DOI: 10.5324/njsteme.v7i1.4827

Students’ response to the introduction of active learning and computational practices in a bachelor-level earth science course

T.E. Møller*, S.J. Kvarøy, and B. Hannisdal,
Department of Earth Science, University of Bergen, Norway.
*Current address: *The Learning Center, BI Norwegian Business School, Oslo, Norway*

ABSTRACT: Earth science education should provide space for students to engage with real-world problems involving complex earth systems and their societal implications. However, the ill-structured nature of such problems creates uncertainty among both teachers and learners. To explore the effect of this uncertainty, we designed and taught a bachelor-level earth science course that introduced geology students to computational practices allowing them to work with authentic data, methods, and ill-structured problems. In addition, the course was designed entirely for active learning and formative assessment. Students provided reflection notes as part of every learning activity, and a series of focus group interviews were conducted at the end of the course to triangulate student experiences with computational practices and problem solving in groups. Our findings suggest that the introduction of computational practices to novices, combined with the transition from passive to active forms of learning, were major sources of uncertainty. This uncertainty needs to be understood and confronted in order for students to engage with real-world problems in an effective manner. On the other hand, our results indicate that collaborative work in groups can alleviate some of the effects of this uncertainty. Our study also supports the systematic use of student reflections as a means of fostering feedback literacy among students as well as teachers.

1 INTRODUCTION

Earth science education has gained new levels of importance as climate change and sustainability have risen on the global agenda [1, 2]. It provides unique opportunities for students to seek the knowledge, practices and dispositions needed to confront the complexity of life-sustaining earth systems and their socio-economic impacts [3-7]. However, the ill-structured nature of real-world problems involving dynamical earth systems raises a challenge for teachers and learners [8]: Unlike classic textbook problems that typically have solution recipes and a correct answer, ill-structured problems require a different approach [9, 10], e.g., the problem itself is often unclear; solution strategies cannot be copied from other situations; information is incomplete and conflicting; students need to confront their personal beliefs, make choices and provide argumentation. In a learning situation, these properties of real-world problems inevitably create uncertainty among both students and teachers [8, 9]. What implications does this inevitable uncertainty have for learning design and students' learning experience at a bachelor level?

In an influential analysis of higher education curriculum, Barnett and Coate [11] emphasise that higher education is about opening up new spaces with opportunities for new ways of understanding the world and of being in the world. Hence, both students and teachers have to dare to – and be allowed to – take chances ("*a pedagogy of mutual risk*" [12]) and be willing to engage with the uncertainty of the world ("*a committed uncertainty*" [13]). In a similar vein, Biesta [14] speaks of the "*beautiful risk*" of education. How do these notions of uncertainty and risk play out in teaching and learning practice?

The course GEOV114 *Introduction to geobiology* was designed with a view to exploring these questions. Geobiology is the study of interactions between the biosphere and other components of the Earth system. For example, billions of years of biological evolution has led to fundamental changes in Earth's surface environments, geochemical cycles and climate, storing energy in a biosphere that has literally fuelled human civilization. Geobiological processes are thus central to the global challenges represented by the sustainable development goals. Although it integrates several sub-disciplines with long traditions (e.g. biogeochemistry, paleobiology), geobiology is considered a relatively young field of research due to its modern orientation towards microbial ecosystems and molecular biology. Geobiology was established as a research group at the University of Bergen Department of Earth Science in 2007, and courses soon appeared at the master's level. Due to the conservative nature of curricula, however, the mandatory bachelor-level course GEOV114 was first offered to 5th semester students in 2019.

We first outline some of the key choices made in the learning design of the course before turning to the specific research questions of this study.

Computational practices. The aim of introducing computational practices was to empower students to reproduce and critically assess the results of scientific research, as a way of opening new spaces in the curriculum: from the safe but inert facts of a textbook to a more turbulent research in constant motion. In our own personal experience, many of the concepts encountered in science education, such as dynamical systems, modelling, and statistical testing, became tangible and open to inquiry and self-determined learning only after figuring out how to implement them in a programming language. Being able to visualise and analyse data, to build and explore systems, and to run numerical experiments, helped open up new ways of asking and answering questions. Hence, our primary motivation for introducing students to computational practices was to provide them with opportunities to have the same kind of experience. Our personal perspective is consistent with reported linkages between programming experience and problem-solving abilities [15], is aligned with the conceptual classification of systems thinking as a subset of computational thinking [16, 17] and is empirically supported by a meta-analysis suggesting that learning computer programming has some transfer value to other domains of learning [18]. We prefer the term "computational practices" rather than "programming" because the latter can be perceived as a more restrictive notion of following precisely structured algorithmic recipes. We thus agree with Weintrop et al. [16], who deconstructed computational thinking into a richer taxonomy of practices used by scientists: data practices, modelling and simulation practices, computational problem-solving practices, and systems thinking practices. Fundamentally, this taxonomy describes various aspects of processing data, from acquisition through analysis to model evaluation. In our experience as working scientists, this process involves a set of important heuristics ("tricks of the trade") that help clarify goals and explore solution pathways in real-world problems. Building proficiency in computational practices may thus be regarded as acquiring a new skill rather than new knowledge [16, 19, 20]. Therefore, teaching computational practices is arguably best carried out by letting the students work on tasks that focus on building skill. For this purpose, we chose an active learning approach.

Active learning. The course was designed as a series of learning activities where two were performed in a geomicrobiology laboratory, one was doing geology and palaeontology in the field, and the rest were computational activities carried out in a room designed for collaborative, active learning. The explicit focus on real data and methods created the need for a diverse range of literature and digital resources that would not be available in any textbook. There were no lectures attempting to cover a pre-determined syllabus. Instead, students used class hours to work together on acquiring data from online databases, during field work, or from experiments in the lab, and to use computational methods for analysis, visualisation, and modelling of these data. The teaching team was present at all times. A final letter grade is a requirement for mandatory courses in our programme, and a total of seven learning activities counted for 100% of the final grade, replacing the traditional final exam. GEOV114 thus differed substantially from most courses students had attended in their previous four semesters, because neither full-fledged commitment to active learning nor computational practices were common in our bachelor programme. We believed that the aggregated uncertainty stemming from the unfamiliar course design would be best met in groups, so we assigned all students to groups of three or four (10 groups in total) that would last for the duration of the semester at random and presented them with a progression of assistance: consult the instructional materials and online resources first, your group second, other groups third, and finally the teachers or teaching assistants (TAs). This setup was facilitated by the course being taught in a room equipped specifically for active learning, with group tables seating up to six students, each with its own screen and whiteboard that students could use for in-group sharing and discussion. A physical learning space of this kind makes it easier to actively approach and encourage students during the activities, both of which are important facilitation strategies to successfully implement active learning [21].

Formative assessment and reflection. In addition to informal feedback during class hours, each learning activity had an associated rubric with assessment criteria describing to students what aspects of their working process they would receive feedback on. Some critiera were generic and common across most activities (e.g. referring to aspects of writing, programming, communication, or collaboration), others were specific to a given task, and some were linked to the use of previous feedback for improvement and progression. Each assessment rubric also contained criteria that prompted students to reflect upon specific aspects of each activity. Three achievement levels were specified for each criterion, and these fine-grained rubrics were used to generate written individual feedback reports for each activity. Students did not receive any scores or grades on the individual activities, only feedback comments (colour coded by achievement level), but at the end of the course the achievement levels were used quantitatively as input to the total, summative assessment and final grade. Even though the order and contents of the learning activities were determined by the teachers, adjustments were made to activities and rubrics during the course in response to student feedback and reflection notes, which is recognised as an important step teachers may take to reduce resistance to active learning [22].

The main challenge in designing this course was introducing real-world computational practices to students that had little or no experience with programming or related practices. The teachers are experienced computational practitioners who are very familiar with having to face the unknown (e.g., a different programming language, a novel method) in the fast-paced world of computing. Hence, they are used to spending time in what has been referred to as the "learning pit" [23]: the sustained feeling of incompetence and frustration associated with not understanding, not knowing, until you slowly work your way out of the pit. This situational awareness prompted specific measures: The learning activities were designed assuming no prior experience with programming, data processing, visualisation or modelling. The first computational activity was a basic introduction to Python using tried and tested instructional materials aimed at novices, ranging from installation and interfacing with the Jupyter notebook environment, via basic syntax, programming concepts and control flow, to functions, file I/O and visualisations. This first introduction to programming also had a detailed assessment rubric and provided a feedback report to the students, but it was not included in the final summative assessment. Subsequent activities built on and referred to these tutorials, but introduced authentic practices involving real data (e.g. streaming from online databases), visualisation (e.g. map projections) and methods (e.g. machine learning). Subsequent computational activities were centred around accessing online databases, in other words authentic data, to, for instance, plot sedimentological maps or extinction rates inferred from fossil records. All these activities combined computational with geobiological questions (e.g., first plot a map and then discuss its main features). In the final activity of the course, activity number seven, students were tasked with programming part of the global carbon cycle for a geological period of their choosing. Before coding the model, they were instructed to make a conceptual schematic of the carbon cycle model and the fluxes between its reservoirs. After having received feedback on the schematic, the students could start programming, running and tuning the model and answering questions related to their task. Given the ill-structured nature of this activity, with no given answers, no clear solution pathway and limited information provided at the beginning [9], the assessment criteria of this activity in particular emphasised argumentation and discussion of one's findings and beliefs.

Our line of reasoning in designing the course and when presenting it to students was that computational practices would enable them to work with authentic data and methods, and help promote expert-like behaviour when faced with real-world (ill-structured) problems. Furthermore, we argued that both the problems and the practices would require an active learning approach and formative assessment. The goal of this study, which is intended as a contribution to the scholarship of teaching and learning, is to explore the consequences of introducing computational practices and active learning for student learning and well-being in a discipline-based earth science course. Specifically, we are interested in how students experience the uncertainty that is expected to arise from the nature of ill-structured problems as well as from a lack of previous experience with both computational methods and active learning pedagogies. We pose the following research questions: 1. How do students respond to the uncertainty elicited by the introduction of computational practices in a discipline-based course? 2. How is the students' motivation to learn computational skills as well as discipline-based content influenced by two main pedagogical choices: (a) an active-learning approach to teaching computational practices, and (b) assigning students randomly to groups that they will remain in for the duration of the semester?

Our notion of uncertainty is related to the concept of resistance towards active learning, which has received much attention [21, 22, 24-27]. Likewise, strategies to reduce or even leverage uncertainty overlaps with advice on how to aid implementation of active learning, (e.g., [21]). However, uncertainty does not necessarily lead to resistance, and resistance does not necessarily arise from uncertainty, hence the two should not be conflated.

Our study is intended as a contribution to the scholarly discussion of teaching and learning in higher education. While grounded in an earth science context, we attempt to write about our findings and experiences such that we believe our study is relevant to other teachers within STEM who are considering, planning, or in the process of implementing or reviewing computational practices and ill-structured problems in their course syllabus. Our findings highlight fundamental uncertainty regarding what is to be learned, and how it may prevent the course from being completed as intended, and we propose solutions chiefly revolving around cooperation among students as well as between students and teaching staff to solve the problems that are applicable even as the course is being implemented.

2 METHODS

Our specific research focus unequivocally calls for a qualitative design, due to its ability to capture and represent the perspectives, views and emotions of the students attending the class, as well as cover the contextual (i.e. classroom) conditions in which they take place by collecting data from several different sources [28, 29]. Similar approaches have been adopted successfully by comparable studies (e.g., [19, 22, 30]). Moreover, the relatively low number of students enrolled to GEOV114 (n = 38) renders results from quantitative methods ambiguous if not invalid. Furthermore, our combined role, performing research on (the first iteration of) a course we are also teaching, calls for a method that allows for this specific context. Because it permits the researcher an active role [31], we find Grounded Theory (GT) to be a suitable methodology for addressing our research questions.

2.1 Data collection and treatment

A total of, 25 (15 female/10 male) out of 38 (19/19) students volunteered to participate in the study. Of these, all agreed to let us use their handed-in project assignment, 11 (8/3) wrote and handed in at least one reflection note, and 20 (12/8) agreed to participate in one of five focus group interviews after the project assignment was handed in. We also asked all students enrolled in the course whether they would let us use all the assignments they had handed in during the course. Twenty-one students agreed, 17 of which were enrolled in the initial project, and four who were not. As recommended by Yin [29], we use multiple data sources to corroborate our findings, with the aim of strengthening our conclusions.

Because we as investigators were also involved in grading the students, all written research data was submitted to a secure online server and not accessible to anyone involved with grading until after the option to appeal had expired. Interviews were led by two TAs who were not involved in grading. We acknowledge the possible insider's dilemma posed by letting interviewers, possibly carrying preconceptions about the interviewees, or vice versa, conduct these interviews. We have, however, found no indications of such during the processes of transcribing or coding any of the interviews.

Introduction-level courses at Norwegian universities are by default taught in Norwegian, hence all course and research material were originally written in Norwegian. However, students were allowed to answer any task in English if they so preferred. All direct mention of course material or research questions and answers have consequently been translated from Norwegian to English unless stated as verbatim.

2.2 Project-related reflection notes

All participants were asked to regularly write reflection notes for the duration of the research project, corresponding to learning activity seven (described in paragraph two under the section "Formative assessment and reflection" in the introduction). Note that unlike learning activity-specific notes (see below), these project-related notes were not mandatory or part of the assessment. The reflection notes were open-ended, but contained three tentative questions the students could choose to answer:

1) Think of a situation that occurred during today's work with the carbon cycle model where you felt frustration, mastery or another strong emotion. Describe the situation (but not yet what you were thinking or feeling)
2) What did you think and feel in this situation? Why?
3) What did you learn from this situation? In what way will your thoughts and emotions connected to the situation affect your work to come?

A total of 28 reflection notes were submitted.

2.3 Learning activity-specific reflection tasks

Most learning activities throughout the course included two to four brief questions for reflection at the end of each assignment. These questions were mandatory, and assessed based on set criteria, partly to let the students themselves reflect upon their performance and specific aspects of the assignment, partly to provide feedback to the teachers about specific aspects of the learning activity and associated instruction, and to ensure individual grading even of predominantly group-based assignments. The students received feedback on the quality of the reflection, regardless of whether the tone of the reflection was positive or negative.

2.4 Focus group interviews

The five focus group interviews lasting for approximately one hour were carried out using the course's two TAs as interviewers. There were 4, 4, 4, 2, and 6 interviewees in each interview, respectively. All five interviews were based on the same semi-open interview guide developed by the interviewers and the authors according to principles outlined in [32], e.g. relating our questions to specific goals, preparing multiple ways to phrase a question and asking colleagues to comment on them. Audio was recorded for each interview and transcribed verbatim, including pauses, stuttering, hesitation and short replies from the interviewers such as "yes" and "OK". We motivate this choice by noting and acknowledging the importance of the widespread tendency of concurrence and agreement uttered by our interviewees through the recorded material. A total of 3799 utterances were transcribed.

To analyse the transcribed interviews, we applied GT, a method used to construct theory based on data [33]. To maintain a unified and focused text set as the basis for constructing our theory, we chose to not include the reflection notes, nor the activity-specific reflection tasks in this process. Building a GT consists of three primary coding steps: Open, axial and selective coding [29]. For the first step of establishing our GT, open coding, we decided to use a system of main and sub codes to code each utterance. This step was carried out collaboratively between three researchers. Main codes, such as *attitude towards uncertainty*, *perception of (course) relevance*, and *motivation for collaboration*, were generated collaboratively based on one interview. Sub-codes were categorical, such as whether the utterance expressed positive or negative attitude towards the topic of the main code, or whether the utterance concerned a specific aspect of the course, such as programming or collaboration, when appropriate within the context of the main code. We decided that to best convey the expressed meaning and nuances of each utterance, we allowed multiple main codes and sub-codes assigned to each utterance. This choice made it easier for us to break down the expressed sentiment of each utterance as well as map a more enriched context for each code created during the first stages of coding.

The enriched context obtained from allowing multiple codes and sub-codes per individual utterance during open coding allowed us to easily establish coding categories as part of the second step of GT, namely axial coding. During the subsequent step of selective coding, namely careful analysis of utterances, codes and categories, we found that the main category emerging from our interview data was the response to uncertainty. Activity-specific reflection tasks and reflection notes were not coded but often supplement findings from the coded interviews. Excerpts from all three sources of data are referred to below.

3 RESULTS

3.1 Students' response to uncertainty

Students' response to uncertainty emerged from the data as the central category. As underlined by our research questions, our primary concern here is the uncertainty that students experience when computational practices are introduced as a means of finding solution pathways in the context of authentic data, and which must be understood in order for efficient teaching and learning of computational thinking and practices to happen [15, 17, 34]. We use the term "response" because we are interested in student actions beyond their immediate reaction to uncertainty, including choices of strategies for exploring solution pathways, individually and in groups. The focus on response highlights factors that may ameliorate or exacerbate the feeling of uncertainty and therefore, unlike the immediate reaction, covers the impulse to respond to the experienced feeling. As the experienced uncertainty may be compounded by several disparate factors like collaboration, decision-making and lack of information [9], the emphasis on response in the coded material enables us to separate the different sources of uncertainty and examine how they interact and impact one another. This raises the prospect of working more systematically with student uncertainty when applying computational practices in one's teaching.

Axial coding yielded two categories. The first category, transition to active learning, encompassed the entire course from start to finish. The second category, working as a group on computational tasks, was centred around the final learning activity in the course.

3.2 Uncertainty and the transition to active learning

Using computational practices as a means of transitioning the course from a passive to active learning environment was challenging for many students. In line with Weimer's [24] discussion of student resistance, this challenge was probably amplified by the students expecting something else entirely from the course: "*[I] did not expect it to be so based on programming. More on the syllabus, in a way.*" Here it seems that the student conceptualises learning as something you receive rather than acquire; something you can summarise as a set of bullet points on a sheet of paper. In contrast, acquisition of new skills is perceived as "not learning". This is not surprising considering that most prior courses have been graded using a final exam. Final exams are arguably designed to test memory more than skill [35], and in this context the above utterance may be described as a strategy for aligning oneself with the prospect of the best possible grade.

The course content and learning activities were designed to not require previous experience with programming. Nevertheless, as the course progressed, oral feedback from the students during teaching indicated that computational activities were too hard and time-consuming. Such feedback is common in active learning classrooms [36]. Generally having no prior experience with programming, many students were thrown into the learning pit [23]: students expressed a sense of uncertainty when confronted with the mere notion of programming, tantamount to formulating a simple sentence in a foreign language: "*The only thing that really helped us understand the programming was the teacher going through [the code] line by line.*" This utterance, which was given in the context of the final learning activity toward the end of the course, suggests that the student is not yet proficient enough at programming to have ascended from the learning pit.

It has been shown that active learning methods may lead to lower self-reported learning outcome, even though tests show that realised learning outcomes are indeed higher than for passive methods, causing some students to prefer classical, teacher-centred lecturing despite the lower learning outcome [37]. This attitude was true for our students as well: "*I really think all of learning activity seven could have been done as lectures.*" And "*With a little more concrete setup, with a more defined result you are supposed to reach, I think I would have learned a lot more.*" The students here refer to the lack of a clearly defined problem-solving strategy, which is common [15], and not to a lack of clearly defined assessment criteria.

Ill-structured problems place a strong but often implicit emphasis on skills rather than knowledge, for example through having the students themselves design the path to solving the problem [9]. The focus on skill over "bullet point knowledge" is another factor that may leave students demotivated as they are not experiencing the learning outcome they are used to [20, 36]. This was also often reported by our students:

> We were asked about the assignment, that it was open to interpretation, or something. I don't like open questions/tasks. I like to be told what is expected.
>
> It is always easier when you know what you are supposed to do.
>
> The lack of observable progress made me feel like I learned less from activity 7 than previous activities.
>
> We can't see the same progress in activity 7, as we did in the other activities. In the other assignments you solve one question/task, and think: yes, I did it, I feel accomplishment. While activity 7 has one big problem/question that never gets solved.
>
> One of the worst parts with activity 7 was that you read six different papers, where they give six different numbers on the same reservoir you want to work on. And yeah, that is annoying.
>
> [I had] a stronger feeling of accomplishment from structured activities due to a clearer sense of progress.
>
> If you get a concrete example to solve, you are able to get a sense of accomplishment. This is something I don't feel we have had in activity 7, compared to the previous ones.

As our students confirm, maintaining motivation is crucial when trying to have them cross the learning pit. Our students generally expressed an "it would have been a lot easier if I had just known this before I started the course" attitude towards learning computational practices. We cannot tell whether this is just a manifestation of being in the learning pit, or something else, and further research is needed to explore this notion. Regardless of cause, the effect is possibly detrimental to a teacher's motivation to continue the effort with active learning and computational practices.

3.3 Uncertainty and working in groups

After the final learning activity (activity 7), in which students were set to work in groups to model parts of the carbon cycle for a geological time period of their choosing, they reported having to work collaboratively in ways novel to them. The group members first had to understand and agree upon the assignment text and define the problem: geological time period and carbon reservoirs. "*It was a little hard to start. It was very difficult to agree on what to work on. But after a while, and after a lot of discussion within the group, we agreed on something we wished to work on. And after that, everything went smoothly.*" However, not all groups agreed with the above assertion of a smooth ride through the assignment. All groups were assigned at random at the beginning of the semester and remained fixed. Consequently, the individual student's experience with collaboration varied. As one student describes:

> [The] group lacks discipline and motivation to complete the assignment, which in turn resulted in nobody working and me being left alone with all responsibility. [...] I feel frustrated and resentful towards the group for this. [...] The motivation to working with the group plummeted. I don't think anyone else in the group feels this way, since they don't take responsibility or care much for the assignment.

Computers are usually operated by a single individual, and while there exist strategies for pair programming [38], collaboration in groups larger than pairs is not common in beginner-level programming. The challenge of group collaboration on computational practices, combined with the ill-structured nature of the problem they had to solve, caused a sense of urgency with getting the assignment done in time, and resulted in a “forced” distribution of tasks among its members, as two students point out:

> Previously, we've split the assignment; you take that task, I take this task, we write, read through it, and hand in. But now, we're dependent on understanding it, reading through everything, and depend upon maybe one plotting and someone else writing. So we're more dependent on each other, and it's a harder task, so... you need more than one person.
>
> You cannot write it [your part of the assignment] yourself without understanding everything else.

To streamline the working process, the student deemed by the group as the most proficient programmer would often work on the code and then explain or disseminate their progress to the remaining group members, who would then proceed with the writing. This division of work was seen as imbalanced by some students, who were concerned about the difficulties of comparing each member's effort on the assignment. This difficulty to a certain extent led to frustration and fear of free riding members that would negatively impact their grade.

Most assignments faced by our students prior to this course had been either puzzles or well-structured as defined by Jonassen [9]. Ill-structured problems, on the other hand, require decision making and argumentation. Having to make choices and provide argumentation to support these choices felt alien and discomforting to a majority of students because they lacked experience with decision making. This was pointed out by one of the older students with prior work experience:

> [Verbatim:] I thought, here we go again. This class' methodology of drastically changing the universities learning method. I liken it to pulling off a bandage that has grown in to the skin. It hurts, it's frustrating, aggravating, creates feeling of mistrust of the individual that is preforming the task, but it is necessary. I believe that the majority of the class has not reflected over that this is in fact what is happening and is only experiencing the feelings with out realising the situation in which it is being done.

However, many students reported being comforted by working collaboratively and thus being able to address and discuss uncertainty, beliefs they held, and share knowledge among peers. Crucially, several students highlighted an increasingly efficient and satisfactory learning process. It is further worth noting that the discomfort related to making choices seems strongly dependent on context for our students. A few weeks prior to the final learning activity, they participated in a field trip where they, with little prior instruction, were told to collaborate on solving a challenging field work assignment under difficult circumstances. This task was highly ill-structured, but due to experience with field work from other courses, the students expressed very little negative sentiment towards this task, and many highlighted the field trip as their favourite part of the course in the focus group interviews.

4 DISCUSSION

The uncertainty faced by the students enrolled in GEOV114 was compounded, because multiple aspects of this course were new to them. Having little or no previous experience with programming, and at the same time transitioning from their familiar role as passive recipients of knowledge to active drivers of their own learning the students were slowly becoming accustomed to a new pedagogical setting. Uncertainty is uncomfortable, and students, like anyone else, will resist exposure to it [24]. In this regard, our findings reflect those of a previous meta-study on resistance to active learning [36].

Learning programming and growing accustomed to computational practices may be likened to learning a new language or skill [16, 19, 20]; it often takes considerable time – and frustration – before the fundamentals are sufficiently operationalised to reach a feeling of flow, or competence. This initial struggle has been referred to metaphorically as the learning pit [23], and crossing it is crucial to building the student's confidence as a programmer (in the context of programming) for subsequent courses and later. The ease – or lack thereof – with which the students navigate and eventually re-emerge from the learning pit is closely connected to the teacher's ability to approach the course assignment design as a novice as well as their ability and willingness to process feedback from the students and adjust the course accordingly in a continuous process of constructive alignment [21, 22, 39]. We as teachers found it highly beneficial to accept a lack of total control over course design and simultaneously encourage TAs and students alike to actively influence their learning environment. We did this by actively including TAs in the design, alignment and assessment of each learning activity as well as encouraging constructive feedback from students informally during teaching hours as well as in the assignment hand-ins, creating a two-way formative assessment. As expected from the conclusions of [40], this feedback loop was particularly helpful when it came to weeding out ambiguity in the assignment texts, which we found was often a key factor creating uncertainty in otherwise well-structured programming tasks. Second, according to [8], the role of the expert should be to show and demonstrate and to address the students' "unknown unknowns". In our experience, expert demonstrations, in the form of guided plenary programming, had a polarising effect on the students: While some said this made them understand the coding process and outcome better, others said it just caused them to blindly transcribe our code without gaining any understanding of it.

According to Weintrop et al. [16], proficiency with programming and computational practices involves, beyond the ability to program, the capacity to choose effective computational tools, to assess different solutions to a problem, and to troubleshoot and debug code. A student who still struggles to understand code line by line will quite likely not have attained the proficiency required to operationalise these features of computational practice. The students spent much time learning how to work with the various errors and quirks of the code, which led to a loss of focus and commitment to the intended discipline-based learning outcome, for instance to plot and interpret relevant maps. Given the students' equating syllabus with list-knowledge, this derailment of the learning process felt demotivating and was perceived as clearly lowered learning outcome. In other words, the experienced lack of self-efficacy with programming caused the computational activity itself to carry the hallmarks of an ill-structured problem: The student did not even know where to start, what problem-solving technique to apply. Repeated trial and error can be frustrating, but in the context of programming there is the benefit of working systematically with errors and learning solution strategies by engaging with explicit error messages generated by the code, learning how to read them and how to effectively search for solutions. As experienced practitioners we use these same strategies because we typically don't remember specific details about syntax, which means that as teachers we had to expose our own uncertainty and demonstrate how we go about navigating our way out of the unknown.

In line with the course pedagogy, the groups were to an extent left to figure things out for themselves beyond the instructions provided by the learning activity assignment text. The discomfort initially linked to working with an ill-structured problem ended up having a predominantly positive effect within a group work context: Students reported enjoying the discussion and sharing their understanding of the problem within the group, and while there were disagreements, these led to a maturation in their understanding of group work. Our findings thus fit well with the problem-solving practices centred around the learning environment as laid out by Holder et al. [8]: Students collaborate on an ill-structured problem, using data from multiple sources, and align with the initial two stages of strategies for solving ill-structured problems listed by [9], namely constructing a problem representation and searching for solutions. Both stages rely intimately on available schema acquired from problem-solving experience [9].

Nowhere was the link between experience, and hence the availability of appropriate schema [9], and the experienced self-efficacy required to solve an ill-structured problem more pronounced than in the field. In a field setting, the students intuitively adopted some of the problem-solving practices characteristic of ill-structured problems highlighted by Holder et al. [8]; with no clear answer available, the students immediately took to the work in groups within which they defined the problem at hand, discussed new findings and adjusted the perceived result accordingly, turning to the teachers only as a last resort. The contrast between student self-reflection and feedback on the computational and field work activities tells us that the response to uncertainty, rather than uncertainty by design itself, is the crucial component to be addressed during instructional design. It furthermore underpins the necessity of creating a culture for working with ill-structured problems that must pervade the bachelor programme rather than be restricted to a few, scattered courses.

This study has strongly emphasised the role of the teacher in creating a learning environment where students can develop their computational practice and use programming to solve ill-structured earth science-related problems. We would also like to add a word of caution, based on experience, that as a teacher, one should be prepared to endure sustained frustration among students stemming simply from the fact that something is novel and not yet understood [36]. Students will try to find ways to be less frustrated, possibly leading to resistance which may range from passive, non-verbal response, via partial compliance, to open resistance [24, 27]. Luckily, recent reviews provide ample tools both for implementing active learning, such as explaining the purpose of learning activities, reviewing student feedback and continuously adjusting the course accordingly [21, 22, 26, 27], and for teaching computational practices, such as working collaboratively on project-based problems akin to learning activity seven reported herein [15-17, 34]. Opening up to iterative rounds of real-time constructive alignment based on student feedback, whatever its nature, is potentially valuable [40] but may be challenging in the midst of a course. Like others [26], however, we found it a powerful and pedagogically meaningful way of empowering the students and lessening – but by no means removing – resistance. In the words of Barnett [13; p. 131]: "*What it is to learn at the highest level, in these circumstances, is to come into an awareness – however hazy – as to the interconnectedness of matters-in-motion and to a state of* reasoned disequilibrium*, all the while unstable in a perplexing world. It is a task of higher education to help to bring about in students this state of* reasoned-and-discomforted-but-not-overly-discomforted disequilibrium."

5 ACKNOWLEDGMENTS

The authors thank Ivar Nordmo for advice on how to design the project, Rie Malm for advice on the focus group interviews, and Ingebjørg Haugland for help conducting the focus group interviews. Thanks to Isabela P. Darcie, Oddfrid T. K. Førland, Dario Blumenschein, and two anonymous reviewers for constructive criticism and comments that improved the quality of the manuscript. Serianna J. Kvarøy received funding from an iEarth seed project (WP46). Tor Einar Møller received funding from an iEarth seed project (WP47) as well as from Olsens Legat at the University of Bergen.

REFERENCES

1. Masson-Delmotte, V., et al., *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. 2021.
2. United Nations, G.A., *Transforming our world: The 2030 agenda for sustainable development*. 2015.
3. Manduca, C.A., D.W. Mogk, and et al., *Earth and mind: How geologists think and learn about the earth*. Vol. 413. 2006.
4. Assaraf, O.B.Z. and N. Orion, *Development of system thinking skills in the context of earth system education.* Journal of Research in Science Teaching, 2005. 42(5): p. 518–560. https://doi.org/10.1002/tea.20061
5. Raia, F., *Causality in complex dynamic systems: A challenge in earth systems science education.* Journal of Geoscience Education, 2008. 56(1): p. 81–94. https://doi.org/10.5408/1089-9995-56.1.81
6. Orion, N. and J. Libarkin, *Earth system science education*, in *Handbook of Research on Science Education, Volume II*. 2014. p. 495–510.
7. Scherer, H.H., L. Holder, and B. Herbert, *Student Learning of Complex Earth Systems: Conceptual Frameworks of Earth Systems and Instructional Design.* Journal of Geoscience Education, 2017. 65(4): p. 473–489. https://doi.org/10.5408/16-208.1

8. Holder, L.N., H.H. Scherer, and B.E. Herbert, *Student Learning of Complex Earth Systems: A Model to Guide Development of Student Expertise in Problem-Solving.* Journal of Geoscience Education, 2017. 65(4): p. 490–505. https://doi.org/10.5408/17-261.1
9. Jonassen, D.H., *Instructional design models for well-structured and ill-structured problem-solving learning outcomes.* Educational technology research and development, 1997. 45(1): p. 65–94. https://doi.org/10.1007/BF02299613
10. Tissenbaum, M., et al., *The case for alternative endpoints in computing education.* British Journal of Educational Technology, 2021. 52(3): p. 1164-1177. https://doi.org/10.1111/bjet.13072
11. Barnett, R. and K. Coate, *Engaging the curriculum in higher education*. 2004: McGraw-Hill Education.
12. Barnett, R., *The ecological university: A feasible utopia*. 2018: Routledge. https://doi.org/10.4324/9781315194899
13. Barnett, R., *The philosophy of higher education: A critical introduction*. 2022: Routledge. https://doi.org/https://www.taylorfrancis.com/books/mono/10.4324/9781003102939/philosophy-higher-education-ronald-barnett
14. Biesta, G.J.J., *The Beautiful Risk of Education*. 2013: Paradigm Publishers. 165.
15. Medeiros, R.P., G.L. Ramalho, and T.P. Falcão, *A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education.* IEEE Transactions on Education, 2019. 62(2): p. 77-90. https://doi.org/10.1109/TE.2018.2864133
16. Weintrop, D., et al., *Defining Computational Thinking for Mathematics and Science Classrooms.* Journal of Science Education and Technology, 2016. 25(1): p. 127–147. https://doi.org/10.1007/s10956-015-9581-5
17. Hsu, T.C., S.C. Chang, and Y.T. Hung, *How to learn and how to teach computational thinking: Suggestions based on a review of the literature.* Computers & Education, 2018. 126: p. 296-310. https://doi.org/10.1016/j.compedu.2018.07.004
18. Scherer, R., F. Siddiq, and B.S. Viveros, *The cognitive benefits of learning computer programming: A meta-analysis of transfer effects.* Journal of Educational Psychology, 2019. 111(5): p. 764–792. https://doi.org/10.1037/edu0000314
19. Yuen, T.T. and K.A. Robbins, *A qualitative study of students' computational thinking skills in a data-driven computing class.* ACM Transactions on Computing Education, 2014. 14(4): p. 1–19. https://doi.org/10.1145/2676660
20. Loftsson, H. and A. Matthíasdóttir, *Using Flipped Classroom and Team-Based Learning in a First-Semester Programming Course: An Experience Report*, in *IEEE International Conference on Engineering, Technology and Education (TALE)*. 2019. p. 1-6. https://doi.org/10.1109/TALE48000.2019.9225985.
21. Nguyen, K.A., et al., *Instructor strategies to aid implementation of active learning: a systematic literature review.* International Journal of Stem Education, 2021. 8(1). https://doi.org/10.1186/s40594-021-00270-7
22. Tharayil, S., et al., *Strategies to mitigate student resistance to active learning.* International Journal of Stem Education, 2018. 5. https://doi.org/10.1186/s40594-018-0102-y
23. Nottingham, J., *The learning challenge : how to guide your students through the learning pit to achieve deeper understanding*. Challenging learning series. 2017, Thousand Oaks, California: Corwin, a SAGE publishing company., 251 pages.
24. Weimer, M. and I. NetLibrary, *Learner-centered teaching: five key changes to practice*. 1st ed. Jossey-Bass higher and adult education series. 2002, San Francisco: Jossey-Bass.
25. Winkler, I. and I. Rybnikova, *Student resistance in the classroom-Functional-instrumentalist, critical-emancipatory and critical-functional conceptualisations.* Higher Education Quarterly, 2019. 73(4): p. 521-538. https://doi.org/10.1111/hequ.12219
26. Andrews, M., et al., *Explanation and Facilitation Strategies Reduce Student Resistance to Active Learning.* College Teaching, 2021. ahead-of-print: p. 1-11. https://doi.org/10.1080/87567555.2021.1987183
27. Finelli, C.J. and M. Borrego, *Evidence-based strategies to reduce student resistance to active learning.* , in *Active learning in college science* J.J. Mintzes and E.M. Walter, Editors. 2020, Springer Cham. p. 943-952. https://doi.org/10.1007/978-3-030-33600-4_58

28. Corbin, J.M. and A.L. Strauss, *Basics of qualitative research : techniques and procedures for developing grounded theory*. Fourth edition. ed. 2015, Los Angeles: SAGE., 431 pages.
29. Yin, R.K., *Qualitative research from start to finish*. 2011, Guilford: New York; London.
30. DeJarnette, A.F., et al., *Students' Use of Computational Thinking Practices in an Undergraduate Biology-Engineering Course.* Journal for STEM Education Research, 2022. 5(1): p. 53-77. https://doi.org/10.1007/s41979-021-00058-y
31. Leavy, P., *The Oxford handbook of qualitative research*. Paperback edition. ed. Oxford library of psychology. 2015, New York: Oxford University Press., 756 pages.
32. Reid, N., Learning, and N. Teaching Support, *Getting started in pedagogical research in the physical sciences*. LTSN physical sciences practice guide. 2003, Hull, England: LTSN Physical Sciences Centre. 70 pages.
33. Glaser, B.G. and A.L. Strauss, *The discovery of grounded theory; strategies for qualitative research*. Observations. 1967, Chicago,: Aldine Pub. Co., 271 p.
34. Tang, X.D., et al., *Assessing computational thinking: A systematic review of empirical studies.* Computers & Education, 2020. 148. https://doi.org/10.1016/j.compedu.2019.103798
35. Black, P. and D. Wiliam, *Assessment and classroom learning. .* Assessment in Education: principles, policy & practice, 1998. 5(1): p. 7-74. https://doi.org/10.1080/0969595980050102
36. Shekhar, P., et al., *Negative student response to active learning in STEM classrooms: A systematic review of underlying reasons.* Journal of College Science Teaching, 2020. 49(6): p. 45--54.
37. Deslauriers, L., et al., *Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom.* Proceedings of the National Academy of Sciences of the United States of America, 2019. https://doi.org/10.1073/pnas.1821936116
38. Hannay, J.E., et al., *The effectiveness of pair programming: A meta-analysis.* Information and Software Technology, 2009. 51(7): p. 1110-1122. https://doi.org/10.1016/j.infsof.2009.02.001
39. Biggs, J.B. and C. Tang, *Teaching for quality learning at university*. 4 ed. 2011: Open University Press.
40. Hattie, J. and H. Timperley, *The power of feedback.* Review of Educational Research, 2007. 77(1): p. 81-112. https://doi.org/10.3102/003465430298487