Contents lists available at ScienceDirect

Computers and Education Open

journal homepage: www.sciencedirect.com/journal/computers-and-education-open

Reflective Writing Supports Metacognition and Self-regulation in Graduate Computational Science and Engineering

Jill Zarestky [a,1,*], Michelle Bigler [a,2], Mollie Brazile [a,3], Tobin Lopes [a,4], Wolfgang Bangerth [b,5]

[a] School of Education, Colorado State University, 1588 Campus Delivery, Fort Collins, CO 80523, USA
[b] Department of Mathematics, Department of Geosciences, Colorado State University, 1874 Campus Delivery, Fort Collins, CO 80523, USA

ARTICLE INFO

Keywords:
21st century skills
Computational engineering and sciences
Graduate education
Flipped class
Reflective writing

ABSTRACT

Computational Science and Engineering (CS&E) is a relatively new discipline for which no consensus exists on how classes should best be taught. Many CS&E courses are compressed add-ons to existing programs and, consequently, must cover a breadth of topics encompassing modules from mathematics, statistics, computer science, and application disciplines. Additionally, these courses would benefit from embedded 21st century skills, including problem-solving, critical thinking, and lifelong learning behaviors, but such skills are often neglected in course design even though the education and workforce literature are clear about their importance for future careers. The breadth and complexity in course design necessary to incorporate all of these components create a challenge for instructors and students. In this interpretive study, we investigate student experiences and perceived outcomes in a technology-mediated graduate-level CS&E course designed to address the difficulties associated with this wide range of disciplinary topics and professional skills. Our course design is based on reflective practice and principles of metacognition, and applies elements such as a flipped classroom, student journals, and reflective writing exercises; these design goals directly support students' metacognition and foster self-regulated learning behaviors that, in turn, develop critical thinking and problem-solving abilities. We evaluate this design using student reflective writing and surveys. Results indicate reflective writing activities in course design helped develop students' metacognitive awareness, self-regulated learning behaviors, problem-solving, and critical thinking skills. This course design can serve as a template for others teaching technology-mediated courses in CS&E and related areas, and aiming to develop students' 21st century professional skills.

1. Introduction

1.1. Background and Motivation

Computational Science and Engineering (CS&E) is a relatively new discipline that is often described as the basis for the "third leg" of scientific inquiry – namely, computational simulation –, augmenting traditional experimentation and theory. It combines computational, mathematical, and statistical skills with concrete applications from both traditional science and engineering fields [64], but more recently also questions from the social sciences (e.g., for large-scale data mining) and the humanities (e.g., in "digital humanties" areas). Despite roots going back to the advent of computing in the 1940s, CS&E has only been recognized as a discipline in its own right since the late 1990s. This is reflected in the fact that there are only a handful of dedicated CS&E departments, and that CS&E is overwhelmingly taught as compressed add-on courses in existing undergraduate and graduate programs that have to cover a broad range of topics in relatively little time: courses must often contain modules on numerical methods, statistics, computer science, and application knowledge. Indeed, many departments teach

* Corresponding author
E-mail addresses: jill.zarestky@colostate.edu (J. Zarestky), michelle.bigler@colostate.edu (M. Bigler), mjbraz@rams.colostate.edu (M. Brazile), tobin.lopes@colostate.edu (T. Lopes), bangerth@colostate.edu (W. Bangerth).
[1] [orcid=0000-0003-1728-1796]
[2] [orcid=0000-0001-6300-8761]
[3] [orcid=0000-0001-9402-6903]
[4] [orcid=0000-0002-8428-9934]
[5] [orcid=0000-0003-2311-9402]

https://doi.org/10.1016/j.caeo.2022.100085
Received 30 June 2021; Received in revised form 18 February 2022; Accepted 3 April 2022

such courses in the form of "Computational X" where "X" may stand for physics, chemistry, materials sciences, or any number of other areas. At the same time, computational skills are widely recognized as very important for today's work force in the sciences and engineering, and questions how to teach these skills are therefore quite relevant.

There are a number of educational challenges that come with teaching CS&E and that we will consider as the backdrop for our work herein:

- CS&E is still a new and evolving area, without a set curriculum based on a long history. Rather, the selection of topics is often left to a small group of instructors and frequently revised. As a consequence, there is little research into effective educational approaches instructors can draw on for their course designs.
- The compressed schedule on which most CS&E courses or course sequences are taught implies that students cannot be given a comprehensive overview of the field. Rather, curricula have to focus on a few specific areas that are representative of what practitioners might require.
- The fluidity of the field means that the focus of CS&E courses should be on *concepts* and *ways of thinking*, rather than on facts. This is not dissimilar to many other computer-related fields where computer languages and computing frameworks may come and go, whereas computational thinking, programming patterns, and abstractions remain relevant.

These sorts of considerations lead us to believe that in CS&E education – like for many other new and emerging areas without a long-established curriculum and multi-semester course sequences –, a focus on concepts, reasoning, the ability to continue learning, critical thinking, problem solving, reflection on one's learning progress and process, and adaptability is important, and maybe more important than fact-based knowledge. *How* to teach these skills – often summarized as "21st century skills" – is, however, largely unexplored in CS&E education despite prominent calls for such work in reports by the National Academies [57] and in the premier professional journals (see, for example, [64,69]). Indeed, the most comprehensive recent report on CS&E education [64], citing reports by the National Academies, the Department of Energy, Interagency Working Groups, worldwide coordinating bodies, and others, focuses almost exclusively on the many topics students need to learn, but has little to say on how we can achieve this. In contrast, the National Academies' report [57, chapter 6] forcefully calls for putting *skills* rather than facts at the forefront of curricula, and for developing technology-enhanced course designs backed by educational evidence; yet, it too has little to say about concrete course design ideas.

Taken together, the aforementioned issues create a teaching challenge that is, as yet, unresolved. Given the difficulty of balancing the CS&E content-based skills with the professional and learning skills, we have applied a research-based approach to innovation in technology-mediated instructional practice.

1.2. Context for Teaching 21st Century Skills in CS&E

As in most STEM fields, CS&E jobs are largely in research and development, and instructional strategies and practices need to adequately prepare students for this work environment [32,33]. [34] conducted a systematic literature review to investigate the question "What makes a software engineer stand out in his or her profession?" Their results identified communication, teamwork, self-reflection, conflict resolution, and mentoring as the five most important skills to incorporate into the curriculum. Interestingly, all of these are non-technical skills, rather than specific technical abilities or knowledge, and embedding these into computing domains is seen as facilitating the development of a holistic set of skills that cultivate the lifelong learning processes required to keep pace in an evolving industry [61].

As outlined above, the challenge for education in Computational Science and Engineering is the need to focus on *skills* in addition to facts – including the skills necessary for becoming a life-long learner –, but that existing research provides little guidance as to how this can be achieved in CS&E instructional practice [40]. Knowing that critical thinking and problem solving skills are essential for the students' future careers within the industries our students tend to work in, we aimed to maximize opportunities for students to practice and develop these skills during the CS&E course we will describe below. A key piece of our work is a focus on *reflection*, in line with [56] who stated that "good problem-solvers are reflective: they reflect all and only when needed. And their reflectiveness is one of the skills that underwrites their understanding" (p. 80). Indeed, reflection can be used to extend problem solving [1,3] and enable metacognitive processes.

1.3. Overview

In this contribution, we heed the calls from the National Academies [57] and others (e.g., [64,69]) and report on a graduate course design and its practical evaluation for a CS&E course that we taught at both Texas A&M University and Colorado State University over the past decade. The centerpiece of this course is based on the *use of digital educational technology*, and includes a flipped classroom, semester-long individualized projects, and students tracking and reflecting on their learning in a digital journal. The purpose of this manuscript is to present an investigation of a graduate-level computational science course designed to *foster students' reflective practices to improve critical thinking and problem-solving*.

Therefore, in the following, let us first review the conceptual and research contexts in which our work is situated, where we focused our course design on developing *metacognition* and *self-regulated learning behaviors* as the key enablers of developing the skills necessary towards becoming a CS&E professional through the use of a flipped classroom and reflective writing in STEM education.

Having so set the stage, the remainder of the paper is structured as follows: We begin in Section 2 with a systematic review of the theoretical underpinning of our work, followed by posing the purpose and the specific research questions of this study in Section 3. In Section 4 we will then provide a description of the study methods and an overview of the design and disciplinary content of the course – the practical implementation that guided the research. We then present results in Section 5, and provide discussion of these results as well as conclusions in Section 6.

2. Conceptual Framework and Literature Review

This study is grounded in the social cognitive perspective of Bandura [9], which provides a comprehensive framework to explain human behavior by examining the interplay between personal, environmental, and behavioral factors. Our study utilized the social cognitive perspective to examine the reciprocity of students' self-regulation of critical thinking and problem-solving skills (personal factors), a flipped classroom model (environment), and metacognition captured in reflective journals (behavior). In the following sections, we address each of these factors. We will in later sections explain how these concepts then inform both our course design as well as our research design.

2.1. Reflective Practice

Schön [66,67] described professionals as "reflective practitioners" who, over time, improve their abilities to navigate situations they encounter in their discipline. Schön's work and related literature on reflective practice (e.g. Harvey et al. [35], Johns [38]) emphasize that *expertise development is grounded in a combined awareness and scrutiny of one's learning processes* and accumulation of relevant experiences. From this awareness and scrutiny, a person can learn lessons that build professional skills and intuition. In turn, the deep expertise we wish to instill in our CS&E students develops from experience grounded in *reflection-in-action* and *reflection-on-action* [66,67].

Reflection-in-action "reshapes what we're doing while we're doing it" [67, p. 26] and is a process that occurs concurrently with action. It is often equated with thinking on one's feet, key for decision-making in real time, particularly in interaction with others, such as a project team. Reflection-on-action occurs when a person has an experience, like finding a bug in code, and later reflects to identify important takeaways; in the context of CS&E, these takeaways could be strategic design decisions for programming, specific uses of debugging tools, or understanding the importance of defensive programming practices. On the whole, reflective practice enables current and emerging experts to build their professional skill sets based on concrete experiences. The compilation of in-the-moment reflections increases over time; subsequent conscious reflection on those moments grows and develops a person's understanding of how such moments occur, how they may go well or badly, and how such moments might be handled moving forward.

Reflective practice is a concept likely to be familiar to most practitioners. Indeed, most CS&E professionals will likely agree that they have become better at programming by observing their own programming and debugging practices, and consciously teasing apart what worked and what did not.

2.2. Metacognition and Self-regulation

Metacognition, defined as someone's "awareness and understanding of their own thinking and learning processes, as well as their regulation of those processes to enhance their learning and memory" [58, p. 363], combines and extends the two types of reflection described in the previous section. Metacognitive functions include learners' assessment and beliefs of their own abilities, monitoring their state of knowledge, understanding their cognition and thought process, and controlling learning activities; it also involves regulating aspects of cognitive enterprise [2,29,49]. In our context, for CS&E professionals, metacognition synthesizes the multiple content- and process-based tasks as well as the interpersonal and self-regulating components of work. People learn from direct experiences (reflection-in-action), subsequent analysis of those experiences (reflection-on-action), and endeavor to achieve learning goals and seek growth opportunities (metacognition). Over time, practical experiences and related reflection build and combine into increased skill and expertise. In sum, reflective practice grows from combined technical or content knowledge, accumulated real-life experiences, deliberate learning from prior experiences, and purposeful professional development and is tied directly to learners' understanding and regulation of their learning processes [63]. Further, metacognition is an underlying function of critical thinking that enables learners to develop judgement and decision-making [29].

Self-regulation is situated in social-cognitive theory [9,68,77] and focuses on the metacognition associated with moving toward goal completion [48]. Self-regulated learning can be summarized as thoughts, behaviors, and actions that have been intentionally adapted by an individual to accomplish a specific goal; this implies that an individual engages in metacognitive awareness that elicits behavioral adjustments to attain and implement knowledge more effectively [18]. Self-regulation combines motivational beliefs, cognitive strategy, and metacognitive control and is a critical skill set for life-long learning and professional skills development [11,18,76,77]. Cognitive strategies involved in self-regulated learning take the form of simple problem-solving and critical thinking [68], and may include activities such as planning, process monitoring, comprehension monitoring, reflection on cognition, and self-explanation [48].

Metacognition and self-regulated learning are inextricably linked to critical thinking, problem-solving, and other human-dimension skills (e.g., comprehension, memory, oral communication, and language acquisition), see Flavell [31], Loksa and Ko [48], Lumpkin [49]. Walker and Finney [75] described life-long learning as extending beyond the simplicity of accumulating further knowledge, but as a *way of being* that engages critical cognitive faculties to continue holistic development. Such practices support the life-long learning skills necessary for sustained professional success [2,29], considering continuous technological change. Indeed, this seems particularly pertinent to CS&E, where, for example, the computers typically used by practitioners have grown from single-processor ones to clusters with more than a million cores, requiring entirely different programming models.

2.3. Writing to Learn

Having described important skills in the past two sub-sections, the question is how students can be prompted to engage in reflection and metacognition. Clearly, as part of the meaning-making process associated with experience, learners need to reflect, consciously or unconsciously, on their experiences and incorporate new learning into their working knowledge base. [42] described a cycle of experiential learning and formalized reflection as part of the learning process, as experiences are interpreted through a process of reflection. *Reflective writing* is one means of developing skills to support students' metacognition and self-regulation of learning. We discuss below how we incorporate reflective writing into our course design by using an electronic journal.

In a broader sense, *writing to learn* is a recognized instructional strategy for deepening students' content learning. In university settings, such as the present study, [78] founded the writing-to-learn movement, and advocated for writing across all disciplines to deepen student engagement with and understanding of content; many universities now require that all undergraduate degree programs have writing-intensive courses. He described reflective writing as an active process through which students may organize and clarify their thinking. Similarly, [12] viewed written reflection as an opportunity for students to share their feelings and thought processes which can result in deeper learning and better connection to content. Such writing may include restating concepts in their own words, describing strategies or approaches to problem-solving, or developing personalized mechanisms to facilitate internalization. More concretely, in science and mathematics education, *writing to learn* has become a widely accepted means to deepen content learning and improve scientific inquiry skills [5,19,27,30,36,39,74].

Reflective writing has been widely studied regarding its impact on learning in the mathematical sciences (e.g., [19,62]). For example, [71] advocate for opportunities for reflection on mathematical processes and group work. In effect, reflective writing enables students to uncover what is known, make connections, ask questions, and recognize areas for growth [12], all important metacognitive tasks for developing the self-regulation skills essential for industry careers within CS&E. *Reflective journaling*, in particular, is a powerful means for fostering metacognition and developing self-regulated learning [2,47], which subsequently supports the development of life-long skills such as critical thinking.

2.4. Reflection in Graduate CS&E Education

Drawing from reflection's history as an educational practice that supports metacognition and meaning-making from experiences [26] but specific to the computing domain, [34] stated, "the better the student's ability to reflect, the better the ability to absorb other skills... [reflection] can be seen as the main enabling skill that increases the likelihood of learning anything else" (p. 5). [33] indicated that the practice of reflecting differs between disciplines as the process is dependent on the experiences and skills accessed; within applied sciences such as CS&E, these practices includes planning, procedures, processes, and problem solving [17,32,33].

While the use of reflective writing has been investigated in undergraduate settings [6,46,59,70], few studies have focused on these strategies for developing skills at the graduate level in CS&E [32]. [4]

redesigned a graduate level programming lab to focus on developing "soft skills" in addition to technical knowledge by incorporating active learning and learning-by-teaching practices. However, minimal reflection was incorporated in this study as teams were only asked to complete a one-page after-action report. Therefore, the level of reflection studied was inconsequential in comparison to the analysis of active learning and learning-by-teaching strategies. As a result, a gap in the literature exists regarding design and implementation of graduate courses within the computing domain that utilize reflective writing to develop 21st-century skills in conjunction with technical skills.

The importance of evolving the design of graduate-level STEM courses to include non-technical skills (such as problem solving and reflection) originates within the gap between industry needs and graduate skills. This gap has been noted in many reports, including the one by the National Academies mentioned above [57]. Specific to PhD programs, [60] stated that "the complex nature of 21st-century challenges requires PhD holders to not only be specialized and independent, but also open-minded and critically reflective" (p. 71). While we are not alone as we attempt to design STEM course structures that incorporate the development of a broader range of skills for graduate students, many curriculum reforms are mainly focused on increased content knowledge as opposed to developing the students as lifelong learners. Among the outliers, the teaching methodology used by [16] for postgraduate students incorporated an interdisciplinary approach in order to provide "average science graduates" with both the mathematical and computational tools to solve a variety of parallel programming and performance engineering problems. Throughout their courses, the "leading role" is balanced between the teacher and students. Initially, the teacher leads the content and process during lecture and lab sessions. Then, students generate concept maps that illustrate the hierarchy of the concepts taught using defined strategies and information structures, use case studies to analyze well-known practical problems, and develop experimental portfolios that provide descriptions of work tasks that help students auto-evaluate their content knowledge.

Similarly, [32] incorporated real problem scenarios, reflective journals, and iterative instructor feedback within their graduate level engineering course to help students develop "five non-technical, career sustaining and career development competencies" (p. 309) needed to meet 21st-century challenges: self-reflection and articulation of knowledge, speculating and identifying gaps, asking questions to investigate gaps, making decisions with incomplete information, and identifying new ways to move forward. [4] chose to focus on increasing student motivation and keeping the content updated when redesigning their graduate level computer science course to develop soft skills as well as technical knowledge. [24] utilized peer reviews of reflective writing to help their undergraduate computer science students to further develop their writing skills. [15,22], and [44] utilized a project-based learning model for their undergraduate courses to help students develop responsibility, independence, and discipline as well as creativity and problem solving while engaging in open ended, real world problems.

Each of these course revisions was intended to broaden students' skill set, increase their knowledge base, and better prepare students for future career opportunities. While this is clearly important, our focus in designing a CS&E course was to use technology-mediated reflective processes as a tool for students to *develop their metacognitive practice and self-regulation skills*, addressing the gap in the literature identified above.

2.5. Flipped Classrooms

The course design we will describe below is based on the flipped classroom model. Flipped format classes are by now a well-established course design that requires students to engage with course materials *before* attending class. The goal of a flipped format is to free time in class for more active student engagement and increased interaction. A number of other articles serve as excellent introductions to the flipped format [10,50,52], but, to summarize, the overarching assumption is that class time is best spent on activities such as faculty interaction with students. Content delivery can be accomplished through technical means, such as videos or reading assignments. In addition to better support for students with diverse learning needs [43], education research has also shown objectively that increased classroom interactivity improves student learning outcomes [25].

The flipped course structure has gained popularity in both K-12 and higher education based on the growing ease of making videos, the availability of quality content-based videos on sites such as the [20,28,41,54,73], and resources provided by publishers. Individual instructors have also developed extensive collections of high-quality video lectures [46]. Within the computational sciences, let us mention just two examples: Tim Davis's 42 lectures on direct methods for sparse linear systems [21], and Maggie Myers and Robert van de Geijn's edX course on foundations of linear algebra [55]; the videos for our course also fall into this category. Finally, an extensive collection of teaching materials is available from [72]. Given the high-quality material readily available for free on essentially every topic imaginable, teaching a flipped-format class is no longer prohibitive in terms of the up-front instructor effort of recording lectures. However, further research is needed to determine how student approaches to learning shift in a flipped classroom model [23].

3. Study Purpose and Research Questions

The centerpiece of this course is the *use of digital educational technology*, and includes a flipped classroom, semester-long individualized projects, and students tracking and reflecting on their learning in a digital journal. The purpose of this study was to investigate a graduate-level computational science course designed to *foster students' metacognitive and self-regulation to improve critical thinking and problem-solving*, as guided by two research questions:

(RQ1) What did students' reflections reveal about their critical-thinking and problem-solving skills and processes during the class?

(RQ2) What were students' perceptions of the experience after the class had concluded?

4. Methods

As we sought to uncover how students' reflective writing illuminated their thinking and learning processes and their perceptions of the course, an interpretive approach was most appropriate [53]. Interpretative research values research participants' perspectives and seeks to uncover the way they view their circumstances and make meaning from their experiences [53]. Therefore, we applied a qualitative research design, with students' writing comprising the majority of the data to examine their experiences in the class (RQ1). Capturing students' perceptions of the class after its conclusion (RQ2) allowed for a longer-term perspective on students' experiences. Next, we present details on the course and its design elements, the students who participated in this study, and our data collection and analysis processes.

4.1. Course and Disciplinary Context

The specific context in which our work is located is a CS&E course on "Finite Element Methods in Scientific Computing". The finite element method is the most widely used method for the simulation of both fluid dynamics and solid mechanics applications, and most student projects come from these areas. Such a course must touch on many of the topics listed in the recent review of CS&E curricula by [64], including the following subset excerpted from that review:

1. Foundations in mathematics, ordinary and partial differential equations;

2. Simulation and modeling, use of simulation tools, and assessment of computational models.
3. Computational methods and numerical analysis, including errors, solutions of systems of linear and nonlinear equations, and numerical methods for PDEs.
4. Computing skills, including compiled high-level languages, algorithms (numerical and nonnumerical), elementary data structures, analysis of algorithms and their implementation, parallel programming, scientific visualization, awareness of computational complexity and cost, and use of good software engineering practices including version control.

In addition, course objectives focus on students' need to understand the scientific or engineering background of the application on which they choose to work. The origin of the course design was how one can structure a course that has to cover such a breadth of topics in just one semester. More specifically, we asked ourselves the following:

What kind of course design is most appropriate to teach a broad collection of mathematical, computational, and learning skills? And can we find evidence that a given design indeed works?

4.2. Course Design and Overarching Approach

In our efforts to address the aims previously described, we designed the course to incorporate three approaches not typical of graduate mathematics and engineering education: a flipped-classroom format, research journals, and reflective writing. These choices are based on the considerations outlined in Section 2, namely that we wanted to design the course to strengthen students' reflective practices (see Section 2.1) and metacognitive abilities (see Section 2.2), by using writing-to-learn strategies (see Section 2.3). We discuss each of the three components mentioned above in the following subsections, along with a short overview of a typical class period. An early version of this design was previously described in [7].

While the course has a concrete list of topics students are supposed to learn over the course of the semester, we see these not as isolated components, but instead as building blocks for each student's semester-long, individualized project. In this course, the projects are typically a finite element solver for a problem of the student's choice, and based on the widely used deal.II software library [8]. Project-based designs are common and effective in courses where the students' ability to *apply* their learning is paramount, as is the case for the graduate students in this course [13]. Additionally, the direct relevance of course projects to students' research or interests is an important factor in fostering intrinsic motivation [65] and helps in building critical thinking skills, reflective thought, and metacognition – in line with our course goals, see also Section 2.

At the beginning of the semester, students propose a semester project in a 5-10 minute presentation. Their proposals typically draw from their graduate research or general interests. Over the next weeks, the project is refined in collaboration with the instructor, who also ensures all projects are of roughly the same difficulty level. At the undergraduate level, one might provide students with a list of possible project topics or directions.

To some degree, the topics listed in Section 4.1 can all be seen as prerequisites for completing these projects. The first half of the semester is spent working through foundational material common to all projects. In the current course, such material begins with the basics of finite element methods and programming essentials. The course then gradually transitions towards necessary background material that students can learn in parallel to working on their projects – e.g., visualization techniques, parallel computing strategies, or the use of version control. The second half of the semester may require students to study project-specific topics using material provided by the instructor, available in textbooks or the research literature, or online. For the purposes of this course, such content might include knowledge of specific spatial or temporal discretization techniques, or particular linear and nonlinear solvers and preconditioners.

At approximately midterm, students give ten-minute presentations about their project progress. This presentation provides students with a timeline for switching from learning background material to self-directed work. The semester concludes with twenty-minute presentations from all students on their project results. The course is graded based on these presentations and other materials students need to hand in, such as the commented source codes for their project.

4.2.1. Flipped Classroom

Given the project-focused nature of the course, we felt student-instructor interaction needed to be the central design component. In previous incarnations of the course, we recognized that even spending half of a class period lecturing resulted in too little time for necessary one-on-one interactions with students to talk about their projects.

For this course, we therefore recorded 67, professionally produced lectures at the public television studio at one of our universities. The videos alternate between a view of the instructor for longer phases of verbal explanation, and the instructor's screen for pre-written slides and interactive demonstrations of tasks such as visualizing data, programming, debugging, or using the command line. The videos are hosted on YouTube. A major benefit is that students can stop the video, perform the same steps on their own data sets, and then continue watching at their own pace, repeatedly if necessary [46]. An additional advantage of this format is that one can provide material for students who lack some background knowledge or simply want to work at their own pace, without slowing down the more experienced students. Subtitles and rewinding also supports students who have difficulty with English language issues.

The videos are available through the *Youtube* user interface, but are also linked to at a central page hosted at https://www.math.colostate.edu/~bangerth/videos. They have found widespread use also outside the course described herein, receiving a combined total of approximately 200,000 views since 2013. Given that *YouTube* is not available in China, the central page above now also links to copies of the videos hosted on the Chinese *bilibili* service.

4.2.2. Learning Journals

It is a challenge to watch a 30- or 45-minute video lecture without getting distracted. Students are tempted to let them run in the background without really paying attention - or simply not watch them at all, and then ask for help during the in-class portion of the course. As a consequence, video lectures by themselves are well understood to not be very effective teaching tools. On the other hand, with careful teaching structures built around and supporting these videos, they can create a much more efficient environment for learning. Many of these techniques are discussed in the references we have provided in Section 2.5.

Specifically, we required students to keep a journal. Our original goal was for students to use these journals to document and digest what videos they had watched, and to share these notes with the instructor. Consequently, we required journals to contain (i) a table of contents; (ii) a record of the lectures they watched, along with a summary of each lecture with their three most important insights or observations, and two or three questions they still had; (iii) a log that showed when students worked on which parts of their projects, progress (or lack thereof, both often illustrated by copy-pasting formulas, error messages, or visualizations of results), and notes for what they want to try next. In some sense, this structure reflects lab books used by experimental scientists. We discuss below how our experience with student journals changed over time regarding what we hoped they write.

We used Google Documents as the platform for these journals as almost everyone is already familiar with this interface. The platform allows simultaneous editing, enabling student and instructor to have conversations. It accommodates text, images, formulas, code snippets, and most other pieces of information students may want to share or instructors may want to reply with. As an instructor, one can see the documents shared by students in a single place, and the view can be configured to highlight documents that have changed since the instructor last reviewed journals.

The initial goal with the journals was to judge who was watching the videos as assigned and address students' common questions or points of confusion. They were also intended to synthesize and resolve common misconceptions about the material in the video lectures – an important point given that video lectures were recorded without an audience and consequently without immediate feedback.

4.2.3. Metareflection

In addition to the reflective writing in the regular journal entries, we incorporated targeted metareflection by requiring students to write two essays. The prompts for these essays are as follows:

> *You will need to submit an essay, about 1-2 pages, that summarizes an important insight you have had regarding this class and how you arrived at that insight. Examples might include understanding a new technique to debug programs; discovering another reason why version control systems are useful; etc. To inform these essays, read back over your journal and look for patterns, a-ha moments, or anything that stands out to you as particularly important. I encourage you to reference specific journal entries as part of your essay.*

Education research has found substantial benefits from prompting students to *reflect about what they have learned and how they have learned*, rather than just requiring them to demonstrate learning [14,59]. In other words, students gain deeper insights into their learning by reflecting on what they did right or wrong, and why, with a periodic and intentional focus on the big picture of their learning.

4.2.4. Class Meetings

The flipped class format created opportunities for one-on-one or small group interactions, as intended. Once processes had become clear to students, we generally spent the first 10–15 minutes in discussion, oftentimes to address open questions found by reviewing student journals and isolating common misconceptions. On other occasions, we also discussed particularly good solutions to problems or specific insights isolated from student journals. By asking students in their journals whether they were willing to bring these issues up themselves, many of these discussions were student-led.

During the remainder of the class, students worked on assignments or projects, with the instructor engaged in interactions answering questions and helping with project work. Such personal interaction required awareness of how much time is spent with each student. We also rotated the order in which we talked to students between class periods. Augmented by interactions in journals, and the fact that neighboring students often participated in discussions, we felt most students received sufficient individual attention to make progress on their projects. In practice, few students come to additional office hours for additional help – offsetting some of the instructor time spent on reading through journals.

4.3. Participants

We used the course format described above for classes in 2013, 2015, and 2018. While the course was the same in all three iterations, it was conducted at two different universities due to the instructor changing institutions. Both universities are large U.S. public land-grant institutions, one in the Southern region and one in the Mountain West region, and both are classified as "very high research activity" by the Carnegie classification system [37]. All students enrolled at any time were eligible to participate in this study, for a total of 41 students across the three classes; 39 (95%) consented to participate. Students' informed consent was obtained through protocols approved by both universities' Institutional Review Boards. Names of participating students were withheld from the instructor, this study's last author, until after final course grades were submitted. All participants were graduate students at the time of the study, enrolled in doctoral programs in Mathematics (43.59%), Petroleum Engineering (12.82%), Mechanical Engineering (10.26%), Nuclear Engineering (7.69%), Electrical Engineering, Geology, Geophysics, Atmospheric Sciences, Physics, Aerospace Engineering, and Civil Engineering (2.56% each).

Participants predominantly identified as men (87.18%). Students came from a variety of countries, including the U.S. (23.08%); China (48.72%); India, Korea, and Taiwan (5.13% each); South East Asia, Norway, Saudi Arabia, Mexico, and Iran (2.56% each). When referring to students below, we use pseudonyms that aim to reflect students' gender and national origin.

4.4. Data Collection and Analysis

The student journals comprise the majority of the qualitative data for this study. Over the three instances of the course, we have accumulated 1,441 pages of student journals; we downloaded these journals from Google Docs after each semester ended, for each student who consented to participate. Journals on average were 40 pages per student by the end of the semester, reaching a maximum of 126 pages for one particularly diligent and verbose student.

The journals were then uploaded into nVivo qualitative data analysis software, which facilitated a thematic analysis process [53]. One of the authors unitized the journal documents, i.e., divided the text into units, each of which possesses independent meaning. We used a multicoder approach to support the confirmability and dependability of the analysis [45]. Independently, one researcher used open coding, which allowed themes to emerge through a constant comparative method of assigning an existing theme or a new theme to new data as appropriate [53], as opposed to *a priori* themes. Given this study's exploratory purpose to uncover student problem-solving and critical thinking processes, open coding enabled reliance on the data rather than preconceptions of what students would or ought to write about. Coding was independent of course-specific content or instructor comments or feedback. Then, using the lens of the research questions, a researcher applied axial coding to group initial codes into shared themes or categories. Independently, a second researcher reviewed the coded data; researchers then met to reconcile their analyses. Initially, the team matched on approximately 85% of initial codes and, through dialog, reached consensus.

For the longer-term reflective component of this study, we asked all participants to complete a survey designed to investigate perceptions of the course after one or more years had passed. Of the 39 participants we contacted, 14 (36%) completed the survey. The survey provided students an opportunity to share how they had (or had not) benefited from course content, format, and processes in the longer term. Survey questions asked participants to compare the course to traditional format courses, reflect on the journal-writing experience, and provide recommendations to the instructor and future students. Examples of the Likert-scale questions include the following:

- I preferred the format of [this] class over lecture-based formats
- I took advantage of the opportunity to ask [the instructor] questions via the journal.
- Journal entries helped me to get feedback to guide my learning process.
- I learned more in lecture-based courses than [this] class.
- I had enough opportunity to ask [the instructor] questions in class.
- The video lectures distracted from my learning.
- I had more interaction with [the instructor] than other instructors.
- Journal feedback from [the instructor] helped me to improve my project code.

To analyze survey data, we calculated frequency distributions for all fixed-response (Likert) questions. For the open-ended questions, we grouped responses thematically in a process mirroring that of the journal data analyses. In the following, we present details of the findings from both analyses.

5. Results

As discussed in the previous section, the journals were rather unstructured and encompassed a wide variety of content primarily determined by the students. Students mainly utilized the journals for documenting procedures (31 students), taking notes (39 students), describing their strategy/process for coding (35 students), or sharing visuals from their coding strategy or results (28 students). Similarly, 30 students described the debugging process for their projects. Interestingly, among the largest categories are journal entries in which students posed questions to the professor about content from the video lectures, checks for their understanding, and guidance for next steps in their individual projects. Many of these cases precipitated student-instructor conversations via the journal – in other words, the journals allowed for precisely what makes a *teacher* more valuable to a student than a book or a video lecture. These interactions also allowed the instructor to focus on answering specific questions from the journals during class meetings as needed. In general, students acknowledged the value of self-determined learning strategies that helped develop professional skills for content learning, software development processes, and professional and personal development, presented in the following sections.

5.1. Content Learning and Software Development Processes

As part of the research journals, students recorded and reflected on their content learning, which comprised the largest volume of text. Students identified what they were learning and sometimes how they were learning, including material related to course topics and tools and processes they found exciting or valuable. These entries also served as formative self-assessment and checks for the instructor to determine if students were mastering the course objectives.

In their journals, students demonstrated that they developed a new or deepened awareness of software development practices. For example, they wrote about the benefits of developing a design plan, adding comments in code, breaking their programs into smaller tasks, and reusing previously written code. The journals supported 28 students in identifying areas of growth and posing ideas on how they would foster continued learning, such as recognizing the need for further depth with programming (C++, Linux, deal.II), developing good habits for processes (version control, including good documentation using tools such as Doxyen, checking units and kinds of boundary conditions when implementing formulations), or developing more background skills in writing and mathematics. Thirty-one students stated they would further pursue content knowledge through websites, asking friends, completing future classes, reading books, and continued practice.

A substantial part of the course is students developing software for individualized projects but, as Matthew acknowledged, "serious code development is a team effort." Most students (77%) perceived benefits from working collaboratively with classmates, office mates, advisors, friends, supervisors, and professors. Connor noted that tapping into the experience of those who have good programming habits, "enlarged my view of what can be done and of how to do it simply". Not only were collaborators sharing valuable programming experience, they inspired each other to look at projects from different angles, and provided encouragement for each other when they found themselves stuck. At times, students received the benefit of others' expertise in the form of a hint or recommendation for a resource. In other situations, students provided support to each other as seen in Abbud's description: "Over the next few weeks, [other student] and I touched base on OOP design issues with the code. I could see in [other student's] code that he is thinking 'procedurally'. He extracts all the information from all sources into the current procedure to produce the result they need. I then explain the idea behind OOP and the importance of 'encapsulation'. I can see his eyes light up when the concept sinks in and see first-hand how it simplifies their program tremendously." These insights into software development processes support students' preparation for team-based professional environments.

5.2. Professional and Personal Development

While the majority of the journal entries focused on documentation of newly learned content and processes for software development, there were also 950 journal references that were examples of *metacognitive processes and/or analysis*. These entries are indicative of not only learning how to *do* something, but indeed of developing a deeper understanding of *why* and *when*. Students shared learning about general professional skills, including developing a healthy work/life balance, reducing anxiety levels around coding and debugging and how this has led to increased confidence and courage, the importance of jumping in and experimenting, and learning from and with others. Self-regulated learning behaviors, such as self-discipline, study habits, organization, time-management, and asking questions more frequently and sooner were also skills that students identified as needing to improve on. Metacognition provides the first step for students in the process of further developing these skills, as metacognition is an essential and foundational component of self-regulating. Additionally, 10 students used the journal to identify goals for their project, course or future work and 27 students contemplated the application of their new journaling/reflective writing skills to future endeavors. Whether it was continued work on the project from class, increased confidence in programming abilities or application to future work, students articulated their future uses for the course content and skills they learned in this class.

Similarly, reflective writing enabled students to identify the range of emotions they experienced during the learning process, spanning from excitement to panic, confidence to confusion, irritation to enthusiasm. Expressing strong emotions may have helped students to reframe their work or empowered students to take risks in class. Some reported feelings of being overwhelmed early in the semesters and ended with some level of increased personal and professional confidence. One student expressed a shift in perspective through reflective processing: "I've always been blaming all these [problems] to my lack of knowledge, not enough effort, bad time management and hard project, until just now... The reason that I got frustrated is not the project, the code or the limited time but what is inside. As one of my friend commented 'the set of obstacles is dense in life', and yes, there will be endless problems to solve, but I can and should change the way I look at it" (Nuan).

While emotions are not typically a consideration in graduate-level computational science course-work, emotions do affect student learning. By giving students an outlet for those feelings in their journals, they are better positioned to deal with those feelings in a constructive manner.

5.3. Students' Perceptions of the Course

Student perceptions of the course and its structure were drawn from both the student journals and the survey. They were mostly centered on the course structure in general, individual projects, and the use of journals and class time.

General course structure. 62% of the students (24 of 39) commented at least once in their journals about the course structure. Students described that they utilized the video lectures to explore new content at a comfortable pace; they would pause, replay and re-watch video lectures many times in order to add depth to their understanding. The recorded lectures provided flexibility and enabled students to choose when and where to engage with the content. As Jinhai summarized, "We learn the theory from the lecture video and learn the programming skill from running examples and doing our individual projects."

Overall, students valued the flipped classroom structure. For example, Connor wrote, "I really enjoyed this semester and found myself wishing more classes were structured in a similar pattern. I liked having the lectures available online and then having class where we could practice and get our hands into the material. It fit my learning patterns perfectly. I could see many more classes successfully patterned in this manner." Commenting on the fact that the out-of-class videos came from a trusted source, Xiang wrote, "The importance of this class also lies in the difference between finding something and being told something."

Individualized projects. Many students identified the project and presentations as some of the most valuable learning experiences during this course. Ishan enthusiastically wrote about the practical applicability of his project stating that "by taking this course and completing the project, I feels like I have already completed one-year worth of my research in just one semester." Presenting their project to their peers also provided learning opportunities that students recognized as relevant to life after graduate studies.

Keeping a journal. Reflective writing is a challenging task for many students in STEM disciplines, especially if they have not had previous opportunities to explore their learning processes through introspection. This new experience was met with mixed reactions, as 20 students commented in their journals and reflection essays. An example of this tension was described by Connor, "I find that most of the time after I have watched a video or worked on some aspect of coding pertaining to the project, the last thing I want to do is come over here to the journal and write. Yet I feel at other times a great desire to come and write down some epiphany or victory that I have had. I know it benefits me a lot to write things down and I have had many of my questions answered merely by the act of trying to explain something in writing even without answers from [the instructor] but it is still hard to be consistent."

Most students quickly realized that the more they write in their journals, the more feedback and help they get with their questions and projects. As a consequence, our use of these journals radically deviated from the more rigid structure outlined above to a much more free-form record of what students were working on, what issues they faced with their projects, exchanges of ideas and suggestions to look at problems they face in a different way, as well as a record of much critical thinking about themselves and their learning progress.

Some students acknowledged that they understand that reflection on learning experiences is a good practice, yet also indicated the difficulty in developing that habit. The benefits that students experienced stem mainly from documenting the justifications for the processes they utilized; when rereading this at a later date, students were able to see patterns in the types of mistakes they had made. When reviewing journal entries, students found it helpful to be able to use keyword searches to quickly find the information they were searching for – a benefit of using digital journals.

Journal writing was also viewed as a way to step away from the coding process to write about issues they were experiencing. Students commented that the temporary step away made resuming work on the project a little easier. A notable finding is that shy students or students whose language and cultural differences made asking questions in class difficult, found the journals to be a comfortable and efficient way for posing questions to the professor and were grateful for the opportunity to communicate directly with the professor. Nuan's journal entry illustrates the anxiety involved in asking questions for some students: "Being a Chinese student, it sometimes is really hard to step in professors office and ask a question. It seems there are a lot to be afraid of, what if the question is too stupid, what if I can't follow the professor's explanation, what if he founds out that I'm so behind other students. So in many cases, I'd rather spend tons of time finding answers by myself than ask a 5 minute question. But so far, all professors answers all my stupid questions with greatest patience that I can ever expect. It seems none of the scary things I thought did not actually happen. When struggling with Linux at first, the installation command was really hard to me and I googled it for several hours without a satisfactory results. But it only take a few minutes to ask and get an answer as well as a smile. Now I'm pretty sure that It's absolutely worth doing so. [...] English is another obstacle, but conversely, my English would never be improved if I don't speak it." The journals also met their goals of clarifying video content and points of confusion and misconception well; the instructor was able to recognize who needs to be reminded to watch assigned videos, and we did find common misunderstandings that we addressed by offering additional information and errata for each video.

Ultimately, many of the students shifted their previous thought processes about learning through the use of the journal. While it took a long time for some students to embrace the concept of reflective writing to enhance learning, student's writing indicated they understood the importance of the process on their learning. Hanjae disclosed that "even though entering journals took a long time for me, I have learned that this tool can help me learn more effectively. I would prefer this kind of journal entries for my future courses."

5.4. Survey results

When asked to recall their experience between one and six years later, 14 students voluntarily completed an online survey. Fig. 1 presents key results. All students who responded to the survey preferred or strongly preferred the flipped-class format over traditional lecture-based formats. The majority of respondents indicated that they took advantage of the opportunity to ask questions via the journal, used the journal to get feedback that guided the learning process, and felt feedback received in the journal helped improve their project. Most respondents also indicated they had enough opportunity to ask questions in class and had more interaction with this instructor than with other instructors. Conversely, most respondents strongly disagreed or disagreed that video lectures distracted from learning, and had a largely neutral opinion about learning more or less compared to traditional, lecture-based course.

When asked what the respondents saw as the purpose of the journal they were required to keep for class, Zach indicated that "one of the most important things is I saw it as a way to keep myself accountable. I just started a personal journal on April 4 [2019] with this idea of accountability in mind and it has helped very much. Our class journal was also a nice way to communicate to [the professor] about things that I decided weren't urgent enough to pester him via email for." Like this respondent, 71% of the former students indicated they have continued to use a journal for research (29%), personal (21%) or both research and personal (21%). Other respondents also shared similar perceptions about accountability, and the journal as an interface for communication with the instructor. 57% of respondents indicated they preferred asking the instructor questions in person in class, 21% preferred asking questions in person during office hours, and 21% of the respondents indicated they preferred asking questions via the journal.

As part of the survey, former students were asked what they saw as the benefits of the course format. Ling indicated, "Having time to review video lectures at my own pace was helpful in allowing for more targeted learning. I found the journal to be a valuable way to help myself think about problems by trying to explain them to someone else. Project time in class works quite well as it allows for far more feedback than I could get otherwise, which is particularly important in this type of class." Other respondents also indicated the benefits of watching the videos on their own time, controlling the speed based on level of understanding of the content, and being able to rewind or re-watch videos was seen as helpful for the learning process. Sigrid suggested future students in these courses should "raise as many questions as possible in the journal, which will help in critical thinking and future journal publications". These comments indicate the persistent value of the experience beyond the course itself.

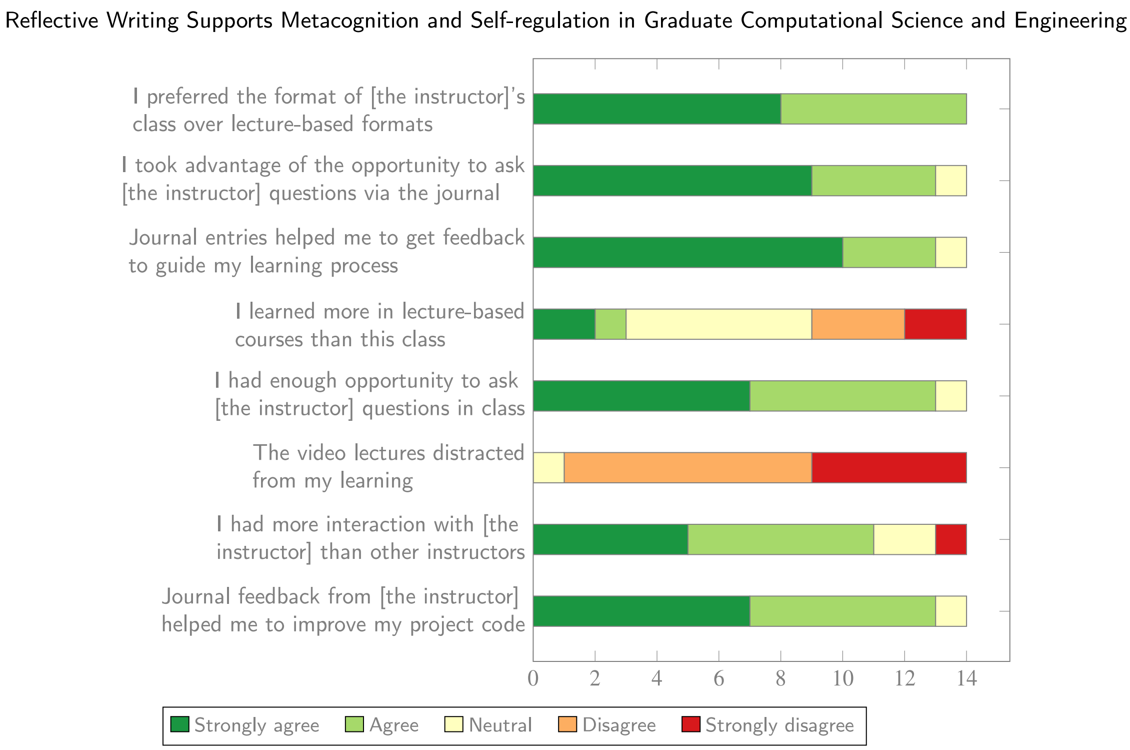

Fig. 1. Summary of survey responses.

5.5. Instructor Perceptions of the Course

This section shifts to the instructor's perspective, this paper's last author. My academic training was in the traditional lecture style and I taught this way myself for several years. However, over time, I came to realize that what I enjoy most, and am presumably best at, is direct contact with students, i.e., a departure from lecture format; in many of my classes, a substantial fraction of class time is spent on discussions. At the same time, I am uncomfortable with wholesale change of my teaching style – as I assume many are – and required backing from experienced education researchers in adopting the design described herein.

Overall, I was pleased with the success of this approach. It allowed me to shift focus away from whiteboard lectures and towards the personal interaction with students, which I enjoy much more and I think makes a teacher truly valuable. It is fun to look at a piece of code on the screen together with a student, write out a software design for a particular project, or demonstrate how to debug a problem. All of these take time, and I would not have had the time to demonstrate them if I had spent more time lecturing. I also think – and know from student feedback – that observing a professional *do* something often provides students with far deeper insight than just *hearing* in theory how to do it. A flipped classroom also blurs the *geographic separation* between lecturer at the front and students in the back, and makes it easier for shyer students to ask whatever questions they undoubtedly have.

Of course, increased interaction, personal growth, and student enjoyment are not the only metrics that determine whether a course, and its design, is successful. Students *learning content material* is still important. For this to happen in a flipped classroom, students first have to watch online videos or use other provided resources. I was worried that students might be watching videos but paying little attention. Indeed, Dingxiang described this: "First, because it's too flexible to watch the lecture, sometimes this will be myself excuse to watch it later. By later I mean two or three days. Then, I will be a bit behind the schedule. Another issue is, since I watch these videos outside of the classroom, I'm less concentrated than in normal classes. For example, I watched the video lectures during my lunch hour, at first I think this would be a better use of my lunch time, because I just need my hands and mouth to eat and my eyes are free. But my experience told me usually I cannot remember the contents in the video and have to watch the second or even the third time. Also, in the help sessions [the student worked as a TA], if I paused the video and go to answer someone's question for 15 minutes, then when I resumed to the lecture, sometimes I actually forget the material before the pause and I have to start all over again." But many students realized within the first couple of weeks that they did not *learn* what they were supposed to, and that there is no place to hide this in a course like this; the problem quickly resolves itself after that.

One significant finding was that the use of student journals diverged significantly from our expectations. First, checking these journals turns out to be surprisingly addictive once one realizes that they can be used as *two-way streets for communication*. We reviewed entries several times a week, providing advice and answers to student questions in differently colored text or as comments in the margins. Part of the appeal to the instructor is that one can frequently *observe learning happen* in the journals, as students formulate a question, backspace through sentences and entire paragraphs after realizing their own mistakes, formulate a new question, and sometimes answer it themselves – all in real time on the instructor's screen. Replying to such questions immediately, or at other times throughout the week provided students far more feedback than is possible in a regular class [59]. The additional benefit of feedback in written form was that students could refer back to it as needed. On a personal note, we found reviewing journals to be a pleasant activity at the end of the day, or during a fifteen minute break between meetings that would otherwise have been too short for more difficult thinking.

When viewed in the context of the literature on reflective writing and writing-to-learn, learning journals not only enable the instructor to keep track of what students are doing, but also serve as legitimate educational activities in their own right, see Section 2.3. Indeed, in view of the programming required for this course, it is worthwhile pointing out that most good programming practices guidelines (see, e.g., [51]) were developed by retrospectively analyzing how and why bugs ended up in large-scale software systems - a reflective process about programming.

We make the point frequently in our classes that the best programmers are often also the best at analyzing their own patterns of mistakes.

Interestingly, a substantial fraction of students express their desire to continue keeping a journal of their research progress after the end of the class, as indicated in the survey results in Section 5.4. I believe that the root for this lies in the metacognitive understanding that, as many seasoned practitioners know, writing down what does not work oftentimes clarifies what is wrong, and makes one think about the approach chosen and whether it is right. Students have informally conveyed that in many conversations: They found the solution to a problem just by writing out their question in the journal, *and that they recognized the usefulness of this approach* – i.e., they exhibit the metacognitive growth we wanted to instill in them.

As for how well students learned subject matter, we rely on our student data, qualitative analyses, and instructor perception. In the end, my conclusion is that students learned subject matter as well or better than I would have expected in a more traditional, lecture-based course built around projects. In particular, I believe that the ability to watch videos again, and to pause them to experiment with what is being demonstrated, really helps translate abstract *knowing* into concrete *doing*. Similarly, my ability to focus on one-on-one interactions allows me to *model* workflows and thought processes interactively; watching me do it allows for students reflection on *how* I do it, and *why*, and to have conversations about my choices and strategies.

Finally, an important metric from the perspective of the instructor is how much time it takes to teach a course. In this case, not noticeably more than any other course: I do not need to prepare for the details of the lecture the next day, nor grade homework; instead, I read through student journals. The time invested in these tasks is roughly equal.

5.6. Limitations

Many quotes are from journals written contemporaneously and, because students knew that the instructor would read these comments, are therefore more likely to be positive. The influence of the instructor is a limitation of this study. At the same time, from our interactions with students, we believe that students genuinely liked the format, and that the quotes presented accurately reflect student views. We have cross-checked this assertion by comparing the survey results with the class evaluations filled in by the students after the end of the semester in which they took the course. These class evaluations were anonymous, and the evidence therein closely matched what we have found in the journals and the survey responses.

While this study is focused on one specific course, we want to emphasize that the strategies discussed in this paper are not specific to our course or any single "Computational X" discipline, or indeed, any specific graduate-level science or engineering course. Indeed, the students in this course typically come from a broad spectrum of graduate programs in engineering and the natural sciences. The commonality among all of these students was that they wanted to learn the computational sciences content and needed substantial support to develop the complementary 21st century professional skills. Consequently, we believe that the course design discussed previously and our conclusions regarding its effectiveness will also broadly apply to other CS&E courses, if taught using similar principles.

6. Discussion and Conclusions

As described in the introduction, CS&E is a relatively new field, and no consensus has formed so far on how best to teach courses in this field. Rather, current instructional practices are likely best understood as an extension of the traditions of the academic backgrounds of instructors. These are most often mathematics, statistics, and computer science; and, in the case of "Computational X" courses, the traditions of the host "X" discipline.

At the same time, since most CS&E courses are not taught in stand-alone programs but, rather, as an add-on to existing programs, these courses are often exceptionally broad: As outlined in the introduction, it is not uncommon that a single course has to cover modules that relate to computer use, programming and programming models, mathematical background, and applications. As others have suggested, it is not immediately clear how one would best teach such a course [40]. In the following, we discuss our findings and their implications for the future use of technology in CS&E education.

6.1. Response to Research Questions

In this work, we have considered one CS&E course and transformed it via a project-based, flipped-classroom design, supported by technology-mediated journals and reflective writing. Although well backed up by education research (see Section 2), many of the components of our design are a deviation from the "traditional" teaching style used in many STEM disciplines; it requires a leap of faith to incorporate them into a class. At the same time, the evidence we have collected by evaluating hundreds of pages of student journals as well as a survey sent to all past students, suggests that, in response to the two research questions, students enjoyed the course design and learned lessons that extend *beyond* disciplinary content and into the realm of *becoming a professional scientist*. Indeed, this was the goal: As discussed in the introduction, the fluidity of the field, along with the demands of CS&E jobs, implies that we should focus much more on *skills* than on facts [40], and our evaluation of the course suggests that the design we chose supports this focus on reflective writing and metacognition. The principles we apply are also aligned with the broad goals described in [57].

Our most radical departure from traditional STEM course design, but also the one most useful in developing these 21st century skills, as in [59], was the inclusion of a learning journal and reflective writing essays. Most STEM students have essentially no experience with writing about themselves or their personal perspectives [24]; that is particularly true for international students from cultures in which students do not express sentiments to instructors. Many students try rather hard to avoid writing about themselves and, when prompted to be introspective about their experience, resort to statements such as "I *feel* that algorithm A is better than algorithm B". As aligned with [70], we found students initially focused on scientific content in their journals. The level of introspection was substantially raised after providing an example reflective writing sample by the instructor writing about their own work. On the other hand, a subset of students really understood the purpose of these assignments and wrote in depth about, for example, how they have learned to analyze patterns of mistakes and what measures they are taking to prevent these from happening again.

These findings allow us to answer the research questions we have laid out in Section 3. First, students generally have difficulty expressing their own thought processes and being introspective about their skills in their journals at the beginning of the semester; but, they ultimately demonstrated attention to their metacognitive and self-regulation processes that enabled them to make explicit their critical-thinking and problem-solving skills, and were able to see the utilization of those skills for themselves (RQ1). Second, the results from the survey indicate that the metacognitive practices and self-regulated learning behaviors developed throughout the class formed the basis for the ongoing use of a journal and students' positive assessment of the course after it had concluded (RQ2).

6.2. Future Considerations for Technology and Faculty Development

Preparing professional scientists and engineers is arguably the most fundamental goal of STEM graduate programs. Courses that teach students content *and* develop them as skillful and independent professionals should be the focus of that preparation. We continue to believe, and are supported by our data, that the essays we have made part of our course design are useful tools in teaching students the skills to be independent scholars, researchers, and professionals. This is also supported by experiences we have from other areas; for example, [6] found that reflective writing helped students connect their mathematics content learning to future professional practice. While the students in the present study did not clearly present themselves as future professionals, our data show that they did display increased self-regulated behaviors throughout the course, followed succinctly by an increased display of the targeted professional skills of problem solving and critical thinking. As a result, we hope readers will take from this article a sense that such course designs are not only desirable from a theoretical perspective, but are indeed practically achievable *and that they work*. Implementing a course design like the one we present requires a bit of an adventurous spirit as it deviates from how most professionals in the CS&E are teaching and were taught themselves – particularly regarding the educational technology –, but the preponderance of evidence suggests it is worthwhile, manageable, and surprisingly fun.

Given the tremendous growth in distance education since the beginning of the pandemic in 2020, the technological supports available today for making videos and interacting with students online continue to ease the burden of a course design like we present here. As a consequence, this work does not actually advocate for new or different technology: Everything necessary to design and implement a course like this exists, and is readily available at most universities through their learning management system. But, as in many STEM fields, most teaching in CS&E does not use much technology beyond whiteboards and PowerPoint slides, and our results indicate that thoughtful use of widely available technology shows substantial benefits to students learning 21st century skills.

Much of our discussion emphasizes student learning and development, yet this course design also facilitated *faculty professional development* and expansion of one's teaching repertoire. If we argue that 21st century skills are important, then that also needs to include teacher experience with modern teaching styles [40]: In addition to new skills one may acquire by managing the technology associated with videos, journals, and interactive class time, there is a lot to learn from the dense communication with students. Many faculty feel it is difficult to get meaningful student feedback about their teaching. Although this was not the original goal, we found journals facilitated student feedback and a greater awareness of general teaching practices. This includes small improvements in practices (e.g., font sizes and colors that students noted in the journals were difficult to see in video lectures, and that we changed in subsequently recorded video lectures; or making slides available alongside the videos that use them). Similarly, one journal included a request for homework – not extra credit – indicative of a realization that smaller assignments (independent of the semester-long project) early in the course would have helped guide the student during their exploration of deal.II as part of their project work. Most of the earlier tutorials of this library now have suggested mini-projects as a consequence. Finally, journals also included comments about student learning – such as some of the quotes presented above – and this has enabled us to both refine our course design and become better teachers.

In summary, we believe that the course design discussed herein has worked well for the course we teach and could serve as a model for many other CS&E and "Computational X" courses. Importantly, we believe that the course design facilitated student growth in "soft skills" that include a better awareness of how they learn, how to effectively communicate, and critical thinking and problem-solving skills. It is these skills that we believe will be important to our students in becoming and being productive professionals.

Acknowledgments

W. Bangerth was partially supported by the National Science Foundation under award OAC-1835673 as part of the Cyberinfrastructure for Sustained Scientific Innovation (CSSI) program; by award DMS-1821210; and by the Computational Infrastructure in Geodynamics initiative (CIG), through the National Science Foundation under Award EAR-1550901 and The University of California Davis. The authors also gratefully acknowledge discussions with the many colleagues who gave feedback on early versions of this manuscript, the course design, and the video lectures. This includes, in particular, Matthias Maier, Luca Heltai, Timo Heister, and Daniel Garcia. The video lectures were recorded at the KAMU television studio on the Texas A&M University campus; the support of the professionals there was fantastic. The support by Zhang Jiankang, who made copies of the video lectures available on the Chinese bilibili service, is also greatly appreciated. Finally, we are grateful to the students in these classes who willingly let us evaluate their journals and replied to the survey!

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Wolfgang Bangerth reports financial support was provided by National Science Foundation.

References

[1] Alrashidi H, Ullman TD, Joy M. An empirical evaluation of a reflective writing framework (RWF) for reflective writing in computer science education. 2020 IEEE Frontiers in Education Conference (FIE). IEEE; 2020. https://doi.org/10.1109/fie44824.2020.9273975.
[2] Alt D, Raichel N. Reflective journaling and metacognitive awareness: Insights from a longitudinal study in higher education. Reflective Practice 2020;21(2):145–58.
[3] Anderson JR, Fincham JM. Extending problem-solving procedures through reflection. Cognitive psychology 2014;74:1–34.
[4] Angeli L, Laconich JJJ, Marchese M. A constructivist redesign of a graduate-level CS course to address content obsolescence and student motivation. Proceedings of the 51st ACM Technical Symposium on Computer Science Education. ACM; 2020. https://doi.org/10.1145/3328778.3366910.
[5] Aspinwall L, Miller LD. Diagnosing conflict factors in calculus through students' writings. The Journal of Mathematical Behavior 2001;20(1):89–107. https://doi.org/10.1016/s0732-3123(01)00063-3.
[6] Zarestky Jill, Bigler Michelle. Reflective journaling in mathematics: Insights into the development of future teachers. 3rd33. Journal of New Horizons in Adult Education and Human Resource Development; 2021. p. 52–65. https://doi.org/10.1002/nha3.20334.
[7] Zarestky Jill, Bangerth Wolfgang. Teaching high performance computing: Lessons from a flipped classroom, project-based course on finite element methods. EduHPC '14: Proceedings of the Workshop on Education for High-Performance Computing. Piscataway, NJ: IEEE Press; 2014. p. 34–41. https://doi.org/10.1109/EduHPC.2014.10.
[8] Arndt Daniel, et al. The deal.II finite element library: Design, features, and insights. Computers & Mathematics with Applications 2021;8:407–22. https://doi.org/10.1016/j.camwa.2020. 02.022.
[9] Bandura A. Social foundations of thought and action : a social cognitive theory. Prentice-Hall series in social learning theory.. Englewood Cliffs, N.J: Prentice-Hall; 1986.
[10] Berrett D. How 'flipping' the classroom can improve the traditional lecture. The Chronicle of Higher Education 2012.https://www.chronicle.com/article/How-Flipping-the-Classroom/130857
[11] Bjork RA, Dunlosky J, Kornell N. Self-regulated learning: Beliefs, techniques, and illusions. Annual review of psychology 2013;64:417–44.
[12] Borasi R, Rose BJ. Journal writing and mathematics instruction. Educational studies in mathematics 1989;20(4):347–65.
[13] Brodie L, Zhou H, Gibbons A. Steps in developing an advanced software engineering course using problem based learning. Engineering Education 2008;3 (1):2–12. https://doi.org/10.11120/ened.2008.03010002.
[14] Bruno A, Dell'Aversana G. Reflective practicum in higher education: the influence of the learning environment on the quality of learning. Assessment & Evaluation in Higher Education 2017;43(3):345–58. https://doi.org/10.1080/02602938.2017.1344823.
[15] Cavalcante Koike CMe, Viana DM, Vidal FB. Mechanical engineering, computer science and art in interdisciplinary project-based learning projects. International Journal of Mechanical Engineering Education 2018;46(1):83–94.
[16] Cesar E, Cortés A, Espinosa A, Margalef T, Moure JC, Sikora A, Suppi R. Introducing computational thinking, parallel programming and performance engineering in interdisciplinary studies. Journal of parallel and distributed computing 2017;105:116–26.
[17] Chng SI. Incorporating reflection into computing classes: models and challenges. Reflective Practice 2018;19(3):358–75.
[18] Colthorpe K, Ogiji J, Ainscough L, Zimbardi K, Anderson S. Effect of metacognitive prompts on undergraduate pharmacy students self-regulated learning behavior. American journal of pharmaceutical education 2019;83(4).

[19] Cooley L. Writing in calculus and reflective abstraction. The Journal of Mathematical Behavior 2002;21(3):255–82. https://doi.org/10.1016/s0732-3123(02)00129-3.
[20] Coursera, 2021, (https://www.coursera.org/).
[21] T. Davis, 42 lectures on direct methods for sparse linear systems, 2021, (https://www.youtube.com/playlist?list=PL5EvFKC69QIyRLFuxWRnH6hIw6e1-bBXB).
[22] Deepamala N, Shobha G. Effective approach in making capstone project a holistic learning experience to students of undergraduate computer science engineering program. Journal of Technology and Science Education 2018;8(4):420–38.
[23] DeLozier SJ, Rhodes MG. Flipped classrooms: A review of key ideas and recommendations for practice. Educational Psychology Review 2017;29(1):141–51.
[24] Demmans Epp C, Akcayir G, Phirangee K. Think twice: exploring the effect of reflective practices with peer review on reflective writing and writing quality in computer-science education. Reflective Practice 2019;20(4):533–47.
[25] Deslauriers L, Schelew E, Wieman C. Improved learning in a large-enrollment physics class. Science 2011;332(6031):862–4. https://doi.org/10.1126/science.1201783.
[26] Dewey J. How we think. D. C. Heath & Co.; 1933.
[27] Dündar S. Does writing have any effect on mathematics success? Journal of Education and Training Studies 2015;4(1). https://doi.org/10.11114/jets.v4i1.989.
[28] edX, 2021, (https://www.edx.org).
[29] Evans GJ. Windmills of your mind: metacognition and lifelong learning. Proceedings of the Canadian Engineering Education Association (CEEA) 2018.
[30] Finkenstaedt-Quinn SA, Petterson M, Gere A, Shultz G. Praxis of writing-to-learn: A model for the design and propagation of writing-to-learn in STEM. Journal of Chemical Education 2021;98(5).
[31] Flavell JH. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. American psychologist 1979;34(10):906.
[32] Ge X, Allen JK, Mistree F. Career sustaining competencies for managing disruptions and innovative problem solving in a digitized world. 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT). 2018. p. 309–10. https://doi.org/10.1109/ICALT.2018.00128.
[33] George SE. Learning and the reflective journal in computer science. Proceedings of the Twenty-Fifth Australasian Conference on Computer Science - Volume 4. ACSC '02. AUS: Australian Computer Society, Inc.; 2002. p. 77–86.
[34] Groeneveld W, Vennekens J, Aerts K. Software engineering education beyond the technical: A systematic literature review. CoRR 2019;abs/1910.09865. arXiv:1910.09865.
[35] Harvey M, Coulson D, McMaugh A. Towards a theory of the ecology of reflection: Reflective practice for experiential learning in higher education. Journal of University Teaching and Learning Practice 2016;13(2):1–20.
[36] Hoehn JR, Lewandowski H. Framework of goals for writing in physics lab classes. Phys. Rev. Phys. Educ. Res. 2020;16:385. https://doi.org/10.1103/PhysRevPhysEducRes.16.010125.
[37] Indiana University Center for Postsecondary Research, The Carnegie Classification of Institutions of Higher Learning, 2018, (https://carnegieclassifications.iu.edu/).
[38] Johns C. Becoming a reflective practitioner. fifth. Wiley & Sons; 2017.
[39] Jurdak M, Zein RA. The effect of journal writing on achievement in and attitudes toward mathematics. School Science and Mathematics 1998;98(8):412–9. https://doi.org/10.1111/j.1949-8594.1998.tb17433.x.
[40] Kalil CA, Grant T. A social learning theory model for understanding team-based professional communication learning for computer science students. South African Journal of Higher Education 2021;35(3). https://doi.org/10.20853/35-3-2912.
[41] Khan Academy, 2021, (https://www.khanacademy.org/).
[42] Kolb DA. Experiential learning : experience as the source of learning and development. Englewood Cliffs, N.J: Prentice-Hall; 1986.
[43] Lage MJ, Platt GJ, Treglia M. Inverting the classroom: A gateway to creating an inclusive learning environment. The Journal of Economic Education 2000;31(1):30–43. https://doi.org/10.1080/00220480009596759.
[44] Lawler J, Joseph A. Collaborative course design of entrepreneurship projects in a college of computer science and information systems. Information systems education journal. 2020;18(3):54–70.
[45] Lincoln YS, Guba EG. Naturalistic inquiry. Beverly Hills, Calif: Sage Publications; 1985.
[46] Liu Z, Yin H, Cui W, Xu B, Zhang M. How to reflect more effectively in online video learning: Balancing processes and outcomes. British journal of educational technology. 2021.
[47] Liuolienė A, Metiūnienė R. Students' learning through reflective journaling. Coactivity: Philology, Educology/Santalka: Filologija, Edukologija 2011;17(4):32–7.
[48] Loksa D, Ko AJ. The role of self-regulation in programming problem solving process and success. Proceedings of the 2016 ACM conference on international computing education research. 2016. p. 83–91.
[49] Lumpkin A. Metacognition and its contribution to student learning introduction. College Student Journal 2020;54(1):1–7.
[50] Mazur E. Farewell, lecture? Science 2009;323(5910):50–1. https://doi.org/10.1126/science.1168927.
[51] McConnell S. Code Complete: A Practical Handbook of Software Construction. 2nd. Microsoft Press; 2004.
[52] McLaughlin JE, Roth MT, Glatt DM, Gharkholonarehe N, Davidson CA, Griffin LM, Esserman DA, Mumper RJ. The flipped classroom. Academic Medicine 2014;89(2):236–43. https://doi.org/10.1097/acm.0000000000000086.
[53] Merriam SB, Tisdell EJ. Qualitative research : a guide to design and implementation. fourth edition. San Francisco, CA: Jossey-Bass, a Wiley brand; 2016.
[54] MIT OpenCourseWare, 2021, (https://ocw.mit.edu/).
[55] M. Myers, R. van de Geijn, Linear algebra – foundations to frontiers, 2021, (https://www.edx.org/course/linear-algebra-foundations-to-frontiers).
[56] Mărăşoiu A. Understanding, problem-solving, and conscious reflection. Acta Analytica 2019;34(1):71–81.
[57] National Academies of ScienceEngineering, and Medicine. Graduate STEM Education for the 21st Century. National Academies Press; 2018. https://doi.org/10.17226/25038.
[58] Ormrod JE. Human Learning. seventh. Pearson; 2016.
[59] Pedrosa D, Fontes MM, Araújo T, Morais C, Bettencourt T, Pestana PD, Morgado L, Cravino J. Metacognitive challenges to support self-reflection of students in online software engineering education. 2021 4th International Conference of the Portuguese Society for Engineering Education (CISPEE). IEEE; 2021. p. 1–10.
[60] Rashid R. Cultivating 21st century skills in PhD students. Asian Journal of the Scholarship of Teaching and Learning 2019;9(1):70–80.https://nus.edu.sg/cdtl/engagement/publications/ajsotl-home/archive-of-past-issues/v9n1/cultivating-21st-century-skills-in-phd-students
[61] ur Rehman Pasha MA, Pasha S. Missing elements of computer science curricula 2013. Global Journal of Computer Science and Technology 2012.https://computerresearch.org/index.php/computer/article/view/284
[62] Reyes-Santander P, Ramos-Rodríguez E. Learning journals for university students' reflection on advanced algebra. Reflective Practice 2015;16(5):623–35. https://doi.org/10.1080/14623943.2015.1064385.
[63] Rivers ML. Metacognition about practice testing: a review of learners beliefs, monitoring, and control of test-enhanced learning. Educational Psychology Review 2021;33(3):823–62.
[64] Rüde U, Willcox K, McInnes LC, Sterck HD. Research and education in computational science and engineering. SIAM Review 2018;60(3):707–54. https://doi.org/10.1137/16m1096840.
[65] Ryan RM, Deci EL. Self-Determination Theory. Guilford Press; 2018.
[66] Schön DA. The reflective practitioner. Basic Books; 1983.
[67] Schön DA. Educating the reflective practitioner. Basic Books; 1987.
[68] Schraw G, Crippen KJ, Hartley K. Promoting self-regulation in science education: Metacognition as part of a broader perspective on learning. Research in science education 2006;36(1):111–39.
[69] SIAM Working Group on CSE Education. Graduate education in computational science and engineering. SIAM Review 2001;43(1):163–77. https://doi.org/10.1137/s0036144500379745.
[70] Singh P, Tempero E, Luxton-Reilly A, Zhang S. Digging into computer science students learning journals. Koli Calling '20: Proceedings of the 20th Koli Calling International Conference on Computing Education Research. Koli Calling '20. New York, NY, USA: Association for Computing Machinery; 2020. https://doi.org/10.1145/3428029.3428048.
[71] Szydlik JE, Szydlik SD, Benson SR. Exploring changes in pre-service elementary teachers' mathematical beliefs. Journal of Mathematics Teacher Education 2003;6(3):253–79. https://doi.org/10.1023/a:1025155328511.
[72] The Carpentries: Software Carpentry, Library Carpentry, Data Carpentry, 2021, (https://carpentries.org/).
[73] The Open University on iTunes U, 2021, (https://www.open.edu/itunes/).
[74] Towndrow PA, Ling TA, Venthan AM. Promoting inquiry through science reflective journal writing. Eurasia Journal of Mathematics, Science and Technology Education 2008;4(3):279–83. https://doi.org/10.12973/ejmste/75350.
[75] Walker P, Finney N. Skill development and critical thinking in higher education. Teaching in Higher Education 1999;4(4):531–47.
[76] Zheng B, Ward A, Stanulis R. Self-regulated learning in a competency-based and flipped learning environment: learning strategies across achievement levels and years. Medical education online 2020;25(1):1686949.
[77] Zimmerman BJ. Attaining self-regulation: A social cognitive perspective. Handbook of self-regulation. Elsevier; 2000. p. 13–39.
[78] Zinsser W. Writing to learn. Harper & Row; 1988.