International Journal of Engineering Education Vol. 40, No. 2, pp. 410–422, 2024
Printed in Great Britain

0949-149X/91 $3.00+0.00

Application of Computational Thinking for Teaching and Learning of Undergraduate Engineering Numerical Calculus*

JOSÉ LUIS DÍAZ
Faculty of Education at Universidad a Distancia de Madrid (Udima). Madrid. Spain. E-mail: joseluis.diaz.p@udima.es

YANKO ORDOÑEZ ONTIVEROS
Universidad a Distancia de Madrid, Madrid, Spain. E-mail: yanko.ordonez@udima.es

JULIÁN ROA GONZÁLEZ
Universidad a Distancia de Madrid, Madrid, Spain. E-mail: julian.roa@udima.es

The integration of Computational Thinking (CT) into numerical calculus for engineering may be regarded as a new approach for introducing numerical contents promoting a logical and structured form of reasoning. This study explores the reception of this integration by analyzing the insights derived from semi-structured interviews with students. A total of 65 students participated in the classroom sessions and 24 students participated in the interviews to reflect their feedback. The interviews illuminate both the strengths of the approach, such as the tangible benefits seen in algorithmic thinking, and areas for improvement, including the need for more diverse examples and a balanced focus between CT principles and mathematical techniques. The feedbacks, received from students, reveal the importance of real-world applications and the value of iterative curriculum design. This research provides ideas towards understanding the potential of CT in traditional academic disciplines and emphasizes the significance of students' feedback in refining educational methodologies. The findings are intended to support educators when designing learning experiences in engineering education, more specifically, involving numerical calculus.

Keywords: computational thinking; numerical calculus; engineering education; student feedback

1. Introduction

Over the last years, the educational field has increasingly focused on Computational Thinking (CT) and associated concepts such as coding, programming, and algorithmic thinking. The research [15] provided a semi-systematic review of 55 empirical studies concerning the application of CT in education. The review revealed that a quarter of the studies noted differences in learning processes and outcomes among groups, but few examined how teaching strategies could improve further based on CT [15], and also only minor references are provided to students' feedbacks. The understanding of students' feedback can be regarded as a key aspect when seeking for enhancements in teaching methods, as it ensures that education evolves in a way that is responsive and relevant to the needs of diverse student populations.

For the purpose of compiling students' feedback in a course of numerical calculus with CT, we make use of a qualitative research methodology based on the semi-structured interview process as it provides a blend of structure and flexibility. By examining students' reflections, we aim to understand the strengths, challenges, and potential avenues for refining the proposed pedagogical approach focused on the use of CT. Through this exploration, we hope to shed light on the broader implications of integrating CT methodologies into traditional engineering education and the insights that can be gleaned from student feedback.

The primary general objective of this work is to effectively integrate CT principles into numerical calculus within engineering education. This integration aims to enhance students' problem-solving skills, improve their understanding of complex mathematical concepts, and prepare them for real-world engineering challenges. In addition to this general objective, we state the following specific objectives in particular: Firstly, there's a need to introduce and familiarize students with CT concepts in a manageable and engaging manner, using real-world engineering problems to illustrate these principles clearly. Secondly, diversifying teaching contents is important as the curriculum should explore a broader range of applications in connection with CT methodology. Thirdly, it is relevant to balance the core elements of CT with their specific applications in numerical calculus. This involves creating learning activities that demonstrate both CT principles and their practical application in mathematical contexts.

Based on the outlined objectives, several specific research questions can be formulated to guide the investigation. These questions are aimed at explor-

* Accepted 30 January 2024.

ing the effectiveness, challenges, and impacts of the integration of CT into numerical calculus:

1. How do students perceive the introduction and integration of CT concepts in numerical calculus courses? This question seeks to understand student experiences and perceptions regarding the new methodology introduced.
2. What are the challenges and successes of balancing core CT elements with their specific applications in numerical calculus? This question investigates the practical aspects of curriculum design and teaching strategies, focusing on the integration of CT principles and their application in numerical calculus.
3. How do student perceive that the integration of CT in numerical calculus prepare them for real-world engineering challenges? This question aims to examine the long-term impact of this educational approach on students' readiness for real-world engineering problems.

These research questions are designed to delve into key facets of integrating CT into numerical calculus, helping to evaluate the effectiveness of this approach and identify areas for improvement and further exploration.

2. Theoretical Background

Computational thinking (CT) is an intellectual framework that encompasses a set of problem-solving skills and techniques that software engineers use to write programs and which can be applied to a variety of disciplines. It involves formulating problems in a way that a computer can effectively execute. Wing [1] posited that computational thinking should be a fundamental skill for everyone, not just computer scientists, and should be added to every student's analytical ability just like reading, writing, and arithmetic. This approach to education emphasizes the importance of breaking complex problems down into more manageable parts (decomposition), recognizing patterns (pattern recognition), abstracting problems to define the core issue (abstraction), and setting out step-by-step solutions (algorithmic thinking). Thanks to integrating computational thinking into the curriculum, students are better equipped to tackle complex problems and develop logical reasoning skills that are invaluable in today's digitized world. Grover and Pea [2] emphasized that embedding computational thinking in K-12 education can provide a foundation for students to understand and innovate with the advanced technologies they interact with daily. In the educational landscape, CT integrates well with project-based learning, where students can apply computational strategies in real-world contexts. As the authors in [3] found, even brief interventions in teaching computational thinking can improve problem-solving skills in students. However, for computational thinking to be effectively integrated into education, there is a need for teacher training. As highlighted by [4], educators must be equipped with the right tools and knowledge to impart these skills to students. Moreover, computational thinking is not just about programming; it is about a mindset of problem-solving, as emphasized by [5]. Lastly, the integration of computational thinking in education requires continuous assessment and iteration to ensure that the curriculum remains relevant and effective, as discussed in [6].

According to the literature, there are interesting applications of CT in subjects related to software engineering (see, for example, [16] and [17], as well as the studies cited therein). This is quite natural since it is important to teach programming strategies based on a specific and structured approach that aligns correctly with the reasoning skills required of a future software engineer [18]. The introduction of CT in university mathematics areas is not new. For example, the author in [20] conducts a research with the aim of applying CT ideas to the teaching of calculus. The author in [20] highlights the importance of working on problem-solving strategies and modeling from calculus using structured strategies typical of CT. Alternatively, one of the objectives we set out is to try to expand the scope of CT application to other more specific areas of university mathematics, such as numerical calculus. And we think this is possible as CT offers a structured methodology for problem-solving while promoting algorithmic approaches. Numerical calculus revolves around the comprehension and resolution of mathematical problems using numerical approximations supported by adequate algorithms. The introduction of CT can be useful for educators trying to provide instructions that go beyond the algorithm. Indeed, with CT as the main methodology, a systematic teaching approach can be integrated that aligns with the content to be taught in a logical reasoning process. For this, we can consider the following sequence of concepts based on CT: Firstly, decomposition, a core tenet of CT, encourages students to dissect complex problems into smaller, more manageable components. For instance, in tackling differential equations, students can isolate boundary conditions, initial values, and the governing equation as distinct elements before seeking a solution. Secondly, pattern recognition, another CT principle, can be employed in identifying recurring themes or methods in problems, such as recognizing the application of specific integration techniques in diverse scenarios. This not only speeds up problem-solving but also assists in cementing foundational knowledge. Abstraction, the process of removing unnecessary details and focusing on the essential features, is invaluable in numerical methods where the essence of a problem is often buried under layers of complexity. By teaching students to abstract effectively, educators can ensure that learners focus on core mathematical principles rather than getting lost in intricate calculations [7]. Lastly, a step-by-step problem-solving approach, is central to CT and important in numerical calculus. For example, when teaching iterative methods like the Newton-Raphson method, educators can stress the algorithmic nature of the approach, ensuring that students not only understand the mechanics but also the underlying logic [8]. In summary, the integration of CT into the teaching of numerical calculus in engineering can provide students with a robust framework to tackle complex mathematical challenges, grounding their understanding in logic and structured problem-solving.

2.1 Main Challenges for Educators when Teaching Numerical Calculus

Educators teaching numerical calculus to engineering students face a series of challenges that can impact the instruction and the students' comprehension. One primary challenge is the abstract nature of numerical methods, which can be difficult for students to conceptualize without tangible real-world applications [9]. This abstractness often requires educators to create relatable scenarios or simulations to contextualize concepts, demanding more preparation and creativity. Another significant hurdle is the rapid pace of technological advancements. With the continuous development of computational tools and software, educators must stay updated to ensure that the methods taught are relevant and aligned with industry standards. This can mean constant course revisions and the incorporation of new software or tools into the curriculum. A third challenge is the diverse mathematical backgrounds of engineering students that have different motivations to mathematics since their pre-entrance university level [19] affecting their perceptions to numerical methods as an advanced and complex new scenario. As a consequence, educators often grapple with the dilemma of pacing: moving too fast can lose some students, while moving too slowly can disengage others. Lastly, the integration of theory and practice presents a perennial challenge. While numerical calculus is deeply theoretical, its real value in engineering is its application. Striking a balance between imparting theoretical knowledge and providing practical experiences, such as labs or projects, becomes critical [10].

3. Methodological Aspects when Defining Classroom Sessions Based on CT

When incorporating CT into classroom sessions, several methodological aspects must be considered to ensure effective integration and optimized student learning outcomes. Firstly, it is pivotal to structure lessons around the core tenets of CT: decomposition, pattern recognition, abstraction, and algorithmic thinking [1]. Each session should be designed to progressively introduce these principles, allowing students to internalize the concepts through repeated exposure and application. Secondly, the lesson plans should be oriented towards problem-solving rather than direct content delivery. Encouraging students to actively engage with problems, dissect them, and devise solutions fosters a deeper understanding of the subject matter [11]. This can be facilitated through group activities, hands-on tasks, or real-world scenarios that prompt students to employ CT strategies.

Furthermore, it is essential to employ visual aids, simulations, and interactive tools that can help demystify complex ideas and provide tangible representations [2]. Tools such as Blockly or Scratch can be relevant in this regard, as they offer graphical interfaces that allow students to engage with CT concepts in an intuitive manner. Additionally, continuous assessment and feedback are important. Rather than traditional exams, educators might consider project-based evaluations or portfolios that showcase a student's journey in applying CT over time [3]. This provides a more holistic view of a student's grasp of CT and also encourages reflection and iterative improvement. In conclusion, the methodological approach to integrating CT into classroom sessions necessitates a shift from traditional teaching paradigms. It requires a focus on active learning, hands-on engagement, continuous assessment, and the strategic use of digital tools to bring abstract concepts to life.

4. Proposal of Classroom Sessions based on CT for the Teaching and Learning of Numerical Calculus in Engineering Degrees

In this section, we delve deeper into defining a comprehensive series of activities specifically tailored for students engaged in numerical calculus courses within their engineering curriculum. The primary objective of these activities is to embed CT into the fabric of numerical calculus learning. By doing so, we aim to provide a succinct yet thorough overview of various interactive and engaging sessions. These are designed to impart core numerical calculus concepts, and also to foster a deeper understanding and application of CT principles in solving complex engineering problems. The essence of these activities lies in their adaptability, allowing educators with expertise in numerical calculus to tailor them according to the specific needs and learning pace of their students. Each session is crafted to build upon the previous one, gradually escalating in complexity and encouraging students to apply CT methodologies such as problem decomposition, pattern recognition, abstraction, and algorithmic thinking. These methodologies are important in breaking down complex numerical problems into more manageable parts, recognizing underlying patterns, abstracting essential details, and devising efficient algorithms for problem-solving. To facilitate this, we propose a blend of theoretical instruction and practical application. The theoretical aspect will cover key CT concepts and their relevance in numerical calculus. Simultaneously, practical sessions will involve hands-on activities, problem-solving exercises, and project-based learning where students can apply CT approaches to real-world engineering scenarios. Moreover, special emphasis is placed on collaborative learning. Group activities and discussions will be integral, encouraging students to share insights, challenge each other's thinking, and collectively develop robust solutions to numerical calculus problems. In summary, the aim is to equip future engineers with not just the knowledge of numerical calculus, but also the skill to think computationally. This dual focus is expected to significantly enhance their problem-solving capabilities, preparing them to tackle the complex challenges of the engineering world with a more analytical and systematic approach.

Another relevant aspect is related with the specific contents to be discussed. In this regard, we have selected two core areas which are differential equations on one side and integration on the other side. Indeed, the analyses of differential equations and integration in a numerical calculus course for engineering are strongly supported by literature, emphasizing their fundamental role in engineering problem-solving and analysis (see Chapters 19–21 in [21]). In addition, the work [22] provides foundational insights into the role of integration in engineering, demonstrating its application in computing essential quantities like areas and volumes, which are core in disciplines such as fluid dynamics and material science. In addition, [23] bridges the theoretical and practical aspects of these concepts, demonstrating their applications in solving real-world engineering problems. Additionally, we note that [21–23] constitute basic reference texts for the author of this article in preparing the numerical calculus sessions that have been taught throughout his teaching career.

4.1 Session 1: Introduction to Computational Thinking in Numerical Calculus

Objective: Introduce the concepts of CT and how they align with numerical calculus in engineering.

1. Warm-up Activity (15 minutes): Begin with a brief discussion about daily-life problems that require a step-by-step approach to solve. This will naturally lead into the concept of algorithms.
2. Introduction to CT (20 minutes): Present the four pillars of CT: Decomposition, Pattern Recognition, Abstraction, and Algorithmic Thinking [1] using real-world engineering problems as examples.
3. Numerical Calculus Link (20 minutes): Discuss how numerical calculus often involves iterative methods and algorithms. Introduce a simple numerical problem and solve it using a basic iterative method, emphasizing the CT approach.
4. Hands-on Activity (30 minutes): Divide students into groups and provide each with a different numerical problem. Each group will decompose their problem, recognize the pattern, abstract unnecessary details, and design an algorithmic solution.
5. Reflection and Discussion (15 minutes): Groups share their approach and reflect on how CT aided their problem-solving process.

4.2 Session 2: Decomposition in Differential Equations

Objective: Understand and apply the concept of decomposition to differential equations.

1. Review (10 minutes): Recap the principles of CT and its relevance to numerical calculus.
2. Introduction to Decomposition (20 minutes): Discuss how complex problems can be broken down into smaller, more manageable parts.
3. Differential Equations Activity (40 minutes): Present a complex differential equation. Walk students through breaking it down into initial conditions, boundary conditions, and governing equations.
4. Group Work (30 minutes): Students, in groups, tackle a new set of differential equations, decomposing them and discussing their approach. Take the opportunity to introduce

the numerical calculus.: For example provide the basics of Newton-Raphson methods.
5. Wrap-up (10 minutes): Discuss the importance of decomposition in simplifying and understanding complex equations in engineering.

4.3 Session 3: Pattern Recognition in Integration Techniques

Objective: Identify patterns in various integration techniques using CT.

1. Quick Review (10 minutes): Briefly revisit decomposition and segue into pattern recognition.
2. Introduction to Pattern Recognition (20 minutes): Using engineering examples, explain how recognizing patterns can accelerate problem-solving.
3. Integration Techniques Activity (40 minutes): Present various integration problems, highlighting the patterns in choosing techniques like substitution, integration by parts, or partial fractions.
4. Hands-on Activity (30 minutes): Students attempt multiple integration problems, documenting patterns they recognize in technique selection. Take the opportunity to introduce the numerical calculus.: For example provide the basics of Simpson methods.
5. Discussion (10 minutes): Reflect on how pattern recognition can improve efficiency and accuracy in engineering calculations.

4.4 Session 4: Algorithmic Thinking in Numerical Methods

Objective: Design algorithms for various numerical methods using CT principles.

1. Warm-up (10 minutes): Discuss real-world engineering scenarios where algorithms play a crucial role.
2. Introduction to Algorithmic Thinking (20 minutes): Explore how step-by-step processes are essential in numerical methods.
3. Numerical Methods Activity (40 minutes): Dive deep into a specific numerical method (e.g., Newton-Raphson method). Design an algorithm as a class, emphasizing each step. The specific numerical method should be based in Differential Equations and/or Integration as proposed in the previous sessions.
4. Group Activity (30 minutes): Each group gets a different numerical method. They design an algorithm, then swap with another group to execute it.
5. Conclusion (10 minutes): Reflect on the importance of clear, efficient algorithms in engineering and how CT can aid in their design.

As pointed, the sessions commenced with an introduction to CT's foundational pillars, with emphasis in their applicability to real-world engineering problems. We then explored the nuances of decomposition in differential equations, understanding how to break down intricate problems. The third session revolved around pattern recognition, using integration techniques as a focal point. Finally, the fourth session honed in on the basis of algorithmic thinking, where we dissected various numerical methods. Each session was interactive, comprising group activities, hands-on problem-solving exercises, and reflective discussions. By the culmination of these sessions, we discussed the theoretical underpinnings of CT and applied them in class. Once they were delivered in class, the student provided feedback. This is actually the purpose of the coming sections.

5. Research Framework for Collecting Students' Feedback about the Sessions

We employ a qualitative research methodology founded in the semi-structured interview technique. The semi-structured interview technique is a commonly used method in qualitative research, offering a balance between the flexibility of an open-ended interview and the focus of a structured ethnographic survey. Unlike structured interviews, which follow a rigid format of pre-determined questions, semi-structured interviews are characterized by a set of guiding questions, but allow the interviewer the freedom to diverge and explore topics in more depth based on the participant's responses [12]. This approach ensures that the interviewer can probe deeper into specific areas of interest or clarify ambiguous answers, leading to richer data collection. Advantages of this method include its flexibility, which allows participants to express their views openly and provides researchers with the opportunity to explore unexpected avenues. Additionally, it can yield more detailed and nuanced insights compared to structured interviews, as the respondents are not confined to fixed response options. Furthermore, the rapport established during such interviews can lead to the uncovering of sensitive or in-depth information that might not be revealed in a more structured setting [13]. However, it is essential to acknowledge that while the flexibility of this method is a strength, it can also introduce variability in the data, making analysis more challenging [14].

The semi-structured interview questions for students' were given by:

General Experience:

- How would you describe your overall experience with the sessions integrating CT into numerical calculus?
- What aspects of the sessions did you find most engaging or helpful?

Understanding of Computational Thinking:

- Can you explain in your own words what Computational Thinking is and its relevance to numerical calculus?
- Which of the four pillars of CT (decomposition, pattern recognition, abstraction, algorithmic thinking) resonated the most with you, and why?

Application of CT in Numerical Calculus:

- Can you provide an example from the sessions where you felt the use of CT aided in understanding a numerical calculus concept better?
- Were there moments in the sessions where you felt that CT complicated the learning process rather than simplifying it?

Group Activities & Interaction:

- How did the group activities influence your understanding of CT and numerical calculus?
- Were there any specific group discussions or interactions that stood out to you as particularly insightful or challenging?

Tools & Resources:

- Were there any tools or resources introduced during the sessions that you found especially helpful or challenging?
- How comfortable did you feel using tools like Blockly or Scratch to visualize and apply CT concepts?

Feedback & Improvements:

- What elements of the sessions would you like to see more of in the future?
- Are there areas where you feel the integration of CT and numerical calculus could be improved or adjusted?

Future Implications:

- How do you foresee applying the CT concepts learned in these sessions to other areas of your engineering studies?
- Has this integration of CT into numerical calculus influenced your perspective on problem-solving in engineering?

We recall that the effectiveness of the semi-structured interview technique is that it allows for flexibility. While these questions serve as a guide, the interviewer can delve deeper into any topic based on the student's responses, ensuring a comprehensive understanding of their experiences and insights.

6. Students Feedbacks about the Sessions

We provide a synthesized feedback report for each of the sessions. The sessions were provided in Engineering Degrees involving students from different universities in greater Madrid area in Spain. Specifically, we mention a course of Advanced Mathematics in engineering during the period from 2016 to 2019. This course consisted of three main areas: Ordinary Differential Equations, Numerical Calculation of Differential Equations, and Introduction to Partial Differential Equations. In this case, we were particularly interested in the section dedicated to the Numerical Calculation of Differential Equations, which also included a mention of integral calculus since it was essential for solving first-order differential autonomous systems. Subsequently, another course consisted on Mathematical Foundations Applied to Engineering from 2019 to 2023. This course also contained a section dedicated to the numerical calculation of integrals and the formulation of simple first-order Differential Equations. In both courses a methodology was implemented that focused on problem-solving and its applications in engineering, following the structure of the planned sessions proposed in Section 4.

A total of 65 students participated in the sessions and 24 students participated in the interviews to reflect their feedback. The sampling of 24 students was selected based on a probabilistic approach to express homogeneous collections in their feedbacks. We note that the individual academic performance was not considered as a criterion to be part of the interview as the intention here was to gather insights that reflect the collective experience of the class. This probabilistic method for selecting participants ensured that the feedback was inclusive and representative of the entire student cohort without imposing segregations for sampling purposes.

During the interview process, an interesting theme emerged from the informal discussions. The students revealed that their educational backgrounds in mathematics, prior to their current university courses, were rooted in traditional teaching and learning methodologies. Traditional mathematics education typically follows a structured, often rigid curriculum. It focuses on theoretical aspects of mathematics, formula memorization, and procedural problem-solving. In secondary schools and early university courses, this approach is common, where the emphasis is on acquiring mathematical skills through repetitive practice and a clear-cut, often linear presentation of concepts. This method is less about exploring the practical applications of mathematics and more about ensuring a solid grasp of fundamental mathematical principles and techniques. The transition from this traditional framework to a more contemporary, application-oriented approach experienced in numerical calculus course is significant. This shift likely influences their perceptions and highlights the adaptability and openness of the students to new learning ideas, which is essential in fields that constantly evolve, like engineering and technology.

Preparing a synthesized text from the responses in a semi-structured interview involves a systematic process of data collection, transcription, coding, and thematic analysis. Firstly, we ensured that the interviews were accurately transcribed, capturing not only the verbal responses but also noting emotions that might provide context. Once transcribed, a preliminary read-through helped to familiarize with the data. The next step consisted on coding, where responses were categorized based on recurring patterns, ideas, or themes. Software tools NVivo and Atlas.ti assisted us in this phase. After coding, we engaged in thematic analysis, where the coded data was examined to identify overarching themes or narratives that encapsulated the essence of the responses. These themes then formed the foundation of the synthesized text, providing a coherent and consolidated representation of the interview data. Throughout this process, we continually reflected on our role as researcher and we took into consideration potential biases to ensure the integrity of the synthesis.

Let us present now the feedback provided by the students for each of the sessions.

6.1 Synthesized Students' Feedback for Session 1

As a real-world problem to guide this session, we selected a nonlinear equation coming from stress analysis in structures. The intention was to calculate the points where the structure will experience the highest levels of stress under different loads. The relationship between the applied forces and the resulting stress and strain on a structure is often non-linear. This means that the response of the material (deformation, for example) does not directly correspond to the applied force in a simple, linear way. Materials can behave differently under various conditions and loads, leading to non-linear relationships that need to be quantified with a numerical assessment. To introduce a systematic approach under the scope of CT, we began by tackling decomposition, where students were asked to identify and list all the elements involved in the stress analysis of a structure. This step was important in breaking down the complex problem into smaller, more manageable parts, such as isolating different factors that contribute to the non-linear behavior of materials, including types of forces applied and environmental conditions. Following decomposition, we moved to pattern recognition. The students, having listed the various components, were guided to identify patterns or common factors in different scenarios. This part of the session was particularly insightful as it helped students recognize recurring themes, like the impact of varying load types on material stress or how environmental factors affect material behavior. Next, we focused on abstraction. Students were encouraged to concentrate on the general principles of stress and strain, setting aside specific details like material types or structure sizes initially. This approach helped them to grasp the fundamental principles of how materials react to stress, irrespective of specific conditions. Finally, we applied algorithmic thinking. The students were guided to develop a step-by-step method for analyzing material stress under various conditions. This methodological approach included determining the type of force, analyzing the material properties, and applying an appropriate mathematical model to estimate stress. This last step in the session was significant as it provided students with a logical sequence of steps to solve the problem, helping them develop a consistent approach to evaluate how different factors affect material stress.

The synthesized feedback from Session 1 provided a blend of appreciation and constructive criticism. Students found the use of real-world engineering problems to be a highly effective method for illustrating the main tenets of CT. One student remarked:

> "I never realized how applicable these general ideas of CT were until we started solving the actual engineering problem. It made the abstract ideas much more concrete."

This sentiment was echoed by several others, who appreciated how tangible examples helped demystify complex CT concepts.

However, the session also presented some challenges, particularly regarding the introduction of new CT terminology. A significant number of students felt overwhelmed by the amount of new information, one student commented:

> "It was a lot to take in at once. Maybe easing us into these concepts over a couple of sessions would help."

This feedback pointed towards a need for a more gradual introduction to CT concepts, potentially spreading the material over multiple sessions to enhance comprehension.

Additionally, students expressed a desire for supplementary materials to reinforce their learning. We remark the following feedback from one student:

> "Having some extra resources, maybe online tutorials or reading materials, would help consolidate our understanding outside the classroom."

This feedback indicated a keen interest in extending learning beyond the classroom and a recognition of the need for diverse learning.

In their reflections, students also discussed how the hands-on activity in groups helped in solidifying their understanding.

> "Working in groups with different students was difficult but in a good way because it gave us the opportunity to explain the concepts to each other. For example, a group mate explained something to me that I hadn't understood."

However, the feedback also hinted at the need for more guided support during these activities. Another student, who joined a group of somewhat skeptical students with the new methodology (as they made known to the author during the sessions) commented:

> "Sometimes, we felt a bit lost. Maybe more guidance or examples during the hands-on part would make it easier."

Overall, the feedback from Session 1 highlighted the effectiveness of using practical, real-world problems to teach CT. It also underscored the importance of pacing the introduction of new concepts and providing additional resources for learning. In addition, some students expressed feeling inundated by the new terminology associated with CT. They suggested perhaps breaking the introduction into two sessions, allowing for a more paced introduction.

6.2 Synthesized Students' Feedback for Session 2

The second session, dedicated to understanding and applying the concept of decomposition in CT to differential equations, received generally a positive feedback from the students. The theme had to do with the deformation of a beam made of the material analyzed in Session 1. For this, the so-called "elastic curve differential equation" in beam deformation was considered. As they delved into the differential equation, many students found that breaking the equation down into initial conditions, boundary conditions, and governing equation clarified their understanding of numerical procedures to apply in each case. One student stated:

> "I recognize that when it comes to introducing differential equations, it has been useful to know that solving a problem with initial conditions is not the same as solving one with boundary conditions and that the numerical algorithms are different! I found the boundary problem more difficult to solve because we then had to calculate the constants in an indeterminate problem and the numerical analysis more difficult then."

Another student commented as well:

> "I found that the differential equation is difficult to solve numerically in general for any initial condition and even more so if we add boundary conditions."

However, alongside the appreciation for the decomposition technique, there emerged a desire for more diverse examples. They felt that exploring a wider range of applications would provide a clearer understanding of the numerical technique's versatility and its utility across different mathematical problems. One student commented:

> "The session was really helpful, but I'm curious about how decomposition works in other types of problems we might encounter in engineering."

This sentiment echoed a common theme in the feedback, suggesting an eagerness to broaden their learning horizon.

Moreover, during the group work phase, something curious happened. The students could choose between continuing to work on the "elastic curve equation" in beam deformation or selecting other simpler models related to first-order autonomous systems provided by the author. All the students maintained their choice for the beam deformation equation. Within the group activity, they were presented with the basic resolution algorithm to approximate continuous derivatives by finite difference derivatives. Here they could see how a differential problem turned into an algebraic problem in the form of a system. This transition from continuous to discrete raised many questions among the groups, but it was allowed for students who understood the concept to explain it to those who did not understand it as much. In this way, all the groups reached a common understanding of the situation. Some students indicated that while the group work was enriching, they would have appreciated more guidance or examples from the instructor.

> "Working in groups was great for discussion, but sometimes we weren't sure if we were on the right track. A few more examples to start us off would have been helpful,"

a student suggested.

In the wrap-up discussion, the importance of decomposition in simplifying and understanding complex equations in engineering was reiterated.

Overall, the feedback from Session 2 underscored the importance of using decomposition in understanding mathematical problems. It also highlighted areas for improvement, such as incorporating a broader range of examples and providing more guided support during group activities, to enhance the learning experience.

6.3 Synthesized Students' Feedback for Session 3

The content of this session was centered on proposing various integrals extracted from different fields of engineering and physics, areas with which the author had professional experience in engineering. Integrals from the kinematics of particles in spatial environments were considered, as well as integrals resulting from integrating the Poisson equation in fluid mechanics, and from the field of statistics with the famous Gaussian probability function. The integrals were presented from the beginning with some related to their applications. However, it was noticed that the session revolved around the mathematics of the integrals, alongside their integration techniques, and perhaps less on their specific applications.

This session elicited mixed reactions. Many students found value in the focused approach on integration techniques, stating that it made pattern recognition in mathematical contexts clearer. However, there was also feedback suggesting that the session might have leaned too much into the mathematical domain. A subset of students felt that the essence of CT, especially in terms of pattern recognition, got somewhat overshadowed by the mathematical procedures. But actually, the session was considered very practical despite their feeling of excessive mathematical contents:

> "I realized how identifying patterns in integration could simplify the process. This session has made me more confident in tackling complex integrals."

Another student noted the usefulness of the hands-on activities:

> "The exercises were challenging, but recognizing patterns in choosing the right technique was a game-changer. It was like solving a puzzle."

However, alongside these positive remarks, there was a suggestion for more diversity in the types of problems presented. Students expressed a desire to see pattern recognition applied in a wider range of mathematical contexts, not just limited to integration. One student commented:

> "I enjoyed the session, but I'm curious to see how these pattern recognition strategies apply to other areas of mathematics or engineering."

The introduction of numerical calculus, specifically Simpson's methods, was a highlight for many students, providing them with an insight into how computational approaches can complement traditional analytical methods. Additionally, there was a sense of interest and surprise when it was presented that the numerical method could apply to all the integrals that had appeared during the session (provided certain conditions of regularity were met). A student commented that:

> "The Simpson's method seemed interesting to me because it helps to solve many types of integrals without spending so much time on resolution methods."

Yet, this also led to some students feeling overwhelmed by the sudden introduction of the new Simpson's integration concept. One student mentioned:

> "The numerical calculus part was interesting, but a bit sudden. Maybe a more gradual introduction to these concepts would be helpful."

During the group activity, where students worked on various integration problems and implemented the Simpson's method in a computational routine in Matlab, the importance of collaboration and peer learning was evident. Students found value in discussing and comparing their approaches to problem-solving, leading to a deeper understanding of the subject matter. However, some students felt that additional guidance or structured examples from the instructor would have been beneficial. As one student put it:

> "Group discussions were rich, but at times, we were unsure about our methods. A few more structured examples or checkpoints from the instructor would be great."

Overall, the feedback from Session 3 highlighted the success of integrating pattern recognition with mathematical problem-solving, while also pointing out areas for improvement. Students appreciated the focus on practical applications and hands-on learning but also expressed a need for more diverse problem examples and gradual introduction to complex concepts like numerical calculus.

6.4 Synthesized Students' Feedback for Session 4

For the fourth session, students were required to choose a specific differential equation from those discussed in previous sessions. The ultimate goal was to introduce the Newton-Raphson method for calculating roots in nonlinear functions. To achieve this, it was necessary to convert the differential equation (with nonlinear boundary conditions) into an algebraic, discrete system to which the Newton-Raphson method could then be applied.

Many students expressed appreciation for the session's depth, particularly valuing how it unpacked the step-by-step logic behind numerical algorithms like Newton-Raphson. One student remarked:

> "The process of designing the algorithm for the Newton-Raphson method was interesting. It clarified the method itself and the process behind creating such algorithms."

Another student noted:

> "Working through the algorithm made numerical methods less daunting and more logical."

However, alongside the positive feedback, there were calls for a greater emphasis on real-world applications. Students were intrigued by the mathematical and algorithmic aspects but yearned for more concrete examples demonstrating how these algorithms are applied in engineering scenarios. A student commented:

> "I understand the algorithms better now, but I'm curious about how they're used in real engineering projects. That context would make this learning even more valuable."

The session's hands-on component, where students designed and executed algorithms was particularly engaging. The exercise of swapping algorithms with other groups fostered a dynamic learning environment, encouraging students to adapt and understand diverse problem-solving approaches. Yet, some students felt a need for more guidance during these activities. One student suggested:

> "Designing our own algorithms was challenging in a good way. However, a bit more direction or examples would have helped us feel more confident in our approach."

Other students mentioned that they found it somewhat complicated to understand the code they received from another group; it seemed like there were disconnected parts, but it was easy for them to locate where in the code the Newton-Raphson method was used:

> "The resulting code was somewhat extensive, and in addition, the other group did not carry out a clear separation of the different sub-codes programmed. However, after spending some time reading the code, we were able to find the algorithm that was well implemented."

Before concluding session 4, the author conducted a short presentation about Blockly and Scratch as potential technological tools that could facilitate the application of CT in classrooms. The author commented to the students that Blockly and Scratch are particularly useful in the context of CT because they simplify complex programming concepts into visual, interactive elements that are more accessible to learners. These tools translate the abstract elements of coding into a tangible format, enabling users, especially beginners, to grasp the basics of programming without the added complexity of syntax. Moreover, the author commented to the students that Blockly and Scratch foster an environment where experimentation and exploration are encouraged. Users can easily manipulate blocks or components to see real-time effects, fostering a deeper understanding of cause and effect, logic flow, and algorithmic thinking, which are central to CT. The gamified and colorful interface keeps users motivated and interested, turning learning into a more enjoyable and less intimidating experience. In addition, these tools provide a platform where users can immediately apply CT concepts like problem decomposition, pattern recognition, abstraction, and algorithm design in a hands-on manner. The author had prepared a numerical integration code in these types of visual tools for the students to observe their utility, leaving a more concrete application of these technological tools in the classroom pending for future sessions. Nevertheless, the students were asked for their opinions on these types of tools, even if it was just a first approach to them without having used them yet. Several students expressed enthusiasm about these platforms, particularly appreciating their interactive and visual nature. One student noted:

> "I found Blockly really intuitive. It's a great way to see how our logical steps translate into actual programming actions. It made understanding CT concepts much more hands-on."

Another student shared a similar view regarding Scratch:

> "Using Scratch could be motivating. It helped me visualize the flow of algorithms and understand the sequence of operations in a very clear and graphical way."

However, some students encountered challenges, particularly those less familiar with programming or visual-based learning tools. A student remarked:

> "I struggled initially with Blockly as it was presented by the teacher because it was a new way of thinking about code. It took some time to figure out the block-based approach as opposed to traditional coding".

A few students mentioned that the example provided by the author was very simple and that they may feel limited when trying to apply more complex CT principles. As one student explained:

> "The presentation about Scratch was great for basic algorithms, but it seems strange if we consider the difficult code we programmed involving the differential equation and after the Newton-Raphson method."

Overall, Session 4's focus on algorithmic thinking in numerical methods was well-received, with students appreciating the practical insights and hands-on experience. However, the feedback also highlighted a desire for more real-world context and a gradual introduction to new concepts. This feedback is important for future sessions, indicating a need to balance technical depth with practical applications and structured guidance.

In summation, while students acknowledged the significant strides the sessions made in integrating CT with numerical calculus, they also provided important feedback for enhancement. Their insights emphasized the importance of balance between introducing new concepts and ensuring depth, the need for diverse examples to showcase the breadth of techniques, and the continual interplay between pure CT principles and their mathematical applications. This feedback underscores the iterative nature of curriculum design and highlights areas for refinement to ensure an optimized and holistic learning experience.

7. Discussions

The synthesis of student feedback on the integration of CT into numerical calculus sessions unveils a multifaceted perspective on the educational experience. This feedback is relevant in understanding the dynamic interplay between CT principles and their application in numerical calculus, highlighting both the advantages and challenges of this pedagogical approach. From the outset, students have expressed a deep appreciation for the use of real-world engineering problems in elucidating the core tenets of CT. The inaugural session's focus on tangible examples transformed abstract concepts into digestible, relatable experiences. This practical approach is a significant benefit, as it bridges the gap between theoretical understanding and practical application, a crucial element in engineering education. However, the transition into this new realm of learning was not without its challenges. A considerable number of students felt overwhelmed by the new terminology and the breadth of concepts introduced in the initial session. This feedback points to a fundamental constraint in the integration process – the balance between introducing novel, complex topics and ensuring that students are not inundated with information. The suggestion to break down the introduction of CT into more manageable segments is a testament to the need for pacing in learning new and intricate subjects. The second session built upon this foundation, with students lauding the decomposition approach, especially in the context of dissecting complex differential equations. The ability to decompose a daunting problem into smaller, more manageable parts is a key advantage of CT, and its application in numerical calculus clearly resonated with the students. Nevertheless, the desire for a broader range of examples reflects an underlying constraint in the curriculum design. While focusing on a specific area like differential equations provides depth, it may inadvertently narrow the students' exposure to the diverse applications of decomposition in numerical calculus. The third session's focused on integration techniques and pattern recognition revealed mixed reactions. On the one hand, students appreciated the clear illustration of pattern recognition in mathematical contexts. On the other hand, some felt that the essence of CT was somewhat overshadowed by the mathematical procedures. This feedback underscores a critical challenge in integrating CT into existing curricula – maintaining a balance between the core elements of CT and their application in specific subject areas. The final session's emphasis on algorithmic thinking and its application in numerical methods like Newton-Raphson was highly praised. The practical exercise of designing algorithms demystified complex concepts and provided a clear illustration of the logic underlying these methods. However, the feedback also highlighted a need for more real-world context. Students expressed a keen interest in understanding the practical, real-world applications of these algorithms, emphasizing the need for educational experiences that not only delve into theoretical aspects but also demonstrate their relevance in real engineering scenarios.

The integration of CT into the teaching of numerical calculus in engineering offers a distinct contrast to traditional teaching methods, providing a more dynamic learning experience. Traditional methods often focus on the procedural aspects of numerical calculus, emphasizing formula memorization and repetitive problem-solving techniques. While this approach can be effective for learning specific procedures, it may not fully equip students with the skills needed to tackle complex, real-world problems. In contrast, CT introduces a more nuanced approach to learning numerical calculus. It encourages students to understand the underlying principles and logic of mathematical concepts rather than just focusing on the execution of techniques. This was already contemplated in [18, 20] for other disciplines such as calculus and software engineering, thus providing a line of common findings with what we have encountered in the numerical calculus course. Another aspect to highlight is that while traditional teaching might concentrate on how to perform a calculation, CT encourages students to understand why a particular calculation is necessary and how it fits into the broader context of a problem. In addition, traditional methods often teach a one-size-fits-all approach, providing a set pathway to solve specific types of problems. CT, however, promotes flexibility encouraging students to break down complex problems into smaller, more manageable parts and to recognize patterns. This not only makes challenging concepts more approachable but also enables students to apply these skills to a variety of problems. Moreover, CT fosters a deeper level of engagement with the material. Tools like Blockly and Scratch, for example, transform abstract numerical calculus concepts into tangible, interactive experiences. This hands-on approach is less prevalent in traditional methods, where the focus might be more on paper-and-pencil calculations. CT's interactive nature can lead to a more profound and intuitive understanding of numerical calculus, particularly beneficial for students who might struggle with traditional methods. Collaboration is another area where CT may be relevant. Traditional teaching might involve individual problem-solving and learning, whereas CT often includes collaborative projects and discussions. This not only enhances learning but also mirrors the collaborative nature of real-world engineering, where teamwork and communication are important

8. Conclusions

We provide answers to each of the research questions that were raised in Section 1 Introduction. from a student's perspective. Firstly, we address the question concerning the students perceptions of CT methodology in a numerical calculus course: Students generally perceive the introduction of CT in numerical calculus as a positive and enriching shift from traditional learning methods. They appreciate the focus on understanding underlying principles rather than just memorizing formulas and procedures. The use of CT tools and techniques, like problem decomposition and algorithmic thinking, helps them grasp complex concepts more intuitively. However, some students might feel overwhelmed by the new terminology and the shift from conventional problem-solving methods. They may require a gradual introduction to these concepts, with additional resources like tutorials or supplementary readings to better assimilate the new teaching approach.

In what regards with the second research question connecting with the challenges and successes in balancing CT elements with applications in numerical calculus, we remark that one of the main challenges in integrating CT into numerical calculus lies in ensuring that the core elements of CT are not overshadowed by the mathematical complexity. Instructors need to find a balance between teaching CT principles and applying them to specific numerical calculus problems. A successful integration often involves practical examples where CT concepts are directly applied to numerical calculus problems, helping students see the relevance and applicability of these skills. The successes are most notable when students can independently apply CT approaches to new and complex problems so that they can observe the transferability of CT ideas.

The third research question considered the preparation for real-world engineering challenges through CT integration. Students generally feel that the integration of CT in numerical calculus significantly prepares them for real-world engineering challenges. They learned to approach problems systematically, breaking them down into smaller parts, and applying logical reasoning. The hands-on experience with CT tools and methods, such as creating algorithms for solving real-world problems, boosts their confidence in handling complex engineering tasks. Moreover, the focus on collaborative problem-solving and algorithmic thinking mirrors the collaborative nature of the engineering profession.

In conclusion, our exploration of CT's integration into numerical calculus for engineering underscored the transformative potential of this pedagogical approaches. The semi-structured interview technique, as employed here, provided a window into students' experiences, perceptions, and suggestions. Through this method, students' feedback illuminated the strengths and areas for enhancement in the sessions, revealing a roadmap for future refinements. The feedback emphasized the value of real-world applications, the importance of pacing and diverse examples, and the necessity of maintaining a balance between core CT principles and mathematical procedures. This balance is important to ensure that the curriculum remains responsive to students' needs but also fosters an environment of collaborative growth, where educators and students co-create a dynamic and impactful learning.

Acknowledgements – This article has been financed by the Faculty of Education at Universidad a Distancia de Madrid.

References

1. J. M. Wing, Computational thinking, *Communications of the ACM*, **49**(3), pp. 33–35, 2006.
2. S. Grover and R. Pea, Computational thinking in K–12: A review of the state of the field, *Educational Researcher*, **42**(1), pp. 38–43, 2013.
3. A. Yadav, C. Mayfield, N. Zhou, S. Hambrusch and J. T. Korb, Computational thinking in elementary and secondary teacher education, *ACM Transactions on Computing Education (TOCE)*, **14**(1), pp. 1–16, 2014.
4. M. Berland and U. Wilensky, Comparing virtual and physical robotics environments for supporting complex systems and computational thinking, *Journal of Science Education and Technology*, **24**(5), pp. 628–647, 2015.

5. K. Brennan and M. Resnick, New frameworks for studying and assessing the development of computational thinking, *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada, 2012.
6. K. Lee, A. Sullivan and M. Bers, Collaboration by Design: Using Robotics to Foster Social Interaction in Kindergarten, *Computers in the Schools.* **30**, 2013.
7. B. Martin and A. Mitrovic, Abstraction in problem solving and learning, *International Journal of Engineering Pedagogy*, **6**(2), pp. 65–74, 2016.
8. L. Chen, Algorithmic thinking in numerical analysis: A case study, *European Journal of Engineering Education*, **43**(5), pp. 725–740, 2018.
9. P.W. Thompson, Didactic objects and didactic models in radical constructivism, *Mathematical Thinking and Learning*, **4**(2–3), pp. 145–180, 2002.
10. A. Roberts, Theoretical and practical considerations in teaching applied mathematics to engineers, *European Journal of Engineering Education*, **43**(3), pp. 396–412, 2018.
11. V. Barr and C. Stephenson, Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, **2**(1), pp. 48–54, 2011.
12. B. L. Berg, Qualitative research methods for the social sciences, *Allyn & Bacon*, 2001.
13. B. DiCicco-Bloom and B. F. Crabtree, The qualitative research interview, *Medical Education*, **40**(4), pp. 314–321, 2006.
14. A. Galletta, Mastering the semi-structured interview and beyond: From research design to analysis and publication, *NYU Press*, 2013.
15. C. Wang, J. Shen and J. Chao, Integrating Computational Thinking in STEM Education: A Literature Review, *International Journal of Science and Mathematics Education*, 20. 10.1007/s10763-021-10227-5, 2021.
16. E. Kurilovas and V. Dagiene, Computational Thinking Skills and Adaptation Quality of Virtual Learning Environments for Learning Informatics, *International Journal of Engineering Education*, **32**(4), pp. 1596–1603, 2016.
17. V. Dolgopolovas, T. Jevsikova, V. Dagienė and L. Savulioniene, Exploration of Computational Thinking of Software Engineering Novice Students Based on Solving Computer Science Tasks, *International Journal of Engineering Education*, **32**(3(A)), pp. 1107–1116, 2016
18. R. Burbaite, V. Drasute and V. Stuikys, Integration of computational thinking skills in STEM-driven computer science education, *IEEE Global Engineering Education Conference (EDUCON)*, pp. 1824–1832, 2018.
19. S. Bamforth, C. Robinson, T. Croft and A. Crawford, Retention and progression of engineering students with diverse mathematical backgrounds, *Teaching Mathematics and Its Applications*. **26**, pp. 156–166, 2007.
20. E. Clements, Investigating an Approach to Integrating Computational Thinking into an Undergraduate Calculus Course, *Electronic Thesis and Dissertation Repository*, 7043, 2020.
21. E. Kreyszig, *Advanced Engineering Mathematics, Ed. 10.* John Wiley & Sons, 2020.
22. G. B. Thomas, *Thomas' Calculus, Ed. 15.* Pearson Education. 2022
23. S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers, Ed. 7.* McGraw-Hill Education, 2010.

José Luis Díaz holds a Master of Science in Aerospace Engineering and a PhD in Science and Technology. With over 15 years of experience in the aerospace industry, he has specialized in roles such as aeroreactor engineer and fuel system engineer. For over eight years, José has been sharing his extensive knowledge as a university professor, teaching various mathematical subjects to engineering students. Presently, he serves as a professor of mathematics for Informatics degrees and also holds a position as a professor of Innovation and Research in the Didactics of Mathematics.

Yanko Ordóñez Ontiveros PhD is a professor of innovation and research in education with an emphasis on physical education and the didactics of physical education at the Distance University of Madrid (UDIMA). His background as an engineer and educator gives him an interdisciplinary profile. His research interests are related to educational innovation and, more particularly, how it can be applied to physical education classes.

Julián Roa PhD is the Dean at the Faculty of Education at the Distance University of Madrid (UDIMA). He has been a mathematics teacher at various educational levels, from primary school to university. His areas of interest focus on the didactics of mathematics, innovation, and research in education, with a special emphasis on mathematics.