CA, 1988.
[47] N. K. Denzin and Y. S. Lincoln, Strategies of qualitative inquiry, 3rd ed., Sage Publications Inc, Thousands Oaks, CA, 2008.
[48] V. Braun and V. Clarke, Using thematic analysis in psychology, Qual Res Psychol 3 (2006), 77–101.
[49] B. L. Berg, Qualitative research methods for the social sciences, Allyn and Bacon, New York, 1989.
[50] A. L. Strauss and J. Corbin, Basics of qualitative research: Grounded theory procedures and techniques, Vol. 1 Sage Publications Inc, Thousands Oaks, CA, 1990.
[51] K. M. Eisenhardt, Building theories from case study research, Acad Manage Rev 14 (1989), 532–550.
[52] J. P. Barber and K. K. Walczak, Conscience and critic: Peer debriefing strategies in grounded theory research, Presented at the Annual Meeting of the American Educational Research Association, San Diego, CA, 2009.
[53] E. Soloway and J. C. Spohrer, Studying the novice programmer, Lawrence Erlbaum, Hillsdale, NJ, 1989.
[54] R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. McCartney, J. E. Moström, K. Sanders, O. Seppälä, and B. A. Simon, A multi-national study of reading and tracing skills in novice programmers, ACM SIGCSE Bullet 36 (2004), 119–150.
[55] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz, A multi-national, multi-institutional study of assessment of programming skills of first-year CS students, ACM SIGCSE Bullet 33 (2001), 125–180.
[56] B. Du Boulay, Some difficulties of learning to program. In: Studying the novice programmer, E. Soloway and J. C. Spohrer (Eds.), Lawrence Erlbaum, Hillsdale, NJ, 1986, pp 283–299.
[57] R. D. Pea and D. M. Kurland, On the cognitive prerequisites of learning computer programming, New Ideas Psychol 2 (1983), 137–168.
[58] M. Njoo and T. De Jong, Exploratory learning with a computer simulation for control theory: Learning processes and instructional support, J Res Sci Teach 30 (1993), 821–44.
[59] K. Higley, T. Litzinger, P. Van Meter, C. B. Masters, and J. Kulikowich, Effects of conceptual understanding, math and visualization skills on problem-solving in statics, In Proceedings of the 114th American Society of Engineering Education Annual Conference. Honolulu, HI, 2007.
[60] D. Jonassen, Y. H. Cho, and C. Wexler, Facilitating problem-solving tranfer in physics, In Proceedings of the 115th American Society of Engineering Education Annual Conference. Pittsburgh, PA, 2008.
[61] S. P. Brophy and S. Li, A framework for using graphical representations as assessments of engineering thinking, In Proceedings of the 117th American Society of Engineering Education Annual Conference. Louisville, KY, 2010.
[62] A. C. H. Kindfield, Biology diagrams: Tools to think with, J Lear Sci 3 (1993), 1–36.
[63] W. M. Roth, Toward an anthropology of graphing: Semiotic and activity-theoretic perspectives, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.
[64] W. M. Roth and G. M. Bowen, Digitizing lizards or the topology of vision in ecological fieldwork, Soc Stud Sci 29 (1999), 719.
[65] J. Bransford and D. L. Schwartz, Rethinking Transfer: A simple proposal with multiple implications, Rev Res Educ 24 (1999), 61–100.
[66] R. B. Kozma, The use of multiple representations and the social construction of understanding in chemistry. In: Innovations in science and mathematics education: Advanced designs for technologies of learning, R. Jacobson and R. B. Kozma (Eds.), Erlbaum, Mahawah, NJ, 2000, pp 11–46.
[67] M. T. H. Chi, P. J. Feltovich, and R. Glaser, Representation of physics knowledge by experts and novices, (NR 157-421 ONR Tech. Report 2). Learning, Research, and Development Center, University of Pittsburgh, Pittsburg, PA, 1980.
[68] J. Bransford, How people learn: Brain, mind, experience, and school, National Academies Press, Washington, DC, 2000.
[69] R. B. Kozma, E. Chin, J. Russell, and N. Marx, The roles of representations and tools in the chemistry laboratory and their implications for chemistry learning, J Lear Sci 9 (2000), 105–143.
[70] M. T. H. Chi, Theoretical perspectives, methodological approaches, and trends in the study of expertise. In: Expertise in mathematics instruction: Advancing research and practice from an international perspective, Y. Li and G. Kaiser (Eds.), Springer, USA.
[71] J. Bedward, E. Wiebe, L. Madden, J. Minogue, and M. Carter, Graphic literacy in elementary science education: Enhancing inquiry, engineering problem solving, and reasoning skills, In Proceedings of the 116th American Society of Engineering Education Annual Conference. Austin, TX, 2009.
[72] P. Blikstein, Assessing open-ended scientific computer modeling in engineering education: The role of representations, In Proceedings of the 117th American Society of Engineering Education Annual Conference. Louisville, KY, 2010.
[73] C. W. Anderson, Inscriptions and science learning, J Res Sci Teach 36 (1999), 973–974.
[74] P. Van Meter and J. Garner, The promise and practice of learner-generated drawing: Literature review and synthesis, Educ Psychol Rev 17 (2005), 285–325.
[75] J. Clement and M. Ramirez, The role of dissonance in conceptual change, In Proceedings of the National Association for Research in Science Teaching, 1998.
[76] AcqNotes, Introduction to modeling and simulation, 2014. Retrieved from: http://www.acqnotes.com/Attachments/Introduction%20to%20Modeling%20and%20Simulation.pdf
[77] M. P. Cook, Visual representations in science education: The influence of prior knowledge and cognitive load theory on instructional design principles, Sci Educ 90 (2006), 1073–1091.
[78] W. M. McCracken and W. C. Newstetter, Text to diagram to symbol: Representational transformations in problem-solving, In Frontiers in Education Conference, Reno, Nevada, 2001.
[79] A. H. Schoenfeld, Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. Handbook of research on mathematics teaching and learning, Macmillan, New York, 1992, pp 334–370.
[80] T. de Jong and W. R. Van Joolingen, Scientific discovery learning with computer simulations of conceptual domains, Rev Educ Res 68 (1998), 179–201.
[81] J. Sweller, Cognitive load during problem solving: Effects on learning, Cognitive Sc 12 (1988), 257–285.
[82] J. J. G. Van Merriënboer and J. Sweller, Cognitive load theory and complex learning: Recent developments and future directions, Educ Psychol Rev 17 (2005), 147–177.
[83] F. Paas, A. Renkl and J. Sweller, Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture, Instr Sci 32 (2004), 1–8.

Sample of Representative Quotes From Students' Responses

No.	Case: M&S step and student representative quote
Case 1: Programming and executing models with MATLAB®: Simulation model process	
Q1	(H)Charlie_T: That is a rule. At higher and higher temperature, the interaction energy is smaller and smaller, compared to like the overall energy of the system. So it matters less and less. So at higher temperatures you could have more and more configurations. Okay, so then tie that to average magnetization. So the magnetization is higher when they are all aligned in a specific way. So it means at lower temperatures you would have higher magnetization. Because at higher temperatures you have more random configurations. Which means you have lower magnetization. At lower temperatures, that interaction energy is more important; so they tend to align. And if they are aligned then they will have a higher magnetization
Q2	(L)Emerson_T: In general, from thermodynamics, I would say that at high temperatures that involve high energy systems, so the total [magnetization] value would be positive; let us say so the orientations would most likely be opposite at high temperatures and at low temperatures they would be more likely to look like tube B [pointing at Figure], where they are all oriented in the same direction because that would lower the temperature, lower the energy
Case 1: Programming and executing models with MATLAB®: Simulation development process	
Q3	(L)Emory_T: I guess we could relate energy to magnetization. No? I am trying to think how similar the equation or the approach would be to the previous one because I feel without anything relating to the magnetization like variable wise it probably has to be similar to that. In fact I am thinking it is like almost the same except with a B but I do not know
Q4	(H)Sidney_S: So I am thinking that if for each domain, the only ones that matter are the ones that are next to each other. And that is what you need to sort of add up. And if they are different, then the product would be negative. If they are the same, they would be positive. So I would do—if we call this S1, S2, S3, and S4, I would say, oh, the sum of the products of S1, S2 plus S2, to S3 plus S3, S4—that is the number that would be, well, maybe there for—if they are the same, they might be positive. So the more positive value, the lower the energy will be. So just like it is negative.
Q5	(H)Charlie_T: So we just take our little line here for calculating magnetization (Question 3 Equation). And so as energy sum we just have a new one that is magnetization. And that gets—I guess should I just assume we have four total? . . . you add it to the main position
Case 1: Programming and executing models with MATLAB®: Simulation experiment process	
Q6	(L)Emerson_T: Actually just like sitting down in front of MATLAB® just testing new things definitely helps; you just have to spend some time playing with it. This exercise was extremely similar to the computational modules I would say. I did the Monte Carlo essentially twice in structures and in thermodynamics so it is pretty embedded in my mind by this point. I think like the programs definitely help some with comprehending the material because it is kind of hard at least for me it is hard to imagine
Q7	(H)Skyler_T: "The programming itself ... Because you have to know it [the mathematical representation] to make the program work. You can't just wing it"
Case 1: Programming and executing models with MATLAB®: Simulation analysis process	
Q8	(H)Skyler_T: Seeing the colors change was more—Like when the program was running and there was like, they had the mesh grid with the blue and red, that made it clearer than the writing, but, you know, it is pretty math-intensive. You have to read it once or twice before it like seems reasonable
Case 2: Configuring and running simulations with COMSOL®: Simulation model process	
Q9	(L)Tatum_K: So let us just talk about the diffusion mechanism versus its practical application. So, I think that since there is a greater surface-to-volume ratio, it would be good, because we saw that with—when we compared the sphere and the thin-films, since the sphere had a larger surface-to-volume ratio, the drug was able to go through material more consistently, and so, I think in this case, since we have an even greater surface-to-volume ratio, you would see better diffusivity towards the center
Q10	(L)Justice _K: So in the same way it was normally set up, I think you would want to create another sphere. This time it tells you the new radius of the outside. So you would want to create a new sphere, cut out a square again, and then you want to, like, use what current eighth of the sphere you have and use that to cut out your new drug-absorbing sphere. I actually have played around with COMSOL a little bit and I actually found that kind of hard to do, to create a shape within in a shape—to cut it out. Because what happened when I did, it would fuse material that would overlap within it. So. I guess I am not exactly sure how to do that yet
Q11	(H)Rowan_M: Yeah, essentially, it means if we pass that threshold value where we are deviating—which we actually prescribed in the plasticity part. If we pass that threshold value where the material ceases to behave elastically and begins to behave plastically then the material is going to fail because of necking, or at least it is going to lose its properties that make it a valuable material
Case 2: Configuring and running simulations with COMSOL®: Simulation development process	
Q12	(L)Finley_M: "Well I'm just going to go and see if I can add a section to the middle <laughs>....which is 7 millimeters instead of. . . Well I don't know really how COMSOL® works; so I'm going to try and figure it out as I go"
Case 2: Configuring and running simulations with COMSOL®: Simulation experiment process	
Q13	(L)Justice_K: So in the same way it was normally set up, I think you would want to create another sphere. I actually have played around with COMSOL® a little bit and I actually found that kind of hard to do, to create a shape within in a shape—to cut it out. Because what happened when I did, it would fuse material that would overlap within it. So. I guess I am not exactly sure how to do that yet
Q14	(H)Rowan_M: I think the primary thing I learned from this computational module is actually how to use COMSOL®, which was quite nice because I am required to learn it for work. And so this was really a helpful introduction for me, you know, sort of throwing you under the bus and learning how to use it because it is not intuitive your first time around. But after running this computational module I sort of started to develop an intuitive grasp of like the user interface
Case 2: Configuring and running simulations with COMSOL®: Simulation analysis process	
Q15	(L)Justice_K: A lot of—it is not just equation manipulating anymore. We can watch what an equation does to the simulation. If we were to change the way the equation was defined, then the simulation would work out much differently as well. It also—I think it is really important towards reinforcing the material, because if you tell me to think of Fick's second law—if I had not gone through this, I might be more inclined towards memorizing an equation and all the parts of the equation, but I can associate something with this
Q16	(L)River_M: I mean I liked how this [the think aloud exercise] was kinda set up to make you think about each step and I think that may have helped me to kinda just like—maybe not like something that I had to turn in, but more so just like questions to help me think through if I was not getting it. So, I would know what questions to ask while you guys were there if I was not understanding the concept, instead of just following in the steps

APPENDIX B

Sample Interview Questions

Part I: Please take a moment to read the purpose of the activity.

Prompts: Now that you have read it, what are your initial thoughts of going about solving the exercise as a whole? Where would you start? What steps would you take? Please work through the exercise in any sequence that makes the most sense to you. As you are working on each part please explain your thinking as best you can.

Part II: Please describe a plan and a prediction of your expectations.

Prompts: [conceptual questions asked associated with the scenario]. Explain your reasoning.

Part III [programming]: Please describe the model of your analysis.

What model are you planning to use? Please explain your answer. How would you write a mathematical expression for the system in terms of the given parameters? Explain your proposed expression to the best of your ability. How would you go about to implement your model in the computer? Explain to the best of your ability what this algorithm accomplishes and why one would use it.

Part III [configuring]: Please describe the setup of your analysis.

Prompts: Why has the system been configured in such way? How this configuration relates to the problem you are trying to solve? How and why were the boundary conditions set as they were? What are your initial thoughts about adjusting the model? As you make each alteration to the model, please explain your reasoning.

Part IV: Please describe the results of your solution.

Prompts: What did this computational analysis show? What limitations does it imply for the system in question? How might you determine if this is a good or bad result? Why?

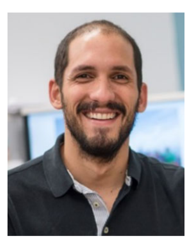
Generic Prompts: Can you clarify for me why you think it should be that way? Can you please tell more about you reasoning process here? That is correct. Can you explain how you came to the correct answer? Can you back up to this step and explain to me how you got from this step to your present answer?

Final reflection: What parts of this exercise were most confusing or challenging? What concepts came up in this exercise that you feel you would like to understand more clearly?

BIOGRAPHIES

Alejandra J. Magana is an associate professor in the Department of Computer and Information Technology with a courtesy appointment at the School of Engineering Education at Purdue University. She holds a BE in Information Systems and a MS in Technology, both from Tec de Monterrey; and a MS in Educational Technology and a PhD in Engineering Education, both from Purdue University. Her research program investigates how model-based cognition in Science, Technology, Engineering, and Mathematics (STEM) can be better supported by means of expert technological and computing software such as computational modeling and simulation tools. In 2015, Dr. Magana received the National Science Foundation's Faculty Early Career Development (CAREER) Award, to investigate modeling and simulation practices in undergraduate engineering education. In 2016, she was conferred the status of Purdue Faculty Scholar for being on an accelerated path toward academic distinction. Dr. Magana serves as associate editor for the *Computer Applications in Engineering Education* journal.

Michael L. Falk is a professor of Materials Science and Engineering with joint appointments in Physics and Astronomy and Mechanical Engineering at Johns Hopkins University (JHU). He received his BA in Physics (1990) and MSE in Computer Science (1991) from JHU and his PhD in Physics from the University of California, Santa Barbara (1998). He was awarded the American Physical Society (APS) Nicholas Metropolis Award for outstanding doctoral thesis in Computational Physics in 2000. His research interests have included topics in non-equilibrium processes in materials including fracture and deformation in amorphous solids, nanoscale friction and wear, transport processes in energy storage materials and atomistic simulation on extended time scales. He also pursues education research related to integration of computation into undergraduate engineering curriculum. In addition to his scientific and education research he serves as P.I. on a NSF funded outreach effort, STEM Achievement in Baltimore Elementary Schools that fields more than 200 volunteers and reaches over 1000 students. He also served as chair of the APS Committee on LGBT Issues and is a proponent for a supportive professional climate for LGBT scientists and engineers.



Camilo Vieira is a postdoctoral researcher for the ROCkETEd group at Purdue University, where he works with computational science education, learning analytics and data visualization. He holds a PhD in Computer and Information Technology with specialization in Computational Science and Engineering at Purdue University, West Lafayette. He completed his undergraduate and his master's studies at Universidad Eafit, in Medellin Colombia. He holds a bachelor's degree in Systems Engineering and a master's degree in Engineering. His research focuses on integrating computational science and engineering within the engineering curricula, as well as using computational tools and methods to support educational research and educational practices.

Michael J. Reese Jr. is the associate dean and director of the Johns Hopkins Center for Educational Resources, the instructional innovation group on the undergraduate campus. In this role, he provides strategic direction for the center, leads educational research, and explores innovative uses of technology in support of teaching with his faculty colleagues. Before coming to JHU, Dr. Reese worked as an instructional designer at Caliber Learning and Booz-Allen and Hamilton. He also worked with the U of MD School of Nursing to launch their distance education program. He received his PhD in Sociology from Johns Hopkins University. His dissertation (Changing Course: The Influence of Social Position and Social Networks on College Faculty's Adoption of Educational Innovations) explored how three educational innovations diffused through colleges and universities in the United States. Dr. Reese also holds an M.Ed in Instructional Technology from the University of Virginia and a BS in Electrical Engineering from Virginia Tech.

Oluwatosin Alabi is a PhD candidate at Purdue University whose research is focused on understanding the role of computational science in supporting undergraduate engineering education. Her interest are related to investigating the use and development of computational tools in supporting problem solving activities in information technology and STEM education. She holds a master's degree in Computer and Information Technology from Purdue University and bachelor's degree in Electrical Engineering from The City College of New York.

Sylvain Patinet is a CNRS researcher working at ESPCI-Paris. His research lies at the interface between physics and mechanics. Using a multiscale approach, they aim to physically justify constitutive mechanical laws based on the modeling of microscopic phenomena via different simulation methods. Its activities concern the study of dissipative phenomena, such as plasticity and cracking, occurring within disordered materials during their mechanical loading. During his career his research focused on two main themes: 1) the study of plasticity at the atomic scale of crystalline alloys (PhD at the French Alternative Energies and Atomic Energy Commission) and of metallic glasses (postdoc at Johns Hopkins University); 2) the determination of the propagation threshold of an elastic line in a disordered environment (postdoct at ESPCI-Paris).