Computational Thinking in Secondary Mathematics Education using GeoGebra: Results from a Design Study

Christos Chytas[a&b]*, Sylvia Patricia van Borkulo[a]*, Paul Drijvers[a]*, Erik Barendsen[b&c]* & Jos Tolboom[d]*

[a]*Freudenthal Instituut, Utrecht University, Utrecht, Netherlands;* [b]*Faculty of Science, Radboud University, Nijmegen, Netherlands;* [c]*Department of Computer Science, Open University, Heerlen, Netherlands;* [d]*SLO, Netherlands Institute for Curriculum Development, Groningen, Netherlands*

Contact Christos Chytas, c.chytas@uu.nl

Computational Thinking in High School Mathematics Lessons Using GeoGebra: Results from a Design Study

Nowadays, mathematics teachers in K-12 strive to promote their students' mathematical knowledge and computational thinking (CT) skills. There is an increasing need for effective CT-embedded mathematics learning material and a better understanding of students' perceptions toward them. In this work, we present the results of a design study, which included the design of a six-lesson learning activity aimed at fostering 16-to-17-year-old secondary students' CT skills in calculus lessons using the dynamic mathematics software GeoGebra. We collected and analyzed data from students' code in GeoGebra, workbooks, semi-structured interviews, and questionnaires. Our findings suggest that most students mastered using CT concepts in calculus activities to a satisfactory degree and could reason about their computational solutions using GeoGebra and the generated graphs. Overcoming mathematics content knowledge gaps was essential for students to complete the lesson series successfully. Our study supports that students appreciate GeoGebra's CT-embedded approach to calculus lessons and its more exploratory character to mathematics problems when provided with appropriate support. We conclude that an integrated approach to mathematics education and computational thinking is viable and might contribute not only to fostering CT but also to increasing interest in mathematics.

Keywords: calculus education; computational thinking; GeoGebra; mathematics education; mathematics software

Introduction

There is a widespread and growing agreement among academics and educators that computational thinking (CT) should be taught to everyone and that it is an important part of scientific literacy. In her impactful article, Jeannette Wing (2006) argues that CT should be fostered as a basic literacy skill like "reading, writing and arithmetic" (p. 33). The popularization of Wing's article led to global efforts to embed CT in school and out-of-school learning environments. Moreover, there are increasing calls to promote CT in computing and other STEAM (Science, Technology, Engineering, the Arts, Mathematics) subjects, including humanities and arts (Perković et al., 2010).

As powerful computational tools become commonplace in mathematics and science, there are promising opportunities for fostering CT in non-programming tasks and unplugged learning activities (Dagienė & Sentance, 2016). Modern mathematics tools intended for education offer opportunities for fostering CT within mathematics learning activities aligned with current curricula and educational policies. Moreover, they provide fruitful visualization and interactivity opportunities for learning (Adelabu et al., 2019). The use of educational technology to foster CT dates decades back to the vision of progressive educators for using powerful programming tools that are accessible to a wide variety of students (Papert, 1980). According to Wilensky and colleagues (2014), technological innovations in education allow students to explore mathematical concepts and "create" mathematics themselves.

Moreover, integrating CT into STEAM subjects is inclusive and provides a more realistic representation of the domains of science and mathematics as potential future career options. For this reason, many researchers see CT as a way to increase participation in computer science and integrate computing into different disciplines (Weintrop et al., 2016). Especially when students in many countries do not get the opportunity to engage in computational problem-solving because computer science is only an elective subject.

However, there is little research on how secondary students perceive CT-embedded learning activities and how these can be integrated into calculus classes. For this reason, integrating CT into mathematics lessons is a challenging task that requires designing, testing, and refining the developed material before being implemented into curricula.

This work presents the results from a design study on a CT-embedded calculus lesson series (van Borkulo et al., 2021). The lesson series took place in mandatory mathematics lessons for Calculus (referred to as Mathematics B in the Dutch educational system). Mathematics B deals with more theoretical aspects of mathematics, especially algebra and geometry. The subject is particularly suitable for students considering studying in scientific fields, enabling them to apply for more STEM-oriented study programs in tertiary education.

Our study used a mixed-methods approach to answer the following research question:

RQ: How do students perceive the integration of CT into calculus lessons with GeoGebra and what challenges do they face in successfully completing CT-embedded assignments?

To tackle the research question at hand, we examine the feasibility of implementing CT-embedded calculus lessons in youth's formal education. Furthermore, we contribute to existing efforts in developing computationally rich learning experiences that allow educators to bring CT into calculus lessons using accessible yet powerful tools like GeoGebra.

Theoretical Background

CT skills are now widely considered essential for everyone. Educators in K-12 strive to foster their students' CT skills and digital literacies to prepare them for professional life and participation in society. According to Wing, CT involves aspects fundamental to computer science, but it is not a skill explicitly targeted at technical experts:

> Computational thinking is a fundamental skill for everyone, not just for computer scientists.... involves solving problems, designing systems, and

> understanding human behavior, by drawing on the concepts fundamental to computer science (Wing, 2006, p. 33).

CT research is growing fast, but there is no consensus on what CT actually is. In later work, Wing (2011) phrased the definition as follows:

> Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer-human or machine—can effectively carry out. (Wing, 2011, p.20)

Early research on CT focused on elements such as abstraction, decomposition, pattern recognition, algorithmic thinking, and generalization (Selby & Woollard, 2010). Besides the different definitions of CT in academic literature, it is widely accepted that CT involves processes fundamental to computer science and computational problem-solving such as making abstractions of the most critical elements of a problem (Wing, 2017), algorithmic thinking (Futschek, 2006) which focuses on understanding and designing algorithms to solve problems, decomposing problems into smaller, more manageable parts (Rich et al., 2019), and pattern recognition to identify common characteristics between problems (Wing, 2006). For more than a decade, CT research has been evolving, and researchers address additional CT aspects like automation (Lee et al., 2014) and modularization (Atmatzidou & Demetriadis, 2016).

Another way of CT classification is included in the CT assessment framework of Brennan and Resnick (2012). The authors classified CT aspects in terms of computational concepts (e.g., loops), computational practices (e.g., testing and debugging), and computational perspectives (e.g., how learners connect computational activities with their lives). These aspects are fundamental in computing and especially programming.

Weintrop et al. (2016) argued about the significance of CT in mathematics and science and developed a taxonomy of CT practices in these fields. These practices include data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices. Additionally, there is consensus that CT is an integral part of the STEM disciplines (Henderson et al., 2007) and integrating computing and CT into such subjects in K-12 is a natural fit. Science and mathematics are becoming increasingly computational, and CT (Wing, 2017) is now considered a scientific practice in education (Weintrop et al., 2016).

Fostering CT in K-12 is an efficient way to prepare future scientists and responsible citizens in an increasingly computational world. Integrating CT into science classes, e.g., by using tools for agent-based modeling in science education, could equip more students with CT skills, considering that computer science is still an elective course for many students in secondary education (Wilensky et al., 2014). In addition, there is consensus that CT is an integral part of the STEM disciplines (Henderson et al., 2007)

Progressive ideas for including computing in education have a long history, dating back to the 60s and 80s when Alan Perlis argued that it is essential to introduce students of all disciplines to the theory of computation (Guzdial, 2008). In addition, Seymour Papert envisioned children and youth developing procedural thinking skills and learning programming and mathematics with Logo (Papert, 1980). Integrating CT into STEM subjects is often linked to higher learning gains in the respective subject. For example, a study in CT-embedded mathematics lessons for 6th graders by (Calao et al., 2015) showed that integrating CT into mathematics lessons can lead to statistically significant increases in the learning outcomes of students in terms of mathematical processes such as modeling, problem formulation, and problem-solving and reasoning among other processes. In general, educational mathematics software is linked with higher learning gains in mathematics and fostering mathematical and computational thinking (van Borkulo et al., 2021).

A study on the effectiveness of the educational mathematics software GeoGebra showed that high school students achieved better learning outcomes and welcomed its use which broadened their perspectives about mathematics learning (Arbain & Shukor, 2015). According to the National Council of Teachers of Mathematics, technology plays a key role in modern mathematics education:

> An excellent mathematics program integrates the use of mathematical tools and technology as essential resources to help students learn and make sense of mathematical ideas, reason mathematically, and communicate their mathematical thinking (Brahier et al., 2014, p.517)

Digital tools are becoming commonplace in K-12 as vehicles to explore and verify mathematical ideas. Moreover, visualization and interactive elements can enhance learning and provide excellent opportunities to promote mathematical thinking (Drijvers, 2018) and CT (van Borkulo et al., 2021). Digital tools for CT-embedded mathematics learning activities in different grades include programming languages like Python (Jenkins et al., 2012), programming languages that are more common in education like Scratch (Calao et al., 2015), Logo-based tools like MaLT (Kynigos & Grizioti, 2018), and spreadsheets (Sanford & Naidu, 2016), among other tools.

In line with the educational practice discussed above, in this study, we focus on fostering pre-university students' CT skills using accessible computational tools that many mathematics teachers in the Netherlands are already familiar with. In previous work (van Borkulo et al., 2021), we argued about the potential of dynamic mathematics software in fostering pre-university students' CT skills, especially algorithmic thinking and generalization in calculus lessons. In this study, we aim to extend the previous findings and capture more CT aspects (including decomposition, pattern recognition, abstraction, and algorithmic thinking/algorithm design) in calculus lessons with GeoGebra. Our work will hopefully be of value to educators and mathematics teachers that wish to address CT aspects in calculus lessons. We provide concrete examples that illustrate students' implementation of CT skills in CT-embedded calculus lessons and identify commonly encountered problems in such settings, building on previous research.

Methods

We designed and implemented CT-embedded calculus learning activities in four classes of 16-to-17-year-old secondary students following an educational design research approach (Gravemeijer & Cobb, 2006; Yazan, 2015) that focuses on designing and evaluating CT-embedded calculus activities. Below, we present the educational context and provide information about the study's participants. Then, we describe the learning activity, the materials/tools used by the student participants, and our design rationale. Finally, we discuss the data collection and analysis procedures.

Study Context

For three of the four classes, the intervention took place in a physical classroom setting at three schools in the Netherlands. The intervention took place online for the fourth class to prevent the spread of covid-19. The intervention that took place online consisted of five 45-minute calculus lessons (including post-experiment activities, e.g., interviews and filling in questionnaires), and the interventions that took place physically consisted of six lessons of 50 minutes.

The lesson series took place in mandatory mathematics lessons for Calculus (referred to as Mathematics B in the Dutch educational system), and the students could work alone or in pairs.

Educational Context

Mathematics B

There are different levels of secondary mathematics education in the Netherlands: A, B, and C. Mathematics B (or Wiskunde B) is addressed to pre-university students in the VWO (Voorbereidend Wetenschappelijk Onderwijs) education and prepares them for studying in scientific fields while enabling them to apply for more STEM-oriented study programs in tertiary education. It includes topics like algebra, geometry, trigonometry, and calculus. The subject is particularly suitable for students considering studying in scientific fields, enabling them to apply for more STEM-oriented study programs in tertiary education. It is expected that students have good knowledge of algebra, geometry, and trigonometry. The course also includes more advanced content like functions, derivatives, integrals, and differential equations.

Description of the Lesson Series

The lesson series was developed as part of a larger project on introducing computational thinking in mathematics education. The calculus content covered in the lesson series focused on the perpendicular bisector, focal points, tangents, and zeros of functions, among other topics. The students were expected to use CT concepts (e.g., variables, conditional statements, and iterations) and practices (e.g., debugging, testing, and evaluating) to find general solutions to calculus problems using the GeoGebra environment. The assignments can be found in the respective link in the Appendix.

The research team supported the teachers during the co-design and implementation of the lesson series in the Mathematics B classroom. GeoGebra is a popular software program in Dutch education, and the teachers in our study were already familiar with it, having used it previously in their lessons. Teachers' prior experience with GeoGebra and GeoGebra being accessible (free, open-source) and providing visualization and interactivity possibilities were determinant factors in choosing this tool.

Digital tools: GeoGebra

GeoGebra can be downloaded and used as software or directly used online on its official website (see footnote 2). It is an accessible and open-source mathematics tool which makes it an attractive choice for teaching calculus, geometry, and algebra (among other subjects) in school and out-of-school settings in primary, secondary, and tertiary education. GeoGebra allows students to generate points, lines, segments, and vectors, among other mathematical representations. Such mathematical representations can be dynamically altered afterward. GeoGebra enables the use of variables for storing mathematical objects (e.g., numbers, points, line segments) that can also be modified via buttons and input fields. Furthermore, it is also possible to use GeoGebra to introduce computational concepts in computing education, e.g., by using iteration lists and conditional statements. GeoGebra is gaining increasing popularity among mathematics teachers in the Netherlands, which made it an appealing choice for our research. GeoGebra's advantages are its interactive graph generation capabilities, which allow learners to explore mathematics concepts through visualization (Adelabu, 2019).

Learning Materials

Workbooks

The learning activity included using a workbook that we developed [(link to the workbook)](#) and addresses CT and mathematics content using GeoGebra as described in Table 1, which presents the mathematics content and CT focus of the learning activity with brief examples.

The successful completion of the developed assignments required both mathematical and computational thinking. Employing abstraction, decomposition, algorithmic thinking, and generalization skills was key to efficiently completing our designed tasks.

Table 1.

Mathematics content and computational focus in the respective chapters

Chapter	**Mathematics content**	**Computational focus**	**Example**
Chapter 1	Line through two given points	Variables, conditional statements, mathematics operations	Calculation of the slope of a line through points A and B
Chapter 2	Perpendicular bisector	Variables, conditional statements, mathematics operations	Considering the case that the line might be vertical, i.e., points A and B are directly above each other
Chapter 3	Centre of gravity	Variables, conditional statements, mathematics operations	Determine the center of gravity of a general triangle equation considering the exception cases
Chapter 4	Tangent to a parabola	Variables, conditional statements, mathematics, and logic operators	Create a general parabola equation and testing the solution by creating a tangent
Chapter 5	Bundles of tangents to a parabola	Variables, conditional	Using iteration to generate a bundle

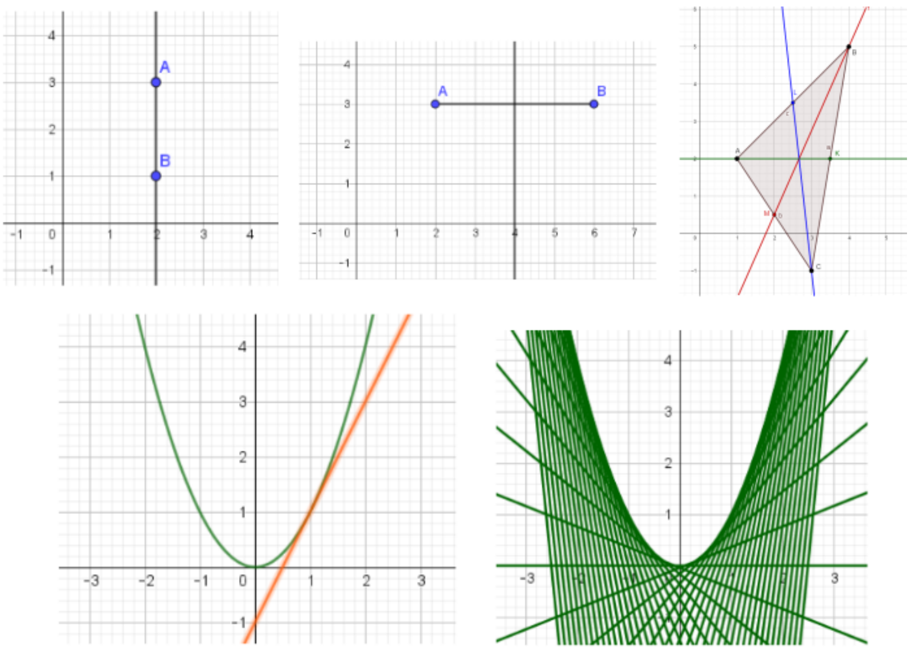
		statements, iterations, mathematics and logic operators	of tangents to specific and general parabola equations
Chapter 6	Tangents to various graphs	Variables, conditional statements, iterations, mathematics and logic operators	Computational experimentation for creating a bundle of tangents to different graphs (e.g., a root function and free choice graphs)
Chapter 7	Final task (not mandatory) - Newton Raphson method	Variables, conditional statements, iterations, mathematics and logic operators	Implementation of a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function.

The workbooks with the assignments served as a scaffolding tool for tackling the CT-embedded calculus problems that the students were asked to solve. The workbooks also included a short introduction to the GeoGebra environment and hints for tackling the problems.

Chapters 1-4 focused on using conditional statements, and chapters 5-7 focused on combining conditional statements with iterations. To generate the intended graphs, the students needed to be precise and fully understand the concept of iterations and conditional statements. In chapter 6, the students could computationally experiment with their own functions and find general solutions to the calculus problems they chose. The seventh chapter about the Newton-Raphson method was the most challenging task. It required using iterations/macros and a deep understanding of the aforementioned method to calculate the zeros of functions. The students were encouraged to write a report about this method where they explained how it works, but it was not mandatory.

Each chapter involved generalizing from a specific case to a more general one using parameters/variables and conditional statements. An example would be creating a general solution in which the equation changes when students drag points A and B. Therefore, students should consider the special cases and use appropriate conditional statements to generate the graphs and possibly evaluate them by dragging the points and testing their equations.

Figure 1. Students' GeoGebra generated interactive graphs for solving tasks in the lesson series material: line through two given points, perpendicular bisector, center of gravity, tangent to a parabola and bundles of tangents to a parabola



Design Rationale

We co-designed these assignments with the teachers and experts in computer science and mathematics education to address specific CT aspects, as illustrated in Table 1.

Table 1.

Mathematics content and computational focus in the respective chapters

Chapter	Mathematics content	Computational focus	Example
Chapter 1	Line through two given points	Variables, conditional statements, mathematics operations	Calculation of the slope of a line through points A and B
Chapter 2	Perpendicular bisector	Variables, conditional statements, mathematics operations	Considering the case that the line might be vertical, i.e., points A and B are directly above each other
Chapter 3	Centre of gravity	Variables, conditional statements, mathematics operations	Determine the center of gravity of a general triangle equation considering the exception cases
Chapter 4	Tangent to a parabola	Variables, conditional statements, mathematics, and logic operators	Create a general parabola equation and testing the solution by creating a tangent
Chapter 5	Bundles of tangents to a parabola	Variables, conditional statements, iterations, mathematics and logic operators	Using iteration to generate a bundle of tangents to specific and general parabola equations
Chapter 6	Tangents to various graphs	Variables, conditional statements, iterations, mathematics and logic operators	Computational experimentation for creating a bundle of tangents to different graphs (e.g., a root function and free choice graphs)
Chapter 7	Final task (not mandatory) - Newton Raphson method	Variables, conditional statements, iterations, mathematics and logic operators	Implementation of a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function.

The design of the lesson series focuses on decomposition, pattern recognition, abstraction, and algorithmic thinking, which are universally accepted core CT elements across the academic literature (Dong et al., 2019; Kynigos & Grizioti, 2018). Table 2 describes the CT skills required in the lesson series and the intended student behavior.

Table 2.

Computational Thinking Aspects in Focus and Intended Student Behaviours

CT skill	Description of CT skill	Intended student behaviour
Decomposition	Breaking down a problem into smaller parts	Calculating the sub-parts A and B in the standard form for linear equations in two variables: Ax+By=C
Pattern recognition	Pattern recognition involves observing and analyzing data and situations to identify patterns that can be critical in solving similar problems	Identifying graphical and coding patterns that can be reused (e.g., re-using parts of code from previous tasks)
Abstraction	Abstraction involves ignoring unimportant information to focus on what is important for solving a specific problem	Separate important from redundant information in specific assignments
Algorithmic thinking/ design	Algorithmic thinking and algorithm design involve understanding and creating algorithms to solve a problem or complete a task in a way that others would achieve the same result if they follow the exact steps.	Using the basic logic structures (sequences, selections, and loops), e.g., using an iteration list for generating 50 tangent lines instead of writing code 50 times that generates 50 tangents for each case 50 times)

In the developed lesson series, the students are called to address the assignments in the workbooks and translate them into computational solutions in the GeoGebra environment. Most parts of decomposition are already offered as a scaffold so that the students can focus on understanding mathematics content and using the logical structures of sequence, selection and loop to solve the computational problems of the lesson series. The students must break equations apart by calculating the slope or other subparts of equations, decomposing iteration lists, formulating conditional statements, and adjusting variables/parameters to consider special cases when creating general solutions.

The students are called to use pattern recognition to identify, reuse and generate the intended graphs in the GeoGebra environments. The workbooks also provide hints for considering the special cases in the first chapters.

The assignments ask students to use algorithm thinking/design skills to translate the calculus assignments into computational solutions in the GeoGebra environment. The computational solutions require using the basic logical structures (sequence, selection and loop), which are integral elements to solving every algorithmic problem. The logical structure of sequence can be used to translate the steps of the workbooks into computational steps in GeoGebra. The logical structure of loop can be used to generate iterative graphs (Chapters 5-7) like a tangent bundle in the respective assignments. The selection structure can be used to consider the special cases of equations that can lead to general calculus solutions (e.g., considering if the fraction's denominator in an equation is zero). Considering such special cases requires crucial elements of algorithmic thinking and also generalization.

Participants

Our study participants were 52 twelfth-grade secondary students who were 16-17 years old. All but one student had no prior experience using GeoGebra, even though some students had seen their teacher generating graphs to help them visualize mathematical concepts. We informed the students about the study's aims and asked for their consent to use their data to evaluate their learning experience. We ensured students that we would handle their data anonymously and respect their privacy, and we took ethical considerations into account to ensure good research practices for underage populations.

Data Collection

Workbooks and GeoGebra Files

To better understand the feasibility of integrating CT-embedded learning activities into calculus lessons, we examined the workbooks and GeoGebra files of students during both the plugged (working on GeoGebra) and unplugged (working on workbooks) phases of the learning activity. Furthermore, evaluating CT-embedded assignments allows us to identify mistakes and potential misconceptions in students' work. At the end of the lesson series we collected all workbooks and GeoGebra files from students. We communicated to the students in advance and provided clear instructions on the procedure for returning their workbooks and uploading their GeoGebra files. Regarding the upload of files to our online repository, students had the option to upload individual files after each lesson or to submit all of their files at the end of the series. We have provided clear guidelines for both options and were available to assist with any questions or technical issues that may arise. The first two authors checked and catalogued each submission to ensure all workbooks and files were accounted for. However, students could choose not to share their files with us. Their decision was completely voluntary and did not impact their participation in any way.

Interviews

Qualitative methods are an appropriate choice for examining complex and sophisticated thinking skills like CT in a comprehensive way. To better understand students' learning experiences with the designed learning activities, as well as which kind of strategies they employed to tackle the computational problems at hand, we conducted one-on-one semi-structured artifact-based interviews (Brennan & Resnick, 2012) with them. The workbooks and GGB files were the foundation for reflection on the developed activity during the interviews. The final part of the conducted interviews included questions regarding students' reflections on their experience with GeoGebra and how they perceived the lesson series.

In total, 25 students participated in the interviews, which took place at the end of the lesson series. Each interview lasted between 10 to 25 minutes. During the semi-structured interviews, we asked students about their 1) opinions on the learning activities, 2) the difficulties they encountered, 3) how they overcame them, 4) how they implemented CT concepts and practices, and 5) their experience with CT-embedded calculus lessons.

Data Analysis

Workbooks and GeoGebra files

We evaluated the successful completion of the CT-embedded assignments on three levels similar to previous evaluations of computationally rich learning activities (Chytas et al., 2018; van Borkulo et al., 2021): “Incorrect” for assignments in which less than half of the tasks were correctly completed, “Partially correct” for assignments in which half of the tasks were correctly completed, and “Correct” for assignments without errors. Some students did not save their GeoGebra files or chose not to upload them for our data analysis. Therefore, we coded them as “absent”. The seventh chapter was not mandatory for the students, but only 12 files of the students/groups were absent.

For the assignments that were evaluated as incomplete and partially complete, we also included notes regarding the mistake of the students, which we later coded inductively. The coding did not aim at evaluating the learning outcomes but at identifying mistakes, potential misconceptions, and difficulties when using GeoGebra in CT-embedded calculus problems.

Interviews

The semi-structured interviews were analyzed using inductive and deductive coding approaches. Our preliminary set of deductive included codes from previous work (van Borkulo et al., 2021) focusing on participants' previous experience, perceptions of the learning activities, encountered problems, and computational thinking aspects, e.g., AT and strategies to tackle computational problems in GeoGebra.

Wherever possible, we triangulated the data from different sources to provide robust evidence of our findings. We used data triangulation and investigator triangulation (Carter et al., 2014) in iterative circles of design and analysis. The researchers worked in close cooperation and met weekly to compare codes and themes that emerged from the collected data.

Results

In this section, we present the study's findings with insights into how students perceive the integration of computational thinking into calculus lessons with GeoGebra and what challenges they face in successfully completing the developed assignments. In the next sections, we take a closer look at the respective data sources one by one to provide a more comprehensible picture of the results.

Findings from the analysis of workbooks and GeoGebra files

According to the teachers who implemented the lesson series in their classes, the learning outcomes of the lesson series were satisfactory. This is also supported by the analysis of workbooks and GeoGebra files which shows that the students started getting more familiar with GeoGebra and computational problem-solving. Even though the difficulty of the assignments was gradually increasing, the students' mistakes significantly decreased in the later chapters (4-7).

Figure 2. Workbooks analysis results (physical class)

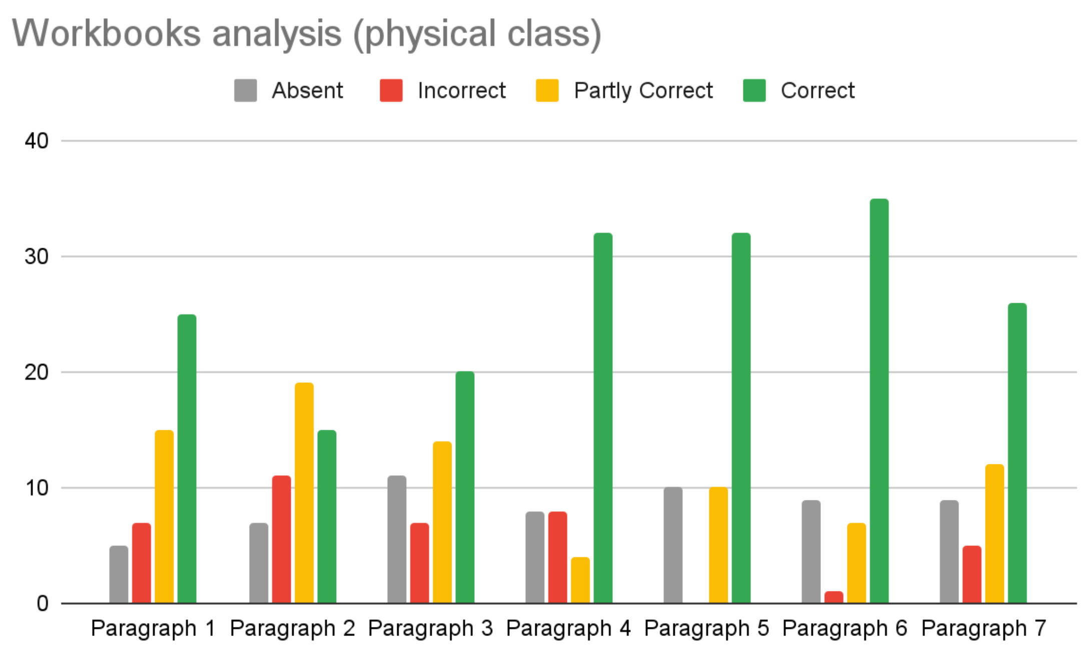
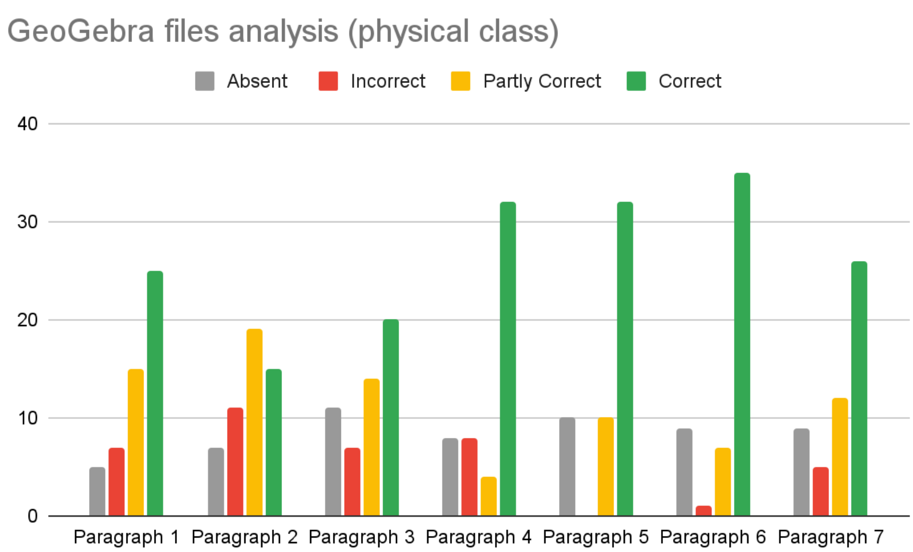


Figure 3. GeoGebra files analysis results (physical class)



The most common mistakes in the workbooks were related to generalization and the students not creating a general solution for the assignment by using variables and conditional statements to consider the special cases. Very often, the students used the wrong formula for equations which led to wrong solutions. Considering that the problems at hand provided space for students to test their solutions, there are indications of students' lack of mathematical content knowledge and misconceptions. Therefore gaps in mathematical content knowledge could become barriers to moving to more sophisticated computational endeavors with GeoGebra. This is also illustrated in Figures 4 and 5. The most common mistakes in the workbooks were related to mistakes in creating the general and specific solutions to the problems. Miscalculations of formulas and wrong use of computational concepts were also commonly found mistakes (using wrong conditional statements for considering special cases, and using wrong conditions for the iteration lists (conditionional loops). These could be related to students' lack of understanding of mathematical concepts being applied, e.g., the properties of a parabola. Another possible reason could be that students did not have adequate time to get used to using computational concepts like iterations and conditional statements, as later on the the GeoGebra files code it seems that they developed a better understanding of it. Finally, some students might have lacked attention to the details and did not evaluate their solution as long as the generated graph in GeoGebra seemed functionable at a first look.

Figure 4. Students' mistakes in workbooks

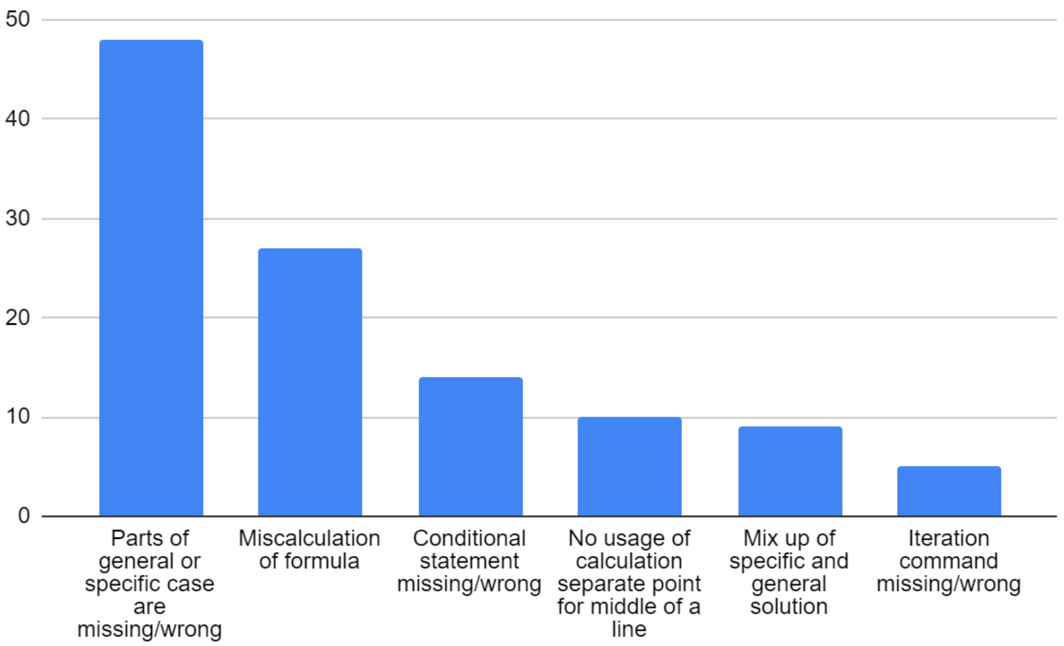
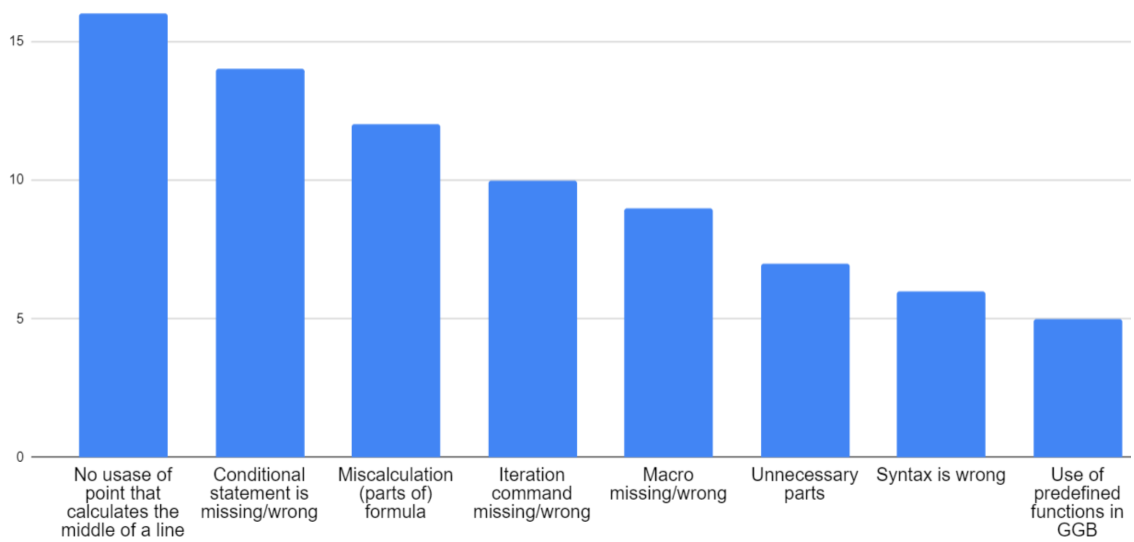


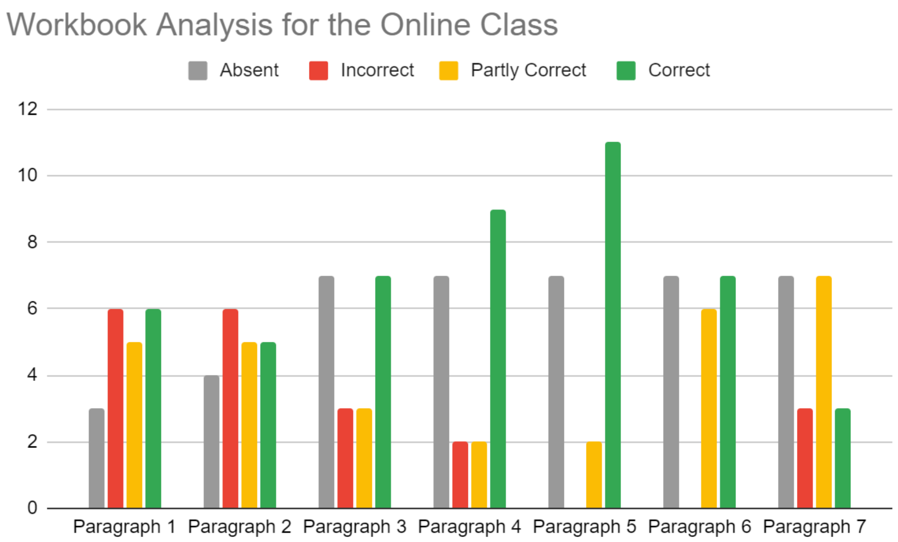
Figure 5. Students' mistakes in GeoGebra files

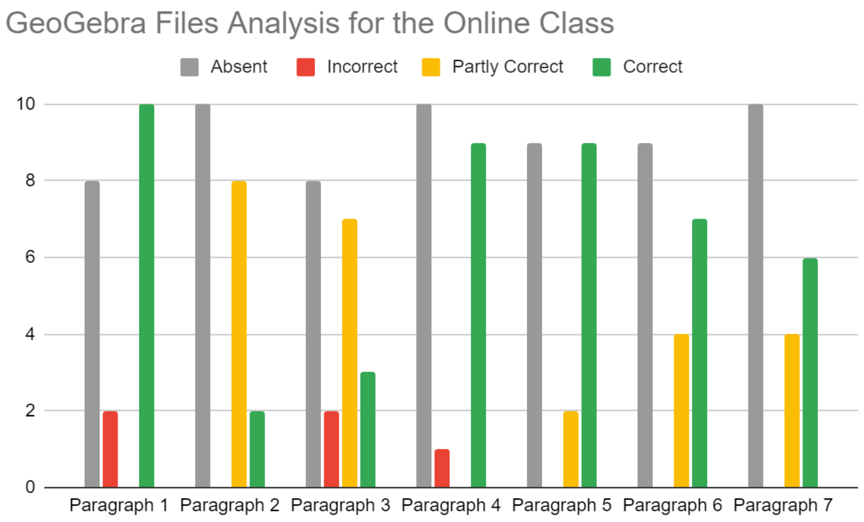


The students in the online class encountered more issues than in the classes that took place at school. The rates for absent files were significantly higher than in the classes that took place physically, and the students seemed more frustrated when working on GeoGebra. This finding was anticipated because the lesson series was shorter, and the teacher of the online class created online rooms for every student or pair of students.

Furthermore, the teacher could not efficiently address all arising questions besides the ones in the first and last 5 minutes of each lesson when all students were in the same online room. The students of the online class who engaged in the lesson series made similar mistakes with the students in the physical class.

Figure 6. Workbooks and Geogebra files analysis (online classes)





Findings from artifact-based interviews

We briefly focus on key themes that emerged and were related to encountering difficulties and the overall experience of students.

Encountered problems

In addition to the student mistakes identified in the workbooks and GeoGebra file analysis, students reported problems they encountered during the interviews.

Getting familiar with GeoGebra

Students faced initial difficulties in using GeoGebra which led to frustration and discouragement with using the software.

> "It's more the program GeoGebra... Bit of trouble with that to fill in everything."

The result of the students' experience with GeoGebra could be a mix of both competence and inefficiencies in using the software for certain tasks. A particular frustration of students when first used the software was that they had troubles with mathematical content that would easily solve with pen and paper (e.g., successfully drawing a tangent in GeoGebra) and took longer times to create with GeoGebra.

> "Yes, for example, drawing up a tangent or something, that's not very difficult. Sometimes it did take a long time to draw up a tangent and you could actually do that very quickly, but otherwise it was just necessary."

However, after multiple attempts students got used to the software, something that is also captured in the GeoGebra files analysis (despite the assignments' increasing difficulty, the students made fewer mistakes in the later chapters)

> "Well in the beginning, I sometimes didn't know how to fill something in GeoGebra, but after a few tries it just worked. And the steps were just clear in that booklet"

Clear instructions and clarity in using GeoGebra in additional resources such as the workbooks was beneficial for the students who grew confidence and competence in using the software.

Syntax

Many students communicated with us the challenges of using GeoGebra and the difficulty they had filling in information correctly. In certain cases, students showed frustration with with receiving error messages and felt unsure about what they were doing because of the interface and syntax of the GeoGebra environment which required time to get used to.

> "Well, I do find it difficult. Yes, I don't know... I do have trouble filling in things and also when I fill it in, I get one of those triangles saying it's wrong and then I get a bit mad, I think 'well, why not? What am I doing wrong again? Because of course you have to fill in exactly what the computer wants."

> "Yeah I wasn't very specific but then in GeoGebra you need to be specific so I think that that also changed a little bit if I need to do that exercise again, I will be maybe more specific with every little thing because normally you just think 'oh that's kind of easy' say 'it is normal, you forget that'... But in GeoGebra you need to be... It doesn't work that way. You need to be very specific and that also... I think if that... In the first exercise I really didn't do that actually I think. "

Students had to to adapt to the software's limitations and realize that the computer does not immediately understand their intentions. The result of the students' experience with GeoGebra could be a realization of the importance of following the software's procedures and limitations. This newfound appreciation for the importance of syntax when using the software can be a constructive opportunity to develop CT.

> "Well, you really had to complete everything step by step because, of course, the computer doesn't immediately know everything you think how it works, so that took some getting used to…"

Gaps in mathematics content knowledge

Students showed a good understanding of how to use computational concepts in GeoGebra, but often mathematics content knowledge gaps hindered effectively applying them in computational problem solving in the developed assignments.

In many cases, students understood computational concepts related to the computational problem solving in GeoGebra to a satisfactory degree but failed to successfully complete the developed assignments due to gaps in mathematics content knowledge. Without a solid foundation in mathematics, students were unable to use computational concepts amd enter the right information into GeoGebra, leading to errors and wrong results. During the interviews, 5 students specifically referred to mathematics content knowledge gaps or need for refreshing their memory as they needed to recall definitions of mathematical terms, characteristics and properties of shapes (e.g., parabolic geometry).

> "At the beginning I had to think for a moment of … okay, what is a parabola again? How do I set it up? And so on … Just a bit of that sort of thing"

> "Yes, setting up tangents and stuff. Of course we knew that but it took some

getting used to again"

"Well... It's been a while so I'm just trying to think of what it was again."

The statements above indicate a need for continued practice or review of mathematical content knowledge in order to maintain students' skills and knowledge before introducing computational concepts and practices.

During the interviews, seven students mentioned they needed their teacher's support to tackle occurring problems, 11 students mentioned that their classmates helped them with assignments, and six students specifically mentioned that they were able to tackle an occurring problem by using the hints of the workbooks.

Students' learning experience

Most students welcomed the idea of using interactive mathematics software like GeoGebra in CT-embedded mathematics lessons. In the interviews of students in the classes that took place physically, the 25 students mentioned positive aspects of the experiences with the lesson series while seven students mentioned negative experiences with the developed learning activity.

Positive attitudes toward working with GeoGebra

GeoGebra is fun

Seven students stated they liked the developed lesson series for different reasons, including the possibility of addressing mathematics problems with digital tools, which makes the activity more fun.

"I actually found it quite fun because normally we do the same things, and now

> we do something different with math... I've never done anything else than just in a book. Because with biology, you have bio practices, of course, but never with mathematics. So it's nice to do something different for a change”

> “But I think it's nice that you don't have a normal lesson with your book and your notebook. That you are doing something different, something new, so to speak. So I like that”

The students’ statements indicate that they have a positive attitude towards using GeoGebra in their math lessons, as they find it to be a fun and engaging change from traditional book-based learning. Moreover, as they have never used anything other than a book to learn mathematics in the past, they highly appreciated the opportunity to use technology to learn and enjoy that they are not doing a traditional lesson with a book and notebook, but instead trying something new and different. The possibility to enable students to take a more active role in learning was also appreciated by the students. Two students stated:

> “I thought it was anyway, sincerely a fun series because it's more fun than just sitting in class and just listening to the teacher and doing assignments. So it was something different and that was kind of fun.”

> “I liked it more than just normal mathematics because it is more problem-solving than just 'here you have the same problem, do it 100 times over... It is more thinking than doing, which I like.”

Structured and clear learning material

16 students connected their positive experiences with being able to successfully solve the assignments and follow the workbook in a clear and structured way. One student

stated:

> “It's going pretty well, I would say. Most of it is finished. It just worked out. Well, with most of them I just read the assignment and I could figure most of it out. And also the GeoGebra worked just fine. It was pretty clear how that worked … And my biggest problem was... Sometimes I messed up a command and I couldn't figure out what went wrong for a while, but I eventually got everything”

Features of GeoGebra

The advantages students saw were related to GeoGebra enabling visualization, evaluation, testing, automating processes, being versatile and useful, and promoting different ways of working and thinking about mathematics problem-solving.

> "Yes, because sort of, you don't have to draw it [the graph] yourself and it's right... It gives the points right away and you can move lines and then it gives the points with it and you can test your solution, that's very convenient"

> "Well, GeoGebra shows it [the graph] when you fill in something and you see what happens. With mathematics, you don't normally see it in front of you and this is very useful"

These statements indicate that students perceived the software as being useful and convenient for learning mathematics and solving mathematics problems, especially those that are difficult to visualize. The software's feature that enables plotting graphs and displaying points quickly and accurately as well as its ability to dynamically adjust lines and test were perceived as major advantages. A student also mentioned that with a tool like GeoGebra you can create general solutions so that if you make a mistake you can revise the assignments without having to start from scratch.

> “The advantage is that when you get a new problem you can easily adjust the lines by just dragging the points around, which is a lot faster than just having to manually adjust everything again. So if you are doing a similar problem multiple times that would be easy. Yeah, just the calculating and you get to see the lines without having to draw it…”

This statement indicates that the student sees the interactivity and ease of use of GeoGebra as a major advantage. The student notes that the ability to quickly adjust lines by simply dragging points is faster and more efficient than manual adjustment. This feature makes it easier to work on similar problems repeatedly, and the student appreciates the ability to calculate and visualize lines without having to draw them by hand. Hence, GeoGebra is perceived to be a time-saving and efficient tool for working on mathematical problems.

Negative attitudes toward working with GeoGebra

Unexpected outcomes, slow loading times, and technical errors

Despite GeoGebra being a reliable and powerful tool, four students experienced technical issues such as software crashing, slow loading times, compatibility issues with their devices and unexpected outcomes that significantly hindered their progress.

> "Sometimes there was just something not coming out or then GeoGebra was loading very long or then you accidentally clicked... Did you kind of have to copy something and then you clicked back, only then you can't continue again in the same formula box, because that's already been used, so then you have to fill it out all over again. So that took a lot of time"

> "When you opened another thing, it didn't keep the other one that you had done before. And it didn't load sometimes. You couldn't click on it because it was maybe too much at once, so that was [annoying]... But the rest was okay"

Such technical issues had a negative impact on the learning experience of students and required valuable time to tackle them. Smooth, user-friendly and stable software environments are essential in engaging students in computational problem solving with mathematical tools and commonly found issues should be addressed by teachers and mentioned in learning material.

Seven students saw disadvantages in using GeoGebra in their lessons because they sometimes do not manage to use the program as intended, they might get stuck in some parts, and they did not know how to proceed using Geogebra. According to them, in contrast to working with pen and paper, technical issues like errors, long loading times in complex assignments, confusing interfaces, and teachers sometimes being unable to help students tackle technical issues that occur when working with GeoGebra.

> "A disadvantage of it: Sometimes it gets stuck when you enter things... With that graph, the computer can't handle it... Especially that tangent, then it gets a little stuck."

> "Well, sometimes it didn't quite do what it was supposed to do. So sometimes it wasn't very easy"

Difficulty with Troubleshooting

In some cases, students had experienced frustration and difficulties when trying to find help with certain problems and there was no one able to support them.

> "As with chapters 5 and 6, you occasionally encounter problems where you need help finding a solution. But you cannot find any help, and that is frustrating because you have the idea of what you do not understand, but no one can help you ... and then it's up to you ... but then it's just a tiny thing, and you are like 'oh, here I made a mistake' ..."

As a consequence, students might feel frustrated and are ultimately forced to rely on themselves to find a solution, even if the solution is fixing a small mistake. This experience highlights the importance of having access to support and resources when encountering problems in the learning process. Three students shared that they had an overall negative experience with the lesson series due to the difficulty of the mathematics content, code errors, and the complexity of the assignments. These challenges have negatively impacted the overall experience and potentially their learning outcomes.

Discussion

Several studies have reported on the benefits of using GeoGebra in mathematics classrooms in terms of developing mathematics content knowledge and mathematical thinking (Arbain & Shukor, 2015). The current study identified advantages and challenges in working on CT-embedded 11th-grade calculus tasks using GeoGebra. The aim of the developed lessons series was on fostering CT skills in calculus lessons but issues like students' mathematics knowledge gaps and inexperience in computing raised challenges in engaging them in computational problem-solving.

Previous studies with GeoGebra focused explicitly on students acquiring mathematics content. Our study considered the CT dimension in the lesson series which was benefited from the same advantages in previous studies with GeoGebra, e.g., visualization, evaluation, testing, automating processes, being versatile and useful, and promoting different ways of working and thinking about mathematics problem-solving. Visualization can be a powerful feature to better comprehend mathematics (Drijvers, 2018) and CT aspects in learning activities. The results of this study showed that interactivity and visualization are particularly useful in understanding and mastering mathematics content knowledge and CT skills through what we call computational experimentation.

As a universally accepted definition for CT does not exist in the academic literature, our design focused on specific aspects of CT that are most frequently considered as its core elements: decomposition, pattern recognition, abstraction, algorithmic thinking/design (Dong et al., 2019; Kynigos & Grizioti, 2018). We presented how these aspects can be addressed in CT-embedded calculus activities with accessible mathematics software that tends to be popular in educational practice, in our case GeoGebra. GeoGebra is widely used by mathematics teachers in the Netherlands and its features provide abundant opportunities to introduce computational problem-solving to secondary students. Even though many mathematics teachers in the Netherlands use GeoGebra in their lessons, our interviews with students, showed that about one-fourth of the students take a passive role in the classroom before participating in the lesson series with the teachers using the software for demonstration purposes only. Our approach puts students in a more active role with mathematics tools and ideas and focuses on promoting CT skills in calculus lessons through computational experimentation.

We collected data for capturing students' CT skills during the developed lesson series from different sources including workbooks, GeoGebra files, and artifact-based interviews with students. The data analysis of different sources provided unique insights from the respective data and methods that allowed us to capture and evaluate the students' learning experiences and the challenges they encountered. The analysis of students' assignments in GeoGebra and the workbooks allowed us to identify misconceptions about mathematics content and also computational concepts and practices e.g. when students were trying to generate graphs under specific conditions – often students used false conditions, macros, and parameters. In addition, during the interviews phase, the students reflected on their learning experiences and problems they encountered during the lesson series.

Our findings suggest that the students participating in the online classes struggled more with the mathematics content than students attending the physical classes. Moreover, the teacher of the online class mentioned that his students were strong in both mathematics and computing, which raises concerns about the didactics approach in the online setting and time distribution for the learning activities. The developed material was designed considering the teacher as a facilitator, not an instructor, but it seems that it would be beneficial to include more scaffolding as well as more time for computational experimentation to get familiar with GeoGebra and basic programming practices.

During artifact-based (Brennan & Resnick, 2012) interviews, we were able to capture students' misconceptions and better understand aspects that they disliked and struggled with. Such insights are not only important for education researchers but also for teachers who strive to equip their students with digital literacies like CT. This calls for adaptations in evaluating the CT learning outcomes of students through simplified pre and post-tests and looking more in-depth at the learning process and CT implementation of students, identifying and addressing the problems that inevitably occur when working with computational tools. Using different methods and data sources enabled us to triangulate our data to better understand how students implemented CT skills. The findings on assessing CT and understanding students' misconceptions and challenges in calculus education reinforce the argument that capturing CT skills requires various assessment methods (Brennan & Resnick, 2012).

Our findings from the workbook and GeoGebra files analysis, as well as the feedback from the teachers of the classes, confirm that most students engaged in computational problem-solving and successfully implemented core CT skills like decomposition, pattern recognition, abstraction, and algorithmic design/thinking to solve the calculus tasks. In contrast with previous work (van Borkulo et al., 2021), we refer to generalization as part of the algorithmic design/thinking for the specific GeoGebra assignments as the students were called to use variables and the sequence structure to translate the calculus problems into GeoGebra code had to use the selection structure to consider the special cases of the equations.

The analysis of the GeoGebra files also revealed that despite the assignments' increasing difficulty, the students made fewer mistakes. Our data analysis also showed that despite students mentioning encountering difficulties in the interviews, they got used to the software and were able to address the calculus problems with more confidence. This indicates that the designed lesson series could potentially contribute to developing pre-university students' CT skills, but more research is needed to effectively evaluate its impact.

The interviews with students indicated that digital tools like GeoGebra are welcome by most students as they add a more fun and independent learning dimension to calculus lessons. The GeoGebra environment enables students to work more efficiently on their assignments under the condition that there is adequate support and that the tasks are structured and clear. Our findings suggest that successfully completing the assignments and overcoming technical difficulties is rewarding for the students. Considering mathematics content knowledge gaps of students and providing adequate resources for tackling technical issues are key to having a positive learning experience.

Conclusion, Limitations, and Future Work

Addressing CT aspects in CT-embedded mathematics education is challenging due to their core similarities with mathematical thinking. In the Netherlands, computing and CT are addressed only in elective computer science courses in secondary education despite being essential for preparing students for employment and digital citizenship. Therefore, introducing CT in mandatory mathematics lessons is beneficial in addressing equity issues and promoting more opportunities for underprivileged and underrepresented students.

Our study showed that powerful tools that are accessible to teachers could be assets in fostering CT skills in pre-university students during mathematics lessons. The students have managed to solve the computational problems in GeoGebra to a satisfactory degree. The didactical approach and providing sufficient time for familiarizing the students with new tools are key to successfully addressing the developed assignments. Sufficient scaffolding during the lessons series is essential, and adjustments for providing enough support for students with no sufficient experience in computing and mathematics content knowledge gaps need to be considered. Similar to traditional computer science learning activities focusing on CT, using a combination of assessment methods for the learning process and outcomes of the CT-embedded learning activities is necessary.

Before discussing the implications of these conclusions for educational practice and research, we should mention the study’s limitations. First, because of the covid pandemic, some students being absent in some lessons, and students not providing consent to analyze their data, our sample is smaller than expected, and the participation of students in the lesson series was not always consistent despite our efforts to provide online alternatives. Moving to online education led to a shorter lesson series and hindered student observation as teachers were overloaded with work and were limited by time constraints. Performing the teaching experiments under more favorable conditions and quantitatively assessing students' learning gains would offer more constructive opportunities to investigate learning gains of students in more depth.

As a second limitation, the study had an exemplary and small-scale character, which implies that its results must be interpreted and extrapolated with caution, especially because the teachers and schools involved in the study volunteered to participate. Therefore, the results may not represent the average high school in the Netherlands or Europe. More teacher professional development opportunities might be needed to equip teachers with the necessary pedagogical content knowledge and familiarize them with computational tools to embed them into their lessons. Cooperation with more experienced internal or external colleagues might be needed too.

The study’s conclusions provide a theoretical basis for the topic of integrating computational and mathematical thinking in calculus lessons. There is evidence that notions of computational problem-solving form a common foundation for simultaneously addressing mathematical and computational thinking goals. Future work could investigate the possible integration of notions from the didactical theories in both computational science and mathematics domains. For example, how do the notions of object formation (reification, encapsulation) commonly used in mathematics education connect to the core CT elements? Even if a start has been made to investigate this interplay, further elaboration in research is needed. Also, more research on the feasibility of the educational approach followed in this study for more scaled-up research designs, including a quantitative measurement of learning gains, would be a valuable contribution to our knowledge in the field.

Adelabu, F. M., Makgato, M., & Ramaligela, M. S. (2019). Enhancing learners' geometric thinking using dynamic geometry computer software. *Journal of Technical Education and Training*, *11*(1), 44–53. https://doi.org/10.30880/jtet.2019.11.01.006

Arbain, N., & Shukor, N. A. (2015). The effects of GeoGebra on students achievement. *Procedia-Social and Behavioral Sciences*, *172*, 208–214. https://doi.org/10.1016/j.sbspro.2015.01.356

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661–670. https://doi.org/10.1016/J.ROBOT.2015.10.008

Brahier, D., Leinwand, S., & Huinker, D. (2014). Principles to Actions: Mathematics Programs as the Core for Student Learning. *The Mathematics Teacher*, *107*(9), 656–658. https://doi.org/10.5951/MATHTEACHER.107.9.0656

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association Meeting, Vancouver, BC, Canada*.

Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015a). Developing mathematical thinking with scratch an experiment with 6th grade students. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9307*, 17–27. https://doi.org/10.1007/978-3-319-24258-3_2/TABLES/2

Carter, N., Bryant-Lukosius, D., Dicenso, A., Blythe, J., & Neville, A. J. (2014). The use of triangulation in qualitative research. *Oncology Nursing Forum*, *41*(5), 545–547. https://doi.org/10.1188/14.ONF.545-547

Chytas, C., Diethelm, I., & Tsilingiris, A. (2018). Learning programming through design: An analysis of parametric design projects in digital fabrication labs and an online makerspace. *IEEE Global Engineering Education Conference, EDUCON*, *2018-April*, 1978–1987. https://doi.org/10.1109/EDUCON.2018.8363478

Dagienė, V., & Sentance, S. (2016). *It's Computational Thinking! Bebras Tasks in the Curriculum*. https://doi.org/10.1007/978-3-319-46747-4_3

Dong, Y., Cateté, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., Andrews, A., & Catetãľ, V. (2019). *PRADA: A Practical Model for Integrating Computational Thinking in K-12 Education*. https://doi.org/10.1145/3287324.3287431

Drijvers, P. (2018). *Empirical Evidence for Benefit? Reviewing Quantitative Research on the Use of Digital Tools in Mathematics Education*. 161–175. https://doi.org/10.1007/978-3-319-76575-4_9

Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. *LNCS*, *4226*, 159–168.

Gravemeijer Koeno, & Cobb Paul. (2006). Design research from a learning design perspective. *Educational Design Research*, 29–63. https://doi.org/10.4324/9780203088364-12

Guzdial, M. (2008). *Education Paving the Way for Computational Thinking*. https://doi.org/10.1145/1378704.1378713

Henderson, P. B., Cortina, T. J., Wing, J. M., & Hazzan, O. (n.d.). *Computational Thinking*.

Jenkins, J. T., Jerkins, J. A., & Stenger, C. L. (2012). A Plan for Immediate Immersion of Computational Thinking into the High School Math Classroom Through a Partnership with the Alabama Math, Science, and Technology Initiative. *Proceedings of the 50th Annual Southeast Regional Conference on - ACM-SE '12*. https://doi.org/10.1145/2184512

Kynigos, C., & Grizioti, M. (2018). Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space. *Informatics in Education*, *17*(2), 321–340. https://doi.org/10.15388/INFEDU.2018.17

Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K-8 curriculum. *ACM Inroads*, *5*(4), 64–71. https://doi.org/10.1145/2684721.2684736

Papert, & Seymour. (1980). *Mindstorms: children, computers, and powerful ideas*.

Perković, L., Settle, A., Hwang, S., & Jones, J. (2010). A framework for computational thinking across the curriculum. *ITiCSE'10 - Proceedings of the 2010 ACM SIGCSE Annual Conference on Innovation and Technology in Computer Science Education*. https://doi.org/10.1145/1822090.1822126

Rich, P. J., Egan, G., & Ellsworth, J. (n.d.). *A Framework for Decomposition in Computational Thinking*. https://doi.org/10.1145/3304221.3319793

Sanford, J. F., & Naidu, J. T. (2016). Computational Thinking Concepts for Grade School. *Contemporary Issues in Education Research (CIER)*, *9*(1), 23–32. https://doi.org/10.19030/CIER.V9I1.9547

Selby, C. C., & Woollard, J. (2010). *Computational Thinking: The Developing Definition*.

van Borkulo, S., Chytas, C., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). *Computational Thinking in the Mathematics Classroom: Fostering Algorithmic Thinking and Generalization Skills Using Dynamic Mathematics Software; Computational Thinking in the Mathematics Classroom: Fostering Algorithmic Thinking and Generalization Skills Using Dynamic Mathematics Software*. https://doi.org/10.1145/3481312.3481319

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/S10956-015-9581-5/FIGURES/2

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Education: Fostering computational literacy in science classrooms: An agent-based approach to integrating computing in secondary-school science courses. *Communications of the ACM*, *57*(8), 24–28. https://doi.org/10.1145/2633031

Wing, J. M. (n.d.). *Research Notebook: Computational Thinking--What and Why? | Carnegie Mellon School of Computer Science*.

Wing, J. M. (2006). Computational thinking. In *Communications of the ACM* (Vol. 49, Issue 3). https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7–14. https://doi.org/10.17471/2499-4324/922

Yazan, B. (2015). The Qualitative Report Three Approaches to Case Study Methods in Education: Yin, Merriam, and Stake Three Approaches to Case Study Methods in Education: Yin, Merriam, and Stake. *Teaching and Learning*, *20*(2), 2–23. http://nsuworks.nova.edu/tqr/vol20/iss2/12