

RESEARCH ARTICLE

Embedding Computational Thinking in Engineering and Energy Fundamentals: A PBL–Experiential Design Supporting Conceptual Engagement and Skill Development

Nuno Soares Domingues¹  | Maria Cristina Costa² | António Domingos³

¹Instituto Politécnico de Lisboa, UniRE, EDUNOVA.ISPA, Lisbon, Portugal | ²Universidade Politécnica de Tomar, Ci2, EDUNOVA.ISPA, Tomar, Portugal | ³FCT NOVA, EDUNOVA.ISPA, Universidade Nova de Lisboa, Lisbon, Portugal

Correspondence: Nuno Soares Domingues (nndomingues@gmail.com; nuno.domingues@iscl.pt)

Received: 13 September 2025 | **Revised:** 28 July 2026 | **Accepted:** 6 August 2026

Keywords: computational thinking | design-based research | energy education | engineering education | experiential learning | information and communication technologies supported learning | problem-based learning

ABSTRACT

Introductory electrical engineering courses frequently emphasise procedural calculation rather than conceptual reasoning and systems-level problem solving. Although computational thinking (CT) has been recognised as relevant for engineering education, its explicit integration within undergraduate engineering practice remains insufficiently explored. This study investigates how CT can be embedded and analysed within an ICT-supported problem-based learning (PBL) environment aligned with Kolb's experiential learning cycle in undergraduate electrical engineering education. A design-based research approach was implemented during a 14-week intervention in electrotechnics and electrical machines courses. Custom-developed simulation tools were integrated into lecture and laboratory activities to support collaborative problem solving, parameter exploration and student-generated engineering problems. Data sources included student-generated artefacts, reflection reports, simulation-based activities and structured classroom observations. Analysis employed predefined coding rubrics aligned with five CT dimensions: abstraction, decomposition, algorithmic thinking, automation and generalisation. Inter-rater agreement exceeded Cohen's $\kappa > 0.80$. Student-generated artefacts increasingly demonstrated explicit circuit decomposition, structured analytical procedures and transfer of reasoning strategies across DC and AC contexts. Simulation tools evolved from being used primarily for answer verification to supporting exploratory modelling and comparative analysis of alternative circuit conditions. Classroom observations indicated increased collaborative reasoning, engagement and autonomous problem formulation during later intervention phases. Descriptive coding frequencies across artefacts and observations provided convergent evidence of increased enactment of CT-related behaviours over time. The study demonstrates how CT can be operationalised within foundational electrical engineering education through the integration of ICT-supported simulation environments, PBL activities and experiential learning structures. The work contributes a replicable instructional and analytical framework linking CT dimensions with observable engineering problem-solving practices and technology-supported learning activities.

Abbreviations: AC, alternating current; CT, computational thinking; DBR, design-based research; DC, direct current; ICT, information and communication technologies; PBL, problem-based learning; RQ, research question; SDGs, sustainable development goals.

All rights reserved, including rights for text and data mining and training of artificial intelligence technologies or similar technologies.

© 2026 Wiley Periodicals LLC.

1 | Introduction

Computer-based approaches have a long history in engineering education and engineering practice, including early efforts to facilitate computer programme exchange and broader visions of computer-enabled engineering transformation [1, 2]. Engineering education increasingly requires students to address complex and ill-structured problems that extend beyond procedural calculation and routine formula application. Contemporary engineering practice demands not only technical knowledge, but also the capacity to structure problems, analyse systems, transfer knowledge across contexts and engage in iterative reasoning processes. However, introductory engineering courses frequently continue to prioritise procedural problem solving, often limiting opportunities for students to develop deeper conceptual understanding and transferable analytical practices. This challenge is particularly evident in foundational electrical engineering education, where students commonly experience difficulty relating mathematical procedures to circuit behaviour and system-level reasoning.

Within this context, CT has emerged as a potentially valuable framework for supporting analytical reasoning and structured problem solving in STEM education and has increasingly been conceptualised as a transferable approach to problem formulation, abstraction, decomposition and systematic solution development rather than as a skill restricted to programming [3–6]. Originally conceptualised by Jeannette Wing as a general problem-solving approach extending beyond computer science, CT encompasses processes such as abstraction, decomposition, algorithmic reasoning, automation and generalisation. In engineering contexts, these dimensions align closely with disciplinary practices associated with modelling, system analysis, iterative testing and solution development. Despite growing recognition of computational thinking as a transferable competency, its systematic integration into higher education remains less developed than its adoption in school-level and introductory computing education [7, 8]. Educational frameworks have progressively translated computational thinking into identifiable practices and learning activities, supporting its integration beyond computer science and into broader educational contexts [9–11].

Existing work in engineering education frequently treats CT as an implicit outcome of programming or simulation activities rather than as an explicitly designed instructional and analytical construct. As a result, limited evidence is available regarding how CT-related practices emerge during engineering problem solving, how they can be supported through instructional design, and how such processes can be analysed systematically within discipline-specific contexts. This limitation is especially relevant in introductory electrical engineering courses, where simulation tools are widely used but are often disconnected from theoretically grounded pedagogical frameworks. Recent research has broadened the scope of computational thinking towards STEM education, AI-supported learning, programming-based problem solving and technology-mediated pedagogies while continuing to emphasise the need for clearer disciplinary integration [12–15].

Research in experiential and PBL suggests that these approaches may provide productive conditions for supporting CT-

related practices in engineering education. Kolb's experiential learning theory conceptualises learning as a cyclical process involving concrete experience, reflective observation, abstract conceptualisation and active experimentation. Similarly, PBL emphasises authentic and collaborative problem-solving activities that encourage knowledge construction, enquiry and analytical reasoning. These perspectives share important conceptual affinities with CT because they foreground iterative exploration, structured reasoning and contextualised problem solving. Nevertheless, the integration of CT, experiential learning and ICT-supported engineering education remains insufficiently explored in empirical research.

The present study addresses this gap by examining how CT can be explicitly embedded within an ICT-supported PBL environment structured according to Kolb's experiential learning cycle in introductory electrotechnics and electrical machines courses. Using a design-based research (DBR) approach, the study investigates observable patterns associated with CT processes during simulation-based engineering problem-solving activities involving DC and AC circuit analysis.

The intervention incorporated custom-developed educational simulation tools integrated into regular lecture and laboratory activities over a 14-week period. Data sources included student-generated engineering problems, reflective artefacts, classroom observations and simulation-based tasks. Analysis focused on identifying coding patterns associated with abstraction, decomposition, algorithmic thinking, automation and generalisation across the intervention phases.

Accordingly, the study addresses the following research questions:

RQ1. *How does the integration of ICT-based problem formulation and simulation tools support the enactment of CT processes in electrotechnics and electrical machines courses?*

RQ2. *What observable patterns emerge in students' conceptual engagement with DC and AC circuit analysis throughout the intervention?*

RQ3. *How do students engage with and appropriate the simulation tools during collaborative engineering problem-solving activities?*

The study adopts a qualitative DBR orientation centred on analytical traceability, coding transparency and descriptive evidence rather than inferential statistical generalisation. Frequencies represent coded occurrences rather than participant counts. Multiple coded occurrences could be identified within a single artefact when distinct CT-related behaviours were observed. Frequencies are therefore presented descriptively to support analytical transparency rather than inferential interpretation.

This study contributes to engineering education research in three ways. First, it extends CT research into undergraduate electrical engineering contexts, where empirical evidence remains limited. Second, it proposes an integrated instructional framework linking CT, experiential learning and PBL within authentic engineering tasks. Third, it provides an analytically

traceable qualitative account of how CT-related practices can be identified and interpreted through student artefacts, observational records and simulation-supported learning activities in engineering education environments.

2 | Theoretical and Conceptual Framework

This study is grounded in an integrated framework that combines CT as the cognitive skill set, Kolb's [16] experiential learning cycle as the process model, and PBL as the instructional strategy. Together, these perspectives provide the theoretical rationale for the design, the expected mechanisms of learning and the analytic lens through which findings are interpreted.

In this study, CT was not treated as a generic or implicit outcome but was explicitly operationalised through a set of observable cognitive processes embedded within learning tasks, digital tools and assessment instruments. Each CT dimension was deliberately mapped to specific technical features of the developed tools, to structured problem-based activities and to corresponding assessment indicators.

The operationalisation followed three principles: (i) explicit elicitation of CT processes through task design, (ii) technical support of those processes via interactive simulation and parameter manipulation and (iii) systematic assessment using scenario-based instruments aligned with each CT dimension.

2.1 | Computational Thinking as a Core Competency

CT refers to the ability to formulate problems and express their solutions in ways that can be effectively carried out by a computational agent [17]. Beyond its origins in computer science, CT has been increasingly recognised as a cross-cutting cognitive competency essential for STEM and engineering contexts [18, 19]. CT encompasses multiple interrelated dimensions, including abstraction, decomposition, algorithmic thinking, automation and generalisation [19]. Algorithmic thinking involves the formulation of systematic and ordered procedures for solving problems, providing an explicit bridge between conceptual reasoning and computational implementation [20]. In engineering education, these skills align closely with the problem-solving processes required to navigate ill-structured problems and to transfer solutions across domains [21]. Computational approaches have progressively become embedded in engineering practice, supporting modelling, analysis, simulation and problem solving, thereby strengthening the relevance of computational thinking within engineering education [1, 3]. Recent work has also emphasised that computational reasoning should be understood not merely as algorithm execution but as the explicit representation of reasoning processes that remain transparent, interpretable and traceable throughout decision-making. Although developed in the context of clinical decision support, the hybrid reasoning framework proposed by Domingues [22] demonstrates how structured reasoning architectures can enhance analytical transparency by making intermediate reasoning steps explicit rather than treating

computational outputs as opaque results. This perspective aligns closely with the educational objectives of computational thinking, where students are expected to externalise and justify their reasoning processes during engineering problem solving.

Despite the broad advocacy for CT, empirical research in higher education has been limited, with most studies focusing on introductory programming or K–12 education [23, 24]. Few studies have investigated how CT can be systematically embedded in discipline-specific undergraduate engineering contexts, where conceptual challenges—such as persistent misconceptions in circuit analysis—remain prevalent [25]. This study addresses this gap by positioning CT as an explicit learning outcome within a core engineering curriculum.

Recent studies further indicate that computational thinking can be developed through different pedagogical modalities, including unplugged activities, block-based programming and robotics-based learning, reinforcing the importance of designing learning experiences around the underlying cognitive processes rather than around a single technological medium [26, 27].

2.2 | Kolb's Experiential Learning Cycle

The emphasis on active engagement, progressive learning and the connection between learning and experience has important historical antecedents in educational thought [28, 29]. Kolb's experiential perspective has also been operationalised through instruments and learning-style dimensions intended to characterise how learners engage with experiential learning processes [30]. The cycle comprises four stages: concrete experience, reflective observation, abstract conceptualisation and active experimentation. Engineering learning environments are well suited to this model, as they require students to iteratively test and refine theoretical knowledge through hands-on application [31]. Prior research suggests that situating learning in such cycles can strengthen students' ability to transfer knowledge, integrate theory with practice and develop adaptive expertise [32].

In this study, Kolb's cycle guided the sequencing of learning activities: ICT-based simulation tasks provided concrete experience, collaborative discussions fostered reflective observation, instructor-guided explanations facilitated abstract conceptualisation and student-generated problem design supported active experimentation.

2.3 | PBL as Instructional Context

PBL positions students as active agents who construct knowledge through engagement with authentic, complex and open-ended problems rather than passive reception of information [33]. Within engineering education, PBL has been shown to enhance motivation, collaborative problem solving and self-directed learning compared with traditional instruction [32, 34]. By progressively shifting responsibility from instructor-provided to student-generated problems, PBL supports higher-order skills in problem formulation, design and decision-making, areas in which engineering students frequently exhibit difficulty [35].

Recent research further confirms the continuing relevance of PBL for current cohorts of higher-education students, whose learning processes are shaped by digital tools, expectations of authenticity and the need for future-ready competencies. A 2024 meta-analysis examining student-centred, problem- and project-based approaches found that PBL is associated with positive effects on students' motivation and affective beliefs, which are critical for engagement and persistence in complex tasks [36]. This aligns with evidence that PBL environments can promote critical thinking and problem-solving skills across domains, although outcomes may vary depending on student characteristics and the quality of facilitation (e.g. attendance and academic preparedness) [37].

In the engineering context, systematic reviews indicate that PBL contributes to the development of discipline-relevant competencies, including teamwork, design thinking and professional reasoning, when instructional models are deliberately aligned with real-world practices and supported by appropriate scaffolding [38]. Furthermore, recent work emphasises the role of technology-mediated and guided PBL, wherein digital simulations, collaborative platforms and iterative feedback mechanisms strengthen student engagement and support conceptual transfer. Technology-supported learning environments can extend conventional classroom interaction by providing additional opportunities for communication, exploration and learner activity beyond traditional instructional settings [39, 40]. Collectively, this evidence suggests that contemporary PBL implementations must extend beyond problem exposure to include structured reflection, digital interaction and formative guidance if they are to meaningfully support deep learning and higher-order competencies.

In the present study, PBL provided the instructional framework within which CT and experiential learning were integrated. Authentic, ill-structured problems were presented through ICT-based tools that mirrored the complexity of engineering practice and required students to exercise CT dimensions within the structure of Kolb's experiential learning cycle. This implementation thereby served not merely as a pedagogical strategy, but as a coherent context for developing and accessing discipline-specific cognitive competencies aligned with both professional practice and contemporary student expectations.

Integrated STEM activities can provide a context for connecting disciplinary knowledge with authentic problem solving and professional practice, including within teacher-development initiatives in Portugal [41].

2.4 | Integrated Framework

The integration of CT, experiential learning and PBL offers a three-layered conceptual model:

- CT defines the competencies to be developed.
- Kolb's cycle describes the process through which those competencies are acquired.
- PBL situates learning in an authentic, collaborative context.

This framework predicts that embedding CT within a PBL-experiential design may contribute to improvements in both

cognitive performance (CT skills and conceptual understanding) and affective engagement (motivation, collaboration and transferability). By empirically testing this integrated design, the study contributes to ongoing debates in engineering education on how to meaningfully embed '21st-century skills' into disciplinary learning environments [31, 42].

3 | Materials and Methods

3.1 | Research Design

This study adopted a DBR approach (Reeves, 2006) to investigate how CT can be embedded and enacted within an ICT-supported PBL environment in undergraduate engineering education. DBR was selected because it enables the systematic study of educational innovations as they are designed, implemented and refined in authentic classroom contexts, while generating practice-oriented and theoretically informed insights relevant to engineering education.

The research involved the iterative development and classroom integration of ICT-based learning tools aligned with Kolb's experiential learning cycle and PBL principles. Rather than evaluating a fixed intervention under controlled conditions, the DBR framework supported continuous alignment between instructional design decisions, observed student activity and analytical refinement across successive implementation phases. This approach is particularly appropriate for examining complex pedagogical constructs such as CT, which must be interpreted in relation to task design, tool affordances and learner interaction.

Data collection followed a qualitative-dominant, multi-source analytical design consistent with DBR methodology. Primary data sources included student-generated problems and solutions, structured classroom observations and reflective artefacts produced during the intervention. These data were complemented by structured assessment instruments designed to elicit specific CT dimensions and to document patterns of conceptual engagement over time. Any numerical indicators derived from these instruments were used descriptively to support interpretation and analytical triangulation, rather than to support causal inference or statistically generalisable claims.

Together, this research design enabled the examination of how CT processes emerged and evolved within authentic engineering learning activities, while maintaining methodological rigour through explicit operationalisation of constructs, predefined analytical protocols and systematic documentation of the intervention.

The investigation was guided by the research questions RQ1 to RQ3 presented in the introduction chapter.

3.2 | Computational Thinking Model

The CT model adopted in this study was adapted from the multidimensional framework proposed by Shute, Sun and Asbell-Clarke [19]. That framework conceptualises CT as a set of interrelated cognitive processes that support problem

formulation, solution development and transfer across contexts. Given the disciplinary focus of this study, the model was selectively adapted to ensure construct relevance, observability and alignment with electrical engineering problem-solving practices.

Five CT dimensions were retained: abstraction, decomposition, algorithmic thinking, automation and generalisation. These dimensions were selected because they directly correspond to the cognitive processes required in circuit analysis and systems reasoning. Abstraction enables students to simplify complex electrical systems by identifying relevant variables and relationships. Decomposition supports the break down of circuits into sub-circuits or functional blocks for analysis. Algorithmic thinking underpins the development of structured, stepwise solution procedures, such as the systematic application of Ohm's law and Kirchhoff's laws. Automation reflects the ability to formalise and repeatedly apply solution logic using computational or simulation tools, without replacing conceptual reasoning. Generalisation supports the transfer of solution strategies across different circuit topologies and operating conditions.

Other CT-related elements discussed in the broader literature, such as debugging or data practices, were not included. These elements are either implicitly embedded within the selected dimensions (e.g., error checking within algorithmic thinking) or are less directly observable in non-programming, circuit-based learning tasks. The adaptation therefore followed the principle of construct alignment, retaining only those dimensions that could be meaningfully elicited and assessed within the instructional context.

Each CT dimension was operationalised through scenario-based assessment items and embedded learning tasks designed to require explicit demonstration of the target cognitive process rather than procedural recall. Parallel pre- and post-intervention instruments were used to minimise testing effects. A detailed description of the final CT model, including dimension definitions, indicators, sample tasks and scoring criteria, is provided in Appendix 12 to ensure transparency and construct validity.

3.3 | Context and Participants

The study was conducted at the Instituto Superior de Engenharia de Lisboa within the Mechanical Engineering programme. Participants were second-year undergraduate students enrolled in the Electrotechnics ($n = 48$) and Electrical Machines ($n = 42$) courses during the 2024–2025 academic year, in both semesters. All students had completed introductory programming, physics and mathematics lectures but had no prior formal training in CT as an explicit learning objective.

3.4 | Intervention Phases and Replication Protocol

To ensure transparency and replicability, the intervention was structured into four clearly defined phases, aligned with Kolb's

experiential learning cycle and implemented consistently across both courses. Each phase had explicit objectives, activities, artefacts and assessment procedures, as summarised below.

3.4.1 | Phase 1—Guided Problem Engagement (Weeks 1–3: Concrete Experience)

Students engaged with instructor-provided circuit problems using the ICT-based simulators during lecture and laboratory sessions. At this stage, problems were fully specified, and students were required to manipulate parameters, observe system behaviour and compare outcomes with analytical solutions derived manually. The instructional focus was on familiarisation with the tools and on eliciting initial CT processes, particularly abstraction and decomposition.

3.4.2 | Phase 2—Structured Exploration and Reflection (Weeks 4–6: Reflective Observation)

Students worked in small groups (3–4 members) to analyse variations of the initial problems. Groups were required to document assumptions, identify relevant variables and justify solution strategies. Structured reflection prompts were provided at the end of each session, requiring students to explain observed discrepancies between simulated and analytical results. These reflections constituted assessable artefacts for qualitative analysis.

3.4.3 | Phase 3—Student-Generated Problem Design (Weeks 7–10: Abstract Conceptualisation)

Responsibility progressively shifted to students, who were tasked with designing original circuit problems using the simulator environment. Allowing students to formulate and modify their own problems shifts part of the learning process from solving predefined exercises towards constructing and testing personally meaningful representations, consistent with constructionist principles [43]. Problems had to include at least one parameter variation and one 'what-if' scenario (e.g., tolerance changes, component faults). This phase explicitly targeted algorithmic thinking and generalisation, as students were required to articulate stepwise solution procedures and explain how their strategies could transfer to structurally similar circuits.

3.4.4 | Phase 4—Experimental Extension and Transfer (Weeks 11–14: Active Experimentation)

In the final phase, students extended their designs to more complex or unfamiliar configurations (e.g., transition from DC to AC circuits). Groups tested hypotheses by modifying circuit topology or operating conditions and evaluated the robustness of their solution strategies. Outcomes were presented and discussed in plenary sessions, allowing peer critique and cross-group comparison.

Across all phases, instructional time allocation, task structure and assessment criteria were standardised. All activities were integrated into normal course delivery to avoid artificial experimental conditions.

3.5 | Intervention

The intervention consisted of the development and integration of self-developed ICT tools (including simulators, collaborative challenges and interactive serious games) into lecture and lab sessions. The tools supported:

- DC and AC circuit simulation with adjustable parameters.
- Fault analysis and tolerance testing.
- Kirchhoff's current and voltage law applications.
- Energy analysis in capacitors and inductors.
- Passive filter design and oscilloscope simulation.

The tools were embedded into a PBL framework, where students progressed from instructor-provided exercises to self-designed problems. Tool use was sequenced to follow Kolb's learning cycle: concrete experience (hands-on simulation), reflective observation (analysis of results), abstract conceptualisation (theoretical linkage) and active experimentation (designing new scenarios).

3.6 | Data Collection

Data were collected over a 14-week semester using the following instruments:

- Student-generated problems and solutions: Collected from project submissions; analysed for complexity, originality and correctness using a coding scheme developed iteratively by two independent raters.
- Classroom observations: Conducted by the instructor and a research assistant using a structured observation protocol to capture collaboration patterns, participation and on-task behaviour.

All instruments, observation protocols, reflection prompts and scoring rubrics were finalised prior to Phase 3 of the intervention and applied uniformly across cohorts. Parallel pre- and post-tests were matched at the item level to ensure construct equivalence while minimising recall effects. Sample items, coding schemes and scoring criteria are provided in the appendices to support replication.

3.7 | Data Analysis

Data analysis followed a qualitative-dominant DBR approach, aligned with the nature of the collected data and the exploratory aims of the study. Analysis focused on identifying patterns in students' problem formulation, use of CT processes and engagement during problem-based activities.

Student-generated problems and corresponding solutions were analysed using a structured rubric focusing on three predefined dimensions: (i) complexity of the problem structure, (ii) originality of problem formulation and (iii) correctness and coherence of solution strategies. The coding scheme was developed iteratively during pilot analysis and finalised prior to Phase 3 of the intervention. All artefacts were coded independently by two

researchers. Inter-rater reliability was established using Cohen's kappa, with values meeting or exceeding the accepted threshold of 0.80. Disagreements were resolved through discussion until consensus was reached.

Classroom observations were analysed using a structured observation protocol to identify recurring patterns of collaboration, participation and on-task behaviour across intervention phases. Observation records were segmented by session and phase to enable systematic comparison over time. Rather than quantifying behaviour frequency for statistical inference, observation data were used to characterise shifts in engagement and interaction patterns as the intervention progressed.

Analytical triangulation was employed across student-generated artefacts and classroom observations to strengthen interpretive validity. Findings were integrated at the level of each research question, allowing convergence between observed learning behaviours and characteristics of student-produced work. To reduce researcher bias, qualitative coding was conducted independently by two researchers, and analytic decisions were documented throughout the process to ensure transparency and replicability.

Consistent with the DBR paradigm, the purpose of the analysis was not to establish causal effects or statistically generalisable outcomes, but to generate empirically grounded insights into how CT is enacted within an ICT-supported PBL environment.

3.8 | Analytical Traceability and Coding Procedure

The analytical process followed a structured multi-stage procedure designed to ensure methodological traceability between raw data, coding decisions and interpretation of findings. This emphasis on explicit analytical traceability is consistent with recent work on transparent reasoning frameworks, where intermediate reasoning stages are made observable to improve interpretability and reproducibility of decision processes [22]. Although proposed for clinical decision support, the underlying principle of exposing reasoning pathways rather than relying solely on final outputs is equally applicable to educational research based on qualitative coding and computational thinking.

In the first stage, all collected artefacts and observation records were anonymised and organised chronologically according to intervention phase. Data sources included student-generated engineering problems, solution reports, reflection artefacts, classroom observation records and simulation-based activities.

In the second stage, the research team applied a predefined coding framework aligned with the five CT dimensions adopted in this study: abstraction, decomposition, algorithmic thinking, automation and generalisation. Each CT dimension was associated with explicit operational indicators defined prior to the main analysis phase.

Coding was conducted at the level of observable behavioural or analytical evidence. A coding occurrence was registered only

when a student artefact, reflection, discussion episode, or simulation interaction explicitly demonstrated the target CT-related behaviour according to the predefined rubric.

For example:

- abstraction was coded when students identified relevant electrical variables while suppressing non-essential circuit information;
- decomposition was coded when circuits were explicitly separated into functional analytical units;
- algorithmic thinking was coded when students articulated coherent sequential solution procedures;
- automation was coded when simulation tools were used iteratively to test parameter variations or alternative conditions;
- generalisation was coded when prior analytical strategies were transferred to novel circuit configurations.

In the third stage, coded occurrences were aggregated by intervention phase to identify longitudinal patterns in CT enactment throughout the semester. Frequencies were used descriptively to support analytical transparency and triangulation across data sources rather than inferential statistical analysis.

To strengthen reliability and reduce interpretive bias, all coding was performed independently by two researchers. Inter-rater agreement exceeded Cohen's $\kappa = 0.80$ across coding categories. Discrepancies were resolved through iterative discussion and consensus review.

Finally, findings from coded artefacts and classroom observations were triangulated and interpreted in direct relation to the research questions, thereby establishing explicit analytical traceability between empirical evidence and reported conclusions.

3.9 | Application

The digital tool employed in this study was designed and developed by the author specifically for the purposes of this research and for use within the associated curricular unit. The tool was implemented using *Visual Studio* and tailored to support lecture delivery and PBL activities in electrical engineering.

The tools developed were conceived as pedagogical support instruments rather than substitutes for fundamental analytical processes. In line with established educational principles, students were not encouraged to rely on automated outputs before developing proficiency in manual calculation and conceptual reasoning. Analogous to the pedagogical role of calculators in engineering education, the digital tools were intended to augment conceptual understanding and facilitate deeper engagement, rather than replace analytical thinking.

The design of the tool was guided by three principal criteria: pedagogical alignment, usability and collaborative support. Pedagogical alignment was ensured by embedding interactive problem scenarios that directly supported the learning

outcomes of the subject and facilitated the operationalisation of CT dimensions examined in this study. Usability considerations prioritised interface clarity, minimal cognitive load and ease of interaction, enabling students to engage with the tool without requiring prior technical training. Collaborative support was incorporated to encourage shared exploration, discussion and cooperative problem solving during classroom activities.

A key advantage of the tool lies in its capacity to rapidly generate circuit data under varying conditions, including parameter variation, component faults and operational anomalies. This functionality enabled students to explore 'what-if' scenarios efficiently, thereby supporting experimentation, reflection and generalisation within the PBL framework.

The development process drew on the author's professional expertise in software development and educational tool design. Iterative refinement was undertaken through classroom use and informal student feedback to ensure functional reliability, instructional suitability and alignment with course objectives. Although the tool was not developed as a commercial product, its design followed established principles of educational software development, with a focus on enhancing student motivation, engagement and effective knowledge transmission in an engineering education context.

Figures 1–3 illustrate selected examples of the developed tools. These applications are capable of producing outputs that not only visualise circuit behaviour but also encourage critical analysis, enabling students to better understand the underlying physical and electrical phenomena.

Some examples for exercises in Figure 2, circuits are:

- With all switches S1, S2 and S3 on and given $R3 = 100$.
- What should the value of the remaining resistances be to obtain the same current in R3, R7 and R8?
- What should the value of the remaining resistances be to obtain the same voltage drop in R3, R4, R5, R6, R7 and R8?
- If the value of R3 decreases by 10%, what happens to the current source? Explain.
- If the value of R3 decreases by 10%, what happens to the voltage drop in R4? Explain.
- If the value of R3 decreases by 10%, what happens to the voltage drop in R7? Explain.
- If the value of R7 decreases by 10%, what happens to the current source? Explain.
- If the value of R7 decreases by 10%, what happens to the voltage drop in R4? Explain.
- If the value of R7 decreases by 10%, what happens to the voltage drop in R7? Explain.
- If the value of R7 decreases by 10%, what happens to the voltage drop in R8? Explain.

Starting from them, students suggest and develop their own exercises. Figure 1 uses Ohm's Law, the power definition and the principles of current division and voltage division. A natural

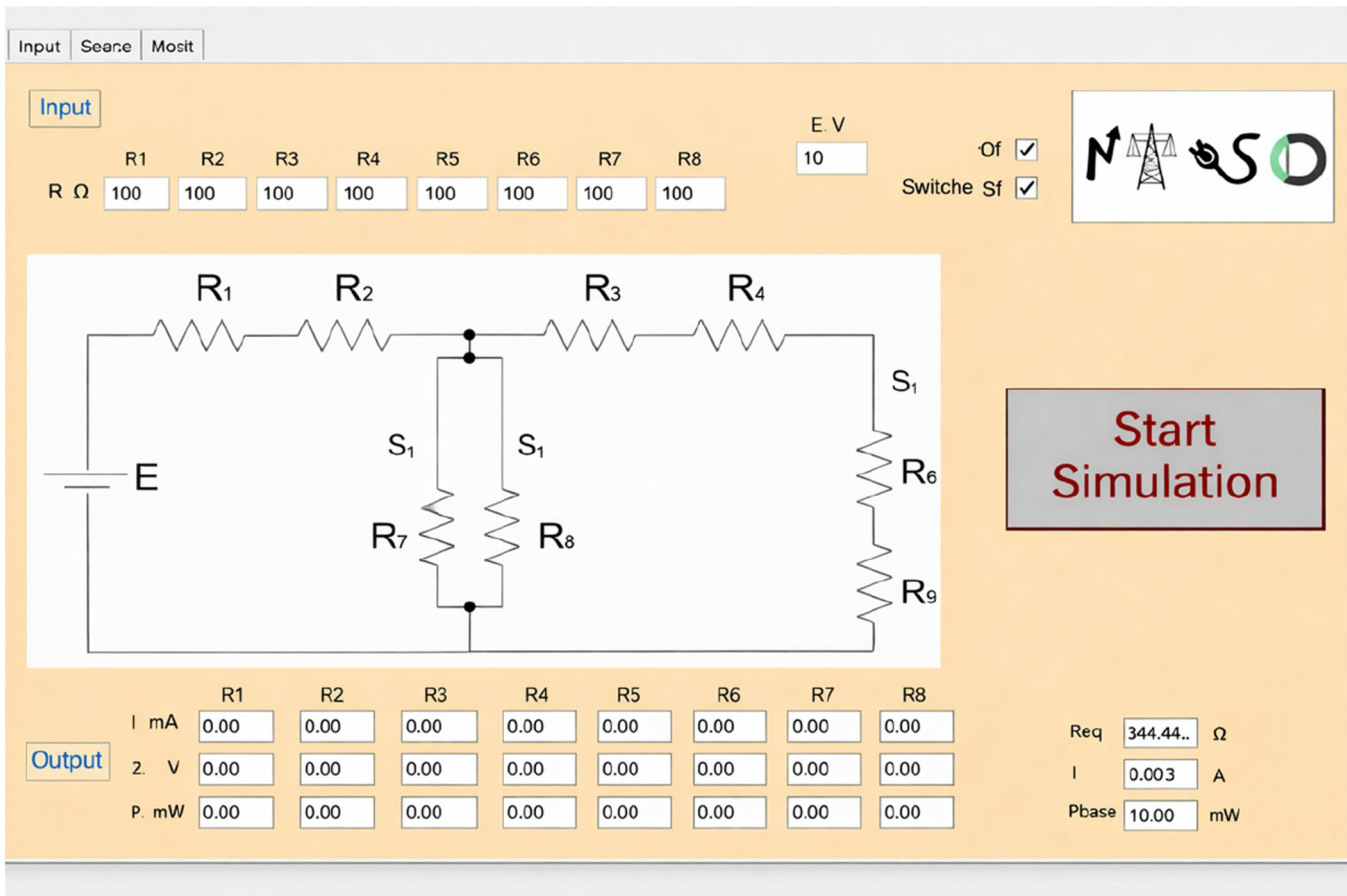


FIGURE 1 | DC circuit analysis with switches.

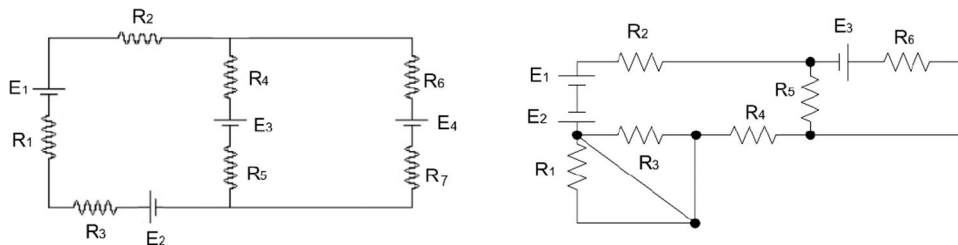


FIGURE 2 | Kirchhoff examples.

evolution of this circuit is to increase the number of resistors and batteries, such as the examples in Figure 2.

To analyse more complex circuits, the application of Ohm's Law alone is insufficient; it becomes necessary to employ Kirchhoff's circuit laws—specifically, the current law and the voltage law. Kirchhoff's laws provide a systematic framework for analysing current at circuit nodes and voltage around closed loops, making them essential for circuits involving multiple power sources and combinations of resistors in series and parallel configurations. Beyond facilitating student engagement and enhancing the clarity of explanations, the use of this analytical approach expands the range of solvable problems. Furthermore, it aids in identifying limitations or inaccuracies in the tools developed by the instructor, while also highlighting opportunities for the creation of new, more advanced educational tools.

Some examples of exercises in Figure 3. circuits are:

Assuming that it is intended to maintain the current source, what changes in the other resistances would you make when:

- the value of R1 decreases 10%?
- the value of R3 decreases 10%?
- the value of R7 decreases 10%?
- the value of R8 decreases 10%?
- What is the evolution of the current in R7 when the value of R7 decreases?
- What is the evolution of the voltage drop in R2, R3, R7 and R8 when the value of R7 decreases?

- What is the evolution of the power P7 when the value of R7 decreases?
- What is the evolution of the current in R3 when the value of R3 decreases?
- What is the evolution of the voltage drop in R2, R3, R5, R7 and R8 when the value of R3 decreases?
- Switch off S3. What is the voltage drop in R3 and in R4? Make considerations.
- Observe, make calculations and comment on what happens with all currents and voltage drops when handling the switches on/off.

With these tools, the problems were more ambitious but more visual, leaving more time to debate and allowing them to go further and deeper in knowledge acquired.

To ensure transparency and to substantiate the technical implementation of CT within the intervention, the relationship between CT dimensions, tool functionality, student activities and assessment evidence is summarised in Table 1. The table makes explicit how each CT dimension was operationalised through concrete technical features of the developed digital tools and embedded within PBL tasks, rather than being treated as an implicit or abstract outcome.

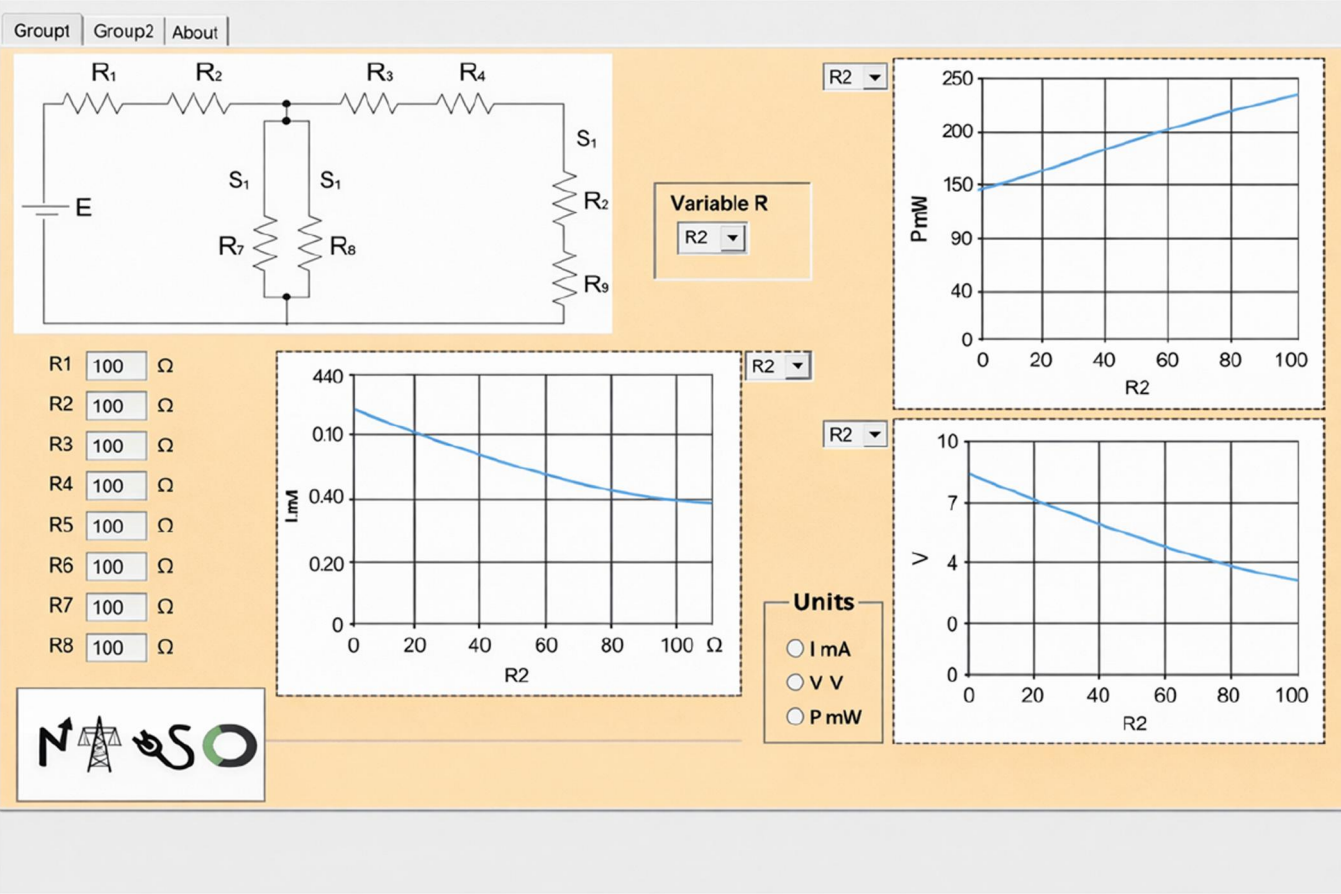


FIGURE 3 | Influence of resistance tolerance and resistance change.

TABLE 1 | CT-implementation list.

CT dimension	Technical implementation in tools	Student activity	Assessment evidence
Abstraction	Parameter selection, suppression of non-relevant circuit elements	Identify relevant variables and assumptions	Scenario-based CT test items
Decomposition	Circuit partitioning into nodes, loops and sub-circuits	Analyse sub-circuits independently	Structured problem solutions
Algorithmic thinking	Stepwise application of Ohm's and Kirchhoff's laws	Develop ordered solution procedures	Pre/post CT algorithmic items
Automation	Repeated execution of solution logic via simulators	Run multiple parameter variations	Tool logs and task outputs
Generalisation	Reconfiguration of circuits and operating conditions	Transfer strategies to new circuits	Novel post-test scenarios

As shown in Table 1, each dimension of CT was deliberately elicited through specific design choices in the simulators and learning activities and was independently observable through aligned assessment instruments. This mapping provides a clear traceability between the theoretical CT framework, its technical implementation in the instructional tools, and the empirical evidence used for analysis, thereby strengthening the methodological rigour and replicability of the study.

- For DC circuit analyses, there are some tools under development, such as:
 - Capacitors and coils behaviour on charging and discharging;
 - Electric energy in capacitors;
 - Magnetic energy in coils.
- For AC circuit analyses, there are some tools under development, such as:
 - Passive filters using resistors, capacitors and coils;
 - Oscilloscope layout and manipulation;
 - Electricity transport and distribution.

The adaptation of the analyses from DC to AC, using the same model as shown in Figure 1, enhances the tool analysis, using sinusoidal sources with different voltage values (level and frequency) and impedances (resistors, capacitors and coils) as consumption.

4 | Results

Several observable shifts in students' CT practices emerged throughout the intervention phases, particularly in relation to abstraction, decomposition, algorithmic reasoning and transfer across circuit analysis tasks, as summarised in Table 2.

To provide greater analytical transparency regarding the qualitative patterns identified throughout the intervention, the relative prevalence of observed CT indicators across the different phases is presented in Table 3.

In early phases, students typically focused on isolated variables (e.g., direct substitution into Ohm's law equations) without identifying system-level relationships. By the final phase, 31 of the 42 analysed artefacts included explicit modelling assumptions and functional grouping of circuit components. Table 4 summarises the principal observation patterns and collaborative practices identified during the intervention, highlighting the progressive emergence of student autonomy, peer interaction and shared analytical reasoning across the different instructional phases.

To strengthen analytical transparency, qualitative coding frequencies were systematically documented across all intervention phases. The reported frequencies do not represent statistical measurements or inferential outcomes; rather, they correspond to the number of identifiable occurrences of predefined CT-related behaviours observed within the analysed corpus.

The analysed corpus comprised:

- 90 student-generated engineering problems and corresponding solution reports;

TABLE 2 | Observed CT-related patterns across intervention phases.

CT dimension	Early intervention phases (Weeks 1–4)	Intermediate phases (Weeks 5–10)	Late intervention phases (Weeks 11–14)	Observable pattern
Abstraction	Isolated variables and formula substitution	Variable relationships and functional circuit block	Systems-level reasoning and modelling assumptions	Progressive shift from procedural to systems-level reasoning
Decomposition	Linear circuit analysis without subproblem decomposition	Functional circuit decomposition and component interactions	Independent decomposition and strategy comparison	Structural decomposition and organised problem-solving
Algorithmic thinking	Incomplete and inconsistent solution procedures	Sequential reasoning and stepwise analytical procedures	Coherent and transferable solution procedures	Procedural coherence and explicit reasoning structure
Automation	Simulation-based answer verification	Iterative simulation for hypothesis and parameter testing	Exploratory modelling and scenario testing	Shift from verification to exploratory experimentation
Generalisation	Limited strategy transfer across circuit types	Partial transfer across structurally similar problems	Strategy adaptation to novel circuit topologies and conditions	Emerging transferable reasoning across contexts

TABLE 3 | Frequency of observed CT indicators across intervention phases.

CT dimension	Observable indicator	Early phase Weeks 1–3	Intermediate phase Weeks 4–10	Late phase Weeks 11–14	Overall trend
Abstraction	Variable identification and circuit relationships	Low	Moderate	High	Progressive development of systems-oriented reasoning
Abstraction	Functional simplification of circuit representations	Low	Moderate	High	Improved modelling coherence in later phases
Decomposition	Separation of circuits into functional sub-units	Low	Moderate	High	Increasing analytical segmentation strategies
Decomposition	Explicit articulation of analytical subproblems	Rare	Moderate	High	Increased structural organisation of problem solving
Algorithmic thinking	Sequential application of analytical procedures	Moderate	Moderate–High	High	Greater procedural coherence across phases
Algorithmic thinking	Explicit justification of procedural decisions	Rare	Moderate	High	Increasingly articulated reasoning processes
Automation	Iterative use of simulation tools for parameter testing	Low	Moderate	High	Shift from verification toward exploratory experimentation
Automation	Comparative evaluation of alternative simulation scenarios	Rare	Moderate	High	Increasing strategic use of simulation environments
Generalisation	Transfer of solution strategies across circuit configurations	Rare	Emerging	High	Increased evidence of transferable reasoning
Generalisation	Adaptation of analytical approaches to novel conditions	Rare	Low–Moderate	High	Greater flexibility in problem adaptation

TABLE 4 | Observation patterns and collaborative practices.

Observation category	Early phases	Intermediate phases	Late phases
Participation patterns	Uneven and instructor-dependent participation	Increased participation in collaborative problem solving	Distributed and student-initiated participation
Collaborative reasoning	Peer interaction for procedural clarification	Debate of alternative solutions and simulation outcomes	Sustained collaborative sense-making and peer critique
Engagement with simulation tools	Calculation verification	Parameter variation and fault analysis	Simulation-based hypothesis testing and model evaluation
Problem formulation behaviour	Reliance on instructor-provided problem structures	Partial autonomy in problem modification	Originality, structural variation and contextual adaptation
Reflective discussion	Reflection was limited and descriptive.	Connecting simulation outputs to theoretical principles	Analytical reflection on discrepancies and modelling limitations
On-task behaviour	Frequent instructor requests for procedural confirmation	Sustained autonomous work periods	Sustained independent collaborative problem solving

- 42 structured reflection artefacts;
- 28 classroom observation records collected across lecture and laboratory sessions;
- 14 group presentation and discussion sessions conducted during the intervention.

The unit of analysis was defined as a single observable occurrence of a predefined CT-related behaviour within a student artefact, problem solution, reflection report, classroom interaction, or simulation-based activity. Multiple coded occurrences could therefore be identified within a single artefact or observation session when distinct CT processes were explicitly demonstrated.

Coding was conducted using a predefined analytical rubric aligned with the five CT dimensions adopted in this study: abstraction, decomposition, algorithmic thinking, automation and generalisation. Each occurrence was coded only when explicit evidence of the target behaviour was identifiable according to the operational indicators defined in the rubric.

Examples of coding criteria included:

- abstraction: identification of relevant variables while suppressing non-essential circuit information;
- decomposition: explicit subdivision of circuits into analytical sub-problems or functional blocks;
- algorithmic thinking: articulation of ordered and coherent analytical procedures;
- automation: iterative use of simulation tools to test parameter variation or alternative scenarios;
- generalisation: transfer of analytical reasoning across structurally different circuit configurations.

All artefacts and observation records were independently coded by two researchers. Inter-rater agreement exceeded Cohen's $\kappa = 0.80$ across all coding categories. Disagreements were resolved through discussion and consensus review.

The resulting frequencies were used descriptively to support analytical triangulation and to provide greater visibility into the evolution of CT-related practices throughout the intervention phases. The analysis included 84 student-generated artefacts, 28 reflection reports and 14 structured observation sessions distributed across four intervention phases. Table 5 summarises the observed frequency of selected CT-related behaviours identified across the intervention phases through analysis of student-generated artefacts, classroom observations and problem-solving activities.

Table 6 presents the coding patterns identified in student-generated artefacts, illustrating how different CT dimensions became progressively associated with increasingly structured analytical reasoning, and problem-solving practices throughout the intervention.

To provide a comprehensive synthesis of the empirical data and establish clear methodological rigour, Table 7 outlines the direct alignment between the primary data sources, analytical foci, coding typologies and the specific research questions addressed throughout the design-based intervention. This overview maps how multi-source evidence—ranging from student-generated artefacts and reflection reports to structured classroom observations and transfer tasks—collectively supports the analytical traceability of the study.

5 | Discussion

5.1 | Pedagogical Reflections

The pedagogical design explored in this study reflects several persistent challenges in contemporary engineering education, particularly those associated with fostering conceptual understanding, sustaining student engagement and preparing graduates for increasingly complex professional environments. The findings suggest that embedding CT within problem-based and experiential learning activities may provide a useful framework for addressing these challenges. Four interconnected themes emerged from the intervention: learner agency and motivation,

TABLE 5 | Descriptive frequency of observed CT-related behaviours across intervention phases.

CT dimension/observable behaviour	Early phase (Weeks 1–4)	Middle phase (Weeks 5–9)	Final phase (Weeks 10–14)	Evidence source
Identification of relevant electrical variables	12	26	41	Student-generated problems
Explicit circuit decomposition into sub-problems	7	18	36	Student artefacts and classroom observations
Use of abstraction to simplify circuit analysis	5	17	33	Problem-solving reports
Transfer of reasoning strategies between DC and AC contexts	2	9	24	Student explanations and artefacts
Use of simulation results to validate analytical calculations	8	21	39	Multisim-based activities
Collaborative explanation of analytical procedures	10	23	37	Structured classroom observations
Formulation of original engineering problems	3	11	29	Student-generated tasks
Identification and correction of modelling inconsistencies	4	14	31	Reflection reports
Systematic comparison of alternative solution paths	2	8	22	Observation logs
Evidence of generalisation across similar circuit structures	1	6	19	Student artefacts

Note: Frequencies correspond to descriptive coding occurrences identified in student-generated artefacts, classroom observations and problem-solving activities. Frequencies correspond to coded occurrences of predefined CT-related behaviours identified within student-generated artefacts, classroom observations, reflection reports and simulation-based activities. Values are presented descriptively to support analytical transparency and triangulation rather than inferential statistical analysis.

TABLE 6 | Artefact coding and analytical patterns.

Code	CT dimension	Observable indicator	Evidence source	Occurrence
ABS-1	Abstraction	Identification of relevant electrical variables and relationships	Student-generated problems and solutions	Increased progressively across phases
ABS-2	Abstraction	Simplification of circuit representations while preserving functional behaviour	Student-generated artefacts	More evident in later phases
DEC-1	Decomposition	Separation of circuits into sub-circuits or functional blocks	Problem solutions	Moderate in intermediate phases; frequent in later phases
DEC-2	Decomposition	Explicit articulation of analytical subproblems	Student-generated solutions	Increased during student-generated problem activities
ALG-1	Algorithmic Thinking	Sequential application of analytical procedures	Solutions and observation records	Present throughout but more coherent in later phases
ALG-2	Algorithmic Thinking	Explicit justification of procedural steps	Student artefacts and classroom discussion	Increased in later intervention phases
AUT-1	Automation	Iterative use of simulation tools for parameter testing	Observation records	Limited initially; frequent in later phases
AUT-2	Automation	Use of simulations to compare alternative scenarios	Student-generated tasks	More evident during transfer activities
GEN-1	Generalisation	Transfer of solution strategies across circuit configurations	Artefacts and observations	Emerging in intermediate phases; sustained later
GEN-2	Generalisation	Adaptation of prior analytical approaches to novel conditions	Student-generated problems	Predominantly observed in final intervention phase

TABLE 7 | Empirical evidence and analytical traceability overview.

Source	Corpus/N	Unit of analysis	Analytical focus	Coding/type	Evidence	Contribution to RQs
Student-generated engineering problems	84 artefacts	Individual problem formulation and solution	CT enactment across core dimensions	Predefined CT rubric; dual coding; Cohen's $\kappa > 0.80$	Tables 5 and 6; longitudinal coding patterns; behavioural indicators	RQ1, RQ2
Student-generated solution reports	Included within artefact corpus	Stepwise analytical reasoning	Structured analytical reasoning	Qualitative coding aligned with CT dimensions	Coding examples ABS-1, DEC-1, ALG-1, GEN-1	RQ1, RQ2
Reflection reports	28 reports	Reflective analytical commentary	Conceptual reasoning and transfer	Behavioural evidence coding	Table 5—modelling inconsistencies and conceptual reasoning patterns	RQ2, RQ3
Classroom observations	14 structured sessions	Session-based behavioural observation	Collaborative engagement and participation	Structured observation protocol	Table 5 – collaborative explanation, engagement patterns	RQ1, RQ3
Simulation-based activities	Across 14-week intervention	ICT tool interactions	Automation and iterative testing	Coding and artefact analysis	AUT-1/2 patterns and validation	RQ1, RQ2
Student-generated transfer tasks	Embedded in later intervention phases	Novel circuit configurations	Generalisation and strategy transfer	CT transfer coding	Increased GEN-1 and GEN-2 occurrences across phases	RQ2, RQ3
Collaborative discussions and plenary sessions	Embedded across intervention phases	Discussion episode	Explicit articulation of analytical reasoning	Observation and artefact triangulation	Collaborative reasoning and explanation	RQ1, RQ3
Longitudinal intervention phases	Weeks 1–14	Phase comparison	Evolution of CT-related practices over time	Descriptive frequency aggregation	Early/Middle/Final phase frequencies in Table 5	All RQs
Coding reliability process	All coded materials	Inter-rater agreement	Analytical consistency and traceability	Independent dual coding	Cohen's $\kappa > 0.80$ reported in Methods	Methodological validity
Analytical triangulation	Artefacts + observations + reflections	Cross-source convergence	Interpretive validity	Multi-source triangulation	Explicitly described in Sections 3.7	Supports all interpretations

collaborative knowledge construction, curricular alignment and the pedagogical tensions associated with integrating CT into engineering education.

Student motivation remains a central factor influencing persistence, engagement and academic performance in engineering education [44]. According to self-determination theory, learning environments that support autonomy, competence and relatedness are more likely to sustain intrinsic motivation and active engagement [45]. However, traditional engineering instruction frequently prioritises procedural repetition and formula application, often limiting opportunities for students to exercise agency in the learning process [34].

The intervention described in this study attempted to address this limitation by integrating CT practices into problem-based activities grounded in authentic engineering contexts. Rather than focusing exclusively on procedural execution, students engaged in problem formulation, abstraction, decomposition and iterative reasoning processes. These practices align with widely recognised dimensions of CT in engineering and computing education [17, 19]. Classroom observations and coded artefacts suggested increasing evidence of students articulating structured reasoning processes and engaging more actively with modelling assumptions and analytical decisions throughout the intervention phases. In one student-generated task, learners reformulated a DC resistor network into a parameter-variation scenario requiring iterative testing of voltage thresholds under modified resistance values. This activity was coded as ABS-2, ALG-2 and GEN-1 because students simplified circuit behaviour, articulated sequential procedures and transferred prior analytical strategies to a novel configuration.

These observations are consistent with prior research indicating that authentic and cognitively meaningful tasks tend to encourage deeper learning approaches and sustained engagement [46]. At the same time, the findings also highlight an important pedagogical tension. Digital tools and simulation environments may support exploratory reasoning and conceptual visualisation, yet excessive dependence on automated procedures can also encourage superficial interaction with underlying engineering principles [35]. Consequently, the educational value of such tools depends not only on their availability but also on how they are pedagogically integrated into reflective and analytically demanding learning activities.

Collaborative learning constitutes a significant component of contemporary engineering education, reflecting both disciplinary practice and constructivist perspectives on knowledge development [47, 48]. Participation in collaborative learning activities may also be influenced by learners' communication practices, educational expectations and prior learning experiences, which shape how students engage with group interaction and knowledge construction [49, 50]. Engineering problems are frequently ill-structured and multidimensional, requiring negotiation of meaning, distributed reasoning and iterative refinement of ideas. Well-designed collaborative tasks can therefore provide opportunities for students to externalise reasoning processes and engage in shared problem-solving practices.

The intervention findings suggested that integrating CT dimensions into group-based activities encouraged students to articulate analytical reasoning more explicitly during discussions, simulations and project development. In several observed activities, students collectively refined decomposition strategies, debated modelling assumptions and evaluated alternative computational approaches. These interactions appeared consistent with Wenger's [51] concept of communities of practice, where participation in shared problem-solving activities contributes to professional identity formation and disciplinary enculturation.

Nevertheless, collaborative learning also presents persistent pedagogical challenges. Prior studies have shown that group work does not automatically produce equitable participation or meaningful engagement [52]. Uneven workload distribution, passive participation, and dominance by stronger students remain common concerns in engineering classrooms. Similar dynamics were occasionally observed during the intervention, particularly in tasks involving simulation-based activities where technically confident students sometimes assumed disproportionate responsibility for computational procedures.

These observations suggest that collaborative CT-oriented learning environments may benefit from additional mechanisms promoting individual accountability and reflective participation. Hybrid pedagogical models combining collective problem-solving with structured individual reflection may help balance teamwork development with personal conceptual engagement [53].

The pedagogical approach adopted in this study also reflects broader transformations in European higher education associated with the Bologna Process and outcomes-based curriculum design [54]. These developments have increasingly shifted emphasis from content transmission towards student-centred learning, transferable competencies, and active participation in knowledge construction.

Within this context, CT may provide a useful integrative framework for engineering curricula by supporting competencies that extend beyond disciplinary content alone. Abstraction, algorithmic reasoning, systems thinking and adaptability are increasingly relevant in engineering practice characterised by digitalisation, automation and interdisciplinary problem-solving. The findings of this study suggest that CT-oriented pedagogical activities may support the development of these competencies while remaining grounded in domain-specific engineering content.

The intervention also aligns with broader international educational and societal priorities, including the United Nations Sustainable Development Goals (SDGs), shown in Figure 4.

In particular, SDG 4 emphasises inclusive and equitable quality education, while SDG 9 highlights the importance of technological innovation and engineering capability in addressing societal challenges [55]. By engaging students in computationally supported problem-solving activities connected to authentic engineering contexts, the pedagogical model explored here may contribute to the development of analytical and adaptive



FIGURE 4 | United Nations sustainable goals.

competencies relevant to sustainability, digital transformation and responsible technological development.

Furthermore, contemporary labour market analyses continue to identify problem-solving, adaptability, collaboration and digital reasoning as increasingly important professional capabilities [56]. From this perspective, integrating CT into engineering education may support not only academic development but also broader professional preparedness in rapidly evolving technological environments.

The findings and reflections emerging from this study point to several enduring tensions within engineering pedagogy.

The first concerns the relationship between innovation and disciplinary rigour. Digital tools, simulations and CT-oriented activities may enhance exploratory learning and conceptual accessibility; however, educators must ensure that these approaches do not unintentionally reduce engagement with the mathematical and analytical foundations of engineering reasoning. Maintaining conceptual depth while adopting flexible and technology-supported pedagogies therefore remains a significant challenge.

The second tension relates to collaboration and accountability. While group-based problem-solving activities may encourage peer learning and distributed cognition, they also risk uneven participation and fragmented engagement if insufficiently structured. Future iterations of this pedagogical model could benefit from integrating more explicit mechanisms for individual reflection, peer evaluation and metacognitive monitoring.

A third tension concerns the balance between disciplinary specialisation and transferable competencies. Engineering curricula are already densely structured, creating challenges when incorporating cross-cutting capabilities such as CT, communication, ethical reasoning and interdisciplinary collaboration. Nevertheless, these competencies are increasingly associated with adaptive expertise and professional flexibility in contemporary engineering practice [31].

These tensions resonate with broader debates within engineering education research regarding the development of

adaptive expertise and transformative conceptual understanding. In particular, CT may function as a threshold concept within engineering education by reshaping how students conceptualise and approach complex problems across disciplinary contexts [57]. Rather than representing a discrete technical skill, CT may constitute a broader epistemic framework influencing how learners structure reasoning, model systems and engage with uncertainty. Beyond engineering education, recent developments in explainable artificial intelligence have highlighted the importance of transparent reasoning frameworks capable of exposing intermediate analytical steps rather than producing opaque conclusions [22]. The present instructional design follows a similar philosophy by making students' reasoning processes observable through coding, artefact analysis and structured reflection, thereby supporting both learning and methodological transparency.

Overall, the pedagogical approaches examined in this study should not be interpreted as universally transferable solutions or empirically validated prescriptions. Instead, the findings provide qualitative evidence suggesting that CT-oriented problem-based and experiential activities may offer a productive framework for supporting student engagement, collaborative reasoning and conceptual exploration within engineering education contexts.

5.2 | Educational and Project Guide

The project described here was conceived not merely as a set of classroom activities, but as a pedagogical model that integrates CT with experiential and PBL principles in the context of engineering education. The guide functions at two levels: as a practical sequence of learning activities for students, and as a conceptual framework for educators aiming to embed CT and collaborative problem-solving into disciplinary instruction.

The model is grounded in constructivist views of learning, which hold that knowledge is actively constructed through interaction with problems, tools and peers [47, 58]. Engineering problems are inherently complex and ill-structured, making them ideal contexts for fostering higher-order thinking.

By embedding CT dimensions—abstraction, decomposition, pattern recognition, algorithmic reasoning and generalisation within authentic engineering challenges, the model aims to cultivate both disciplinary competence and transferable cognitive skills [17, 19].

The guide unfolds through a set of phases, deliberately aligned with Kolb's experiential learning cycle:

1. Problem identification and scoping (concrete experience): Students encounter authentic, open-ended challenges designed to be both technically rigorous and societally relevant. This stage anchors learning in contexts that resonate with professional practice [35].
2. Exploration and conceptualisation (reflective observation and abstract conceptualisation): Students break down problems into sub-components, identify relevant disciplinary knowledge and abstract patterns. Digital simulations and modelling tools are introduced as supports for testing and visualising hypotheses.
3. Collaborative design and implementation (active experimentation): Teams construct solutions, iterating between theory and practice. Group dynamics are intentionally structured to promote dialogue, role negotiation and shared accountability, echoing research on effective engineering teamwork [48].
4. Reflection and meta-cognition: Students evaluate not only their solutions but also the processes by which they arrived at them. This stage reinforces CT as a threshold concept [57], requiring learners to shift from viewing problems as fixed exercises to treating them as spaces for creative reasoning.

The educator's role in this model is multifaceted:

- Learning architect: Designing problems that are open enough to stimulate enquiry yet bounded enough to avoid cognitive overload.
- Facilitator: Guiding group interactions, ensuring equitable participation and prompting metacognitive reflection.
- Evaluator: Balancing assessment of conceptual mastery, problem-solving processes and collaborative competencies.

This shift aligns with the student-centred pedagogy advocated by the Bologna Process [54] and requires professional development for educators, who must adopt new identities as designers and facilitators rather than content deliverers.

Several challenges emerged during the implementation of the model, which are critical for future adaptations:

- Technology as Scaffold versus Crutch: While computational tools enable students to visualise and iterate rapidly, they can also tempt learners to bypass conceptual reasoning. Educators must carefully calibrate tool use to reinforce, not replace, foundational understanding [35].
- Collaboration versus individual accountability: Group projects foster teamwork, but without explicit mechanisms

(e.g., peer evaluation, individual reflections), they risk free-riding and uneven participation [52].

- Authenticity versus attainability: Problems must balance real-world relevance with cognitive feasibility. Tasks that are too open-ended may overwhelm novices, while overly simplified exercises diminish authenticity [34].

This educational guide also aligns with wider policy and societal frameworks. At the European level, the Bologna Process mandates competency-based education and lifelong learning pathways, to which CT directly contributes. Globally, the approach resonates with the United Nations SDGs, particularly Goal 4 (quality education) and Goal 9 (industry, innovation and infrastructure) [55]. By equipping students with transferable skills for innovation and adaptability, the model addresses both labour market demands [56] and the ethical responsibility of engineers to contribute to sustainable development.

The guide is intentionally adaptable. Although developed in an electrical engineering context, its design principles—authentic problem orientation, iterative tool-supported enquiry, structured collaboration and reflective practice—can be applied across STEM and interdisciplinary courses. Future work should explore:

- Scaling the model to large cohorts while maintaining active participation.
- Integrating AI-supported tools to personalise learning while guarding against over-automation.
- Investigating longitudinal impacts of CT-focused curricula on students' professional trajectories and capacity for adaptive expertise [31].

5.3 | Reflection

To ensure transparency and traceability, the results are reported in direct alignment with the research questions. Each subsection explicitly addresses one research question and integrates evidence from student-generated artefacts and structured classroom observations.

The integration of CT within a PBL environment, mediated through custom ICT-supported simulators and aligned with Kolb's experiential learning cycle, generated meaningful insights into how technology can shape engineering learning in introductory electrical engineering. Rather than claiming statistical improvements, the present study offers evidence of how computational tools and structured learning activities supported students' engagement with discipline-specific reasoning and engineering problem formulation.

To directly address the research questions guiding this study, the main findings are synthesised as follows:

RQ1. *How does the integration of ICT-based problem formulation and simulation tools affect students' CT skills in the context of electrotechnics and electrical machines courses? The findings indicate that embedding ICT-based simulation and problem formulation tools within a PBL-experiential framework*

supported the explicit enactment of key CT dimensions, namely abstraction, decomposition, algorithmic thinking, automation and generalisation. Evidence from student-generated problems and structured classroom observations showed progressive shifts toward more systematic problem formulation, clearer identification of relevant variables and more coherent stepwise solution strategies across intervention phases. These results suggest that CT can be deliberately designed for and made observable within core electrical engineering tasks, rather than remaining an implicit by-product of tool use.

RQ2. *What is the impact of these tools on students' conceptual understanding of DC and AC circuit analysis? Analysis of student artefacts and classroom activity indicated improvements in how students reasoned about DC and AC circuits, particularly in their ability to apply electrical laws meaningfully across varying circuit configurations. Students increasingly demonstrated the capacity to relate analytical calculations to simulated behaviours, to interpret discrepancies between theoretical and simulated results and to transfer solution strategies from simpler to more complex circuit contexts. These patterns suggest that the ICT-supported PBL design was associated with deeper engagement with underlying electrical concepts, rather than rote application of formulas.*

RQ3. *How do students perceive the usefulness, usability and transferability of these tools to other engineering contexts? Classroom observations and reflective artefacts suggested that students used the tools as supports for exploration, experimentation and collaborative problem solving. The ability to rapidly test 'what-if' scenarios and visualise circuit behaviour was frequently associated with increased engagement and discussion. Students also demonstrated evidence of transfer by adapting solution strategies to novel circuit designs and operating conditions, suggesting that the tools were not perceived as context-bound but as broadly applicable supports for engineering reasoning.*

Together, these findings demonstrate how CT can be explicitly embedded, enacted and examined within an ICT-supported PBL environment in undergraduate engineering education.

From an instructional design perspective, embedding CT as an explicit component of the curriculum required careful alignment between tool functionality, task structure and learning phases. The ICT-based simulation environment served not merely as a delivery platform but as an interactive cognitive scaffold that foregrounded core engineering processes such as abstraction and decomposition. As students progressed through guided, structured and student-generated problem phases, their work increasingly reflected systematic approaches to framing engineering problems and articulating solution strategies within the simulator environment. These patterns were evident in the design characteristics of student-generated problems and in observed classroom interactions, where learners engaged in collaborative sense-making around simulation results and circuit behaviours.

This study contributes to the literature on computer applications in engineering education by providing a replicable design-based framework that links computational tools with

pedagogical phases and curricular goals. Whereas much prior work has emphasised the presence of technology in classroom settings, our findings highlight the importance of designing tasks that explicitly invoke CT processes and provide opportunities for reflection, conceptualisation and iterative experimentation. In doing so, the study illustrates a practical mechanism for integrating abstract cognitive competencies with authentic engineering tasks—a focus that aligns with the journal's interest in technology-enhanced engineering instruction and assessment practices.

The role of problem formulation emerged as particularly significant. When students were invited to generate their own engineering problems using the simulation tools, CT processes became more visible and served as productive loci for instructor feedback and peer discussion. This observation underscores the potential of student-driven artefacts, supported by computational environments, to function as both learning outcomes and assessment evidence within technology-mediated courses.

At the same time, the research surfaced persistent pedagogical challenges. ICT-based tools can enrich learning only when they are integrated with explicit task structures and reflection opportunities that prompt learners to connect tool interactions with underlying engineering concepts. Collaborative tasks enriched engagement but also required intentional scaffolding to ensure equitable participation. These insights resonate with broader discussions in engineering education regarding how to balance technological innovation with pedagogical coherence and learner support.

In summary, this study offers applied insights into how computational tools and thoughtfully designed problem-based activities can support engineering students' engagement with complex disciplinary reasoning. By foregrounding the interplay between ICT-supported simulation, pedagogical design and learner activity, the work contributes to the evolving understanding of how technology can be harnessed to advance engineering education.

6 | Conclusions

This study examined how CT can be explicitly embedded within an ICT-supported PBL environment structured according to Kolb's experiential learning cycle in introductory electrical engineering education. Through a 14-week design-based intervention integrating custom-developed simulation tools into electrotechnics and electrical machines courses, the study investigated how students engaged with abstraction, decomposition, algorithmic thinking, automation and generalisation during engineering problem-solving activities.

The findings indicate that the integration of simulation-supported PBL activities created conditions in which CT-related practices became observable within authentic engineering tasks. Analysis of student-generated artefacts, structured reflections and classroom observations revealed progressively more explicit decomposition of circuits, greater procedural coherence in analytical reasoning and increased transfer of solution strategies across DC and AC contexts. The simulation

environment also evolved from a verification-oriented tool into a platform for exploratory modelling, parameter variation and collaborative reasoning.

A central contribution of this work lies in demonstrating how CT can be operationalised, analysed and pedagogically integrated within foundational engineering curricula rather than treated as an implicit by-product of programming or software use. The study further contributes a replicable instructional and analytical framework linking CT dimensions, simulation-based learning activities and observable engineering problem-solving behaviours. By combining predefined coding procedures, structured observation protocols and artefact analysis, the work also strengthens methodological traceability for qualitative research in engineering education.

The findings additionally highlight the importance of aligning digital tools with explicit pedagogical structures. The educational value of simulation environments appeared to depend not only on technological functionality, but also on the extent to which activities promoted reflection, collaborative discussion, iterative experimentation and student-generated problem formulation.

Several limitations should be acknowledged. The study was conducted within a single institutional context using instructor-developed tools, and the findings remain context-dependent and qualitative in nature. Consequently, the study does not support causal or statistically generalisable claims. Future research should examine multi-institutional implementations, longitudinal development of CT practices and the integration of complementary quantitative measures capable of further strengthening analytical triangulation and transferability.

Overall, the study provides empirically grounded evidence that ICT-supported experiential and PBL environments can support the explicit enactment of CT within undergraduate electrical engineering education and offers practical guidance for the design of computationally enriched engineering learning environments.

Author Contributions

Nuno Soares Domingues: conceptualization, data curation, formal analysis, investigation, methodology, resources, writing and testing computer code, validation, visualization, writing – original draft, writing – review and editing. **Maria Cristina Costa:** conceptualization, methodology, resources, supervision, validation, writing – review and editing. **António Domingos:** conceptualization, methodology, resources, supervision, validation, writing – review and editing.

Funding

The authors have nothing to report.

Ethics Statement

All analysed data were anonymised prior to analysis and derived from routine educational activities conducted within normal course delivery. Participation involved no identifiable personal data, and the intervention was integrated into standard instructional practice.

Participants were informed of the educational purpose of the digital tool and of its pedagogical foundation in PBL, including its intended role in supporting conceptual understanding, engagement and motivation in engineering education. They were informed that the tool and associated teaching methods were being developed as part of an ongoing effort to enhance the quality of teaching and learning in engineering programmes.

Participants were also made aware that academic staff are encouraged to disseminate pedagogical research and that anonymised, aggregated insights derived from classroom implementation of this approach might be used for scholarly dissemination. It was explicitly stated that participation in tool-supported activities formed part of normal teaching practice and that no aspect of participation would influence assessment, grading, or academic standing.

No personal, sensitive, or identifiable information was collected at any stage of the study. All data were processed in anonymised form and cannot be traced to individual students.

Given the absence of identifiable personal data and the integration of the activities within routine educational practice, formal ethics approval was not required and informed consent was not necessary.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

1. S. J. Fennes, “Scenario for a Third Computer Revolution in Structural Engineering,” *Journal of the Structural Division* 97, no. ST1 (1971): 3–11.
2. J. Chang, “Computer Program Exchange: Myth and Reality,” in *Proceedings of the Second Conference on Electronic Computation* (American Society of Civil Engineers, 1960), 27–34.
3. P. B. Henderson, T. J. Cortina, J. M. Wing, and O. Hazzan, “Computational Thinking,” in *Proceedings of the 38th ACM Technical Symposium on Computer Science Education*, ed. I. Russell, S. M. Haller, J. D. Dougherty, and S. H. Rodger (Association for Computing Machinery, 2007 March), 195–196. <https://doi.org/10.1145/1227310.1227378>.
4. J. J. Lu and G. H. L. Fletcher, “Thinking About Computational Thinking,” in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, ed. S. Fitzgerald, M. Guzdial, G. Lewandowski, and S. A. Wolfman (Association for Computing Machinery, 2009 March), 260–264. <https://doi.org/10.1145/1508865.1508959>.
5. J. M. Wing, “Research Notebook: Computational Thinking—What and Why,” in *The Link*, vol. 6 (Carnegie Mellon University, 2011), 20–23. <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>.
6. F. K. Cansu and S. K. Cansu, “An Overview of Computational Thinking,” *International Journal of Computer Science Education in Schools* 3, no. 1 (2019): 17–30. <https://doi.org/10.21585/ijcses.v3i1.53>.
7. B. C. Czerkawski and E. W. Lyman, “Exploring Issues About Computational Thinking in Higher Education,” *TechTrends* 59, no. 2 (2015): 57–65. <https://doi.org/10.1007/s11528-015-0834-8>.
8. B. H. Majeed, L. F. Jawad, and H. T. Alrikabi, “Computational Thinking Among University Students,” *International Journal of Interactive Mobile Technologies* 16, no. 10 (2022): 223–236. <https://doi.org/10.3991/ijim.v16i10.30143>.

9. A. Csizmadia, P. Curzon, M. Dorling, et al., *Computational Thinking: A Guide for Teachers* (Computing at School, 2015).
10. I. Lee, F. Martin, J. Denner, et al., "Computational Thinking for Youth in Practice," *ACM Inroads* 2, no. 1 (2011): 32–37, <https://doi.org/10.1145/1929887.1929902>.
11. A. Yadav, N. Zhou, C. Mayfield, S. E. Hambrusch, and J. T. Korb, "Introducing Computational Thinking in Education Courses," in *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, ed. T. J. Cortina, E. L. Walker, L. A. Smith King, and D. R. Musicant (Association for Computing Machinery, 2011 March), 465–470. <https://doi.org/10.1145/1953163.1953297>.
12. M. Kim, J. Kim, and W. Lee, "Intellectual Disabilities and Programming: Improving Computational Thinking-Based Problem Solving," *Education and Information Technologies* 30 (2025): 12101–12141, Advance online publication. <https://doi.org/10.1007/s10639-025-12444-1>.
13. R. F. Bataineh and R. F. Bataineh, "Cultivating Computational Thinking Through STEM and AI in Early Education," in *Empowering Early Education With Computational Thinking, AI, and STEM*, ed. M. Kalogiannakis and S. Papadakis (IGI Global Scientific Publishing, 2025), 1–34, <https://doi.org/10.4018/979-8-3693-6210-5.ch001>.
14. R. Tariq, B. M. Aponte Babines, J. Ramirez, I. Alvarez-Icaza, and F. Naseer, "Computational Thinking in STEM Education: Current State-of-the-Art and Future Research Directions," *Frontiers in Computer Science* 6, Article (2025): 1480404, <https://doi.org/10.3389/fcomp.2024.1480404>.
15. C. H. Wu, Y. C. Chien, M. T. Chou, and Y. M. Huang, "Integrating Computational Thinking, Game Design, and Design Thinking: A Scoping Review on Trends, Applications, and Implications for Education," *Humanities and Social Sciences Communications* 12, no. 1 (2025): 1–12, <https://doi.org/10.1057/s41599-025-01544-7>.
16. D. A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development* (Prentice Hall, 1984).
17. J. M. Wing, "Computational Thinking," *Communications of the ACM* 49, no. 3 (2006): 33–35, <https://doi.org/10.1145/1118178.1118215>.
18. S. Grover and R. Pea, "Computational Thinking in K–12: A Review of the State of the Field," *Educational Researcher* 42, no. 1 (2013): 38–43, <https://doi.org/10.3102/0013189X12463051>.
19. V. J. Shute, C. Sun, and J. Asbell-Clarke, "Demystifying Computational Thinking," *Educational Research Review* 22 (2017): 142–158, <https://doi.org/10.1016/j.edurev.2017.09.003>.
20. D. E. Knuth, *The Art of Computer Programming: Vol. 1. Fundamental Algorithms* (Addison-Wesley, 1997). 3rd ed.
21. B. Kules, "Computational Thinking Is Critical Thinking: Connecting to University Discourse, Goals, and Learning Outcomes," *Proceedings of the Association for Information Science and Technology* 53, no. 1 (2016): 1–6, <https://doi.org/10.1002/prat.2016.14505301044>.
22. N. S. Domingues, "From Engineering Principles to Healthcare Practice: A Hybrid Reasoning Framework for Transparent Clinical Decision Support," *Artificial Intelligence in Health* 3, no. 2 (2026): 025470102, <https://doi.org/10.36922/AIH025470102>.
23. F. J. Agbo, S. S. Oyeler, J. Suhonen, and S. E. Adewumi, "A Systematic Review of Computational Thinking Approach for Programming Education in Higher Education Institutions," in *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, ed. P. Ithantola and N. Falkner (Association for Computing Machinery, 2019 November), 1–10, <https://doi.org/10.1145/3364510.3364518>.
24. F. J. García-Peñalvo and J. Cruz-Benito, "Computational Thinking in Pre-University Education," in *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16)*, ed. F. J. García-Peñalvo (Association for Computing Machinery, 2016 November), 13–17. <https://doi.org/10.1145/3012430.3012490>.
25. R. A. Streveler, T. A. Litzinger, R. L. Miller, and P. S. Steif, "Learning Conceptual Knowledge in the Engineering Sciences: Overview and Future Research Directions," *Journal of Engineering Education* 97, no. 3 (2008): 279–294, <https://doi.org/10.1002/j.2168-9830.2008.tb00979.x>.
26. B. Batni and S. N. Junaini, "Redefining Computational Thinking: Synergizing Unplugged Activities With Block-Based Programming," *Education and Information Technologies* 30, no. 2 (2025): 2361–2388, <https://doi.org/10.1007/s10639-024-12832-5>.
27. X. Zhang, Y. Chen, L. Hu, G. J. Hwang, and Y. F. Tu, "Developing Preschool Children's Computational Thinking and Executive Functions: Unplugged vs. Robot Programming Activities," *International Journal of STEM Education* 12, no. 1 (2025): 10, <https://doi.org/10.1186/s40594-025-00677-2>.
28. J. A. Coménio, *Didáctica magna: Tratado da arte universal de ensinar tudo a todos* (J. F. Gomes, Trad.) (Fundação Calouste Gulbenkian, 1985).
29. J. H. Pestalozzi, *Cómo Gertrudis enseña a sus hijos* (J. T. Sepúlveda, Trans.), 1889, León.
30. D. A. Kolb, "The Kolb Learning Style Inventory," in *Hay Resources Direct* (2007).
31. T. Litzinger, L. R. Lattuca, R. Hadgraft, and W. Newstetter, "Engineering Education and the Development of Expertise," *Journal of Engineering Education* 100, no. 1 (2011): 123–150, <https://doi.org/10.1002/j.2168-9830.2011.tb00006.x>.
32. M. Borrego and J. Bernhard, "The Emergence of Engineering Education Research as an Internationally Connected Field of Inquiry," *Journal of Engineering Education* 100, no. 1 (2011): 14–47, <https://doi.org/10.1002/j.2168-9830.2011.tb00003.x>.
33. C. E. Hmelo-Silver, "Problem-Based Learning: What and How Do Students Learn?," *Educational Psychology Review* 16, no. 3 (2004): 235–266, <https://doi.org/10.1023/B:EDPR.0000034022.16470.f3>.
34. M. J. Prince and R. M. Felder, "Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases," *Journal of Engineering Education* 95, no. 2 (2006): 123–138, <https://doi.org/10.1002/j.2168-9830.2006.tb00884.x>.
35. D. H. Jonassen, *Learning to Solve Problems: A Handbook for Designing Problem-Solving Learning Environments* (Routledge, 2014).
36. A. A. Tawfik, A. C. Graesser, J. Gatewood, and J. Gishbaugher, "The Effects of Problem-Based, Project-Based, and Case-Based Learning on Students' Motivation: A Meta-Analysis," *Educational Psychology Review* 36, Article (2024): 29, <https://doi.org/10.1007/s10648-024-09864-3>.
37. Y. Lu and C. K. S. Singh, "The Effectiveness of the PBL Teaching Model on the Achievement and Critical Thinking Skills Development of Chinese Undergraduate Students: A Systematic Review," *Journal of Higher Education Theory and Practice* 24, no. 3 (2024): 207–218, <https://doi.org/10.33423/jhetp.v24i3.6841>.
38. S. Lavado-Anguera, P. J. Velasco-Quintana, and M.-J. Terrón-López, "Project-Based Learning (PBL) as an Experiential Pedagogical Methodology in Engineering Education: A Review of the Literature," *Education Sciences* 14, no. 6 (2024): 617, <https://doi.org/10.3390/educsci14060617>.
39. T. Anderson, *Theory and Practice of Online Learning* (Athabasca University Press, 2004).
40. W. Richardson, *Blogs, Wikis, Podcasts, and Other Powerful Web Tools for Classrooms* (Corwin Press, 2006).
41. M. C. Costa, A. Domingos, and V. Teodoro, "Promoting Integrated STEM Tasks in the Framework of Teachers' Professional Development in Portugal," in *Integrated Approaches to STEM Education*, eds. J. Anderson and Y. Li (Springer, 2020), 553–571. https://doi.org/10.1007/978-3-030-52229-2_27.
42. L. H. Jamieson and J. R. Lohmann, "Innovation With Impact: Creating a Culture for Scholarly and Systematic Innovation in

- Engineering Education,” *Journal of Engineering Education* 101, no. 4 (2012): 363–397, <https://doi.org/10.1002/j.2168-9830.2012.tb00056.x>.
43. S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas* (Basic Books, 1980).
44. J. S. Eccles and A. Wigfield, “Motivational Beliefs, Values, and Goals,” *Annual Review of Psychology* 53 (2002): 109–132, <https://doi.org/10.1146/annurev.psych.53.100901.135153>.
45. E. L. Deci and R. M. Ryan, “The “What” and “Why” of Goal Pursuits: Human Needs and the Self-Determination of Behavior,” *Psychological Inquiry* 11, no. 4 (2000): 227–268, https://doi.org/10.1207/S15327965PLI1104_01.
46. N. Entwistle and P. Ramsden, *Understanding Student Learning* (Routledge, 2015). 2nd ed.
47. L. S. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes* (Harvard University Press, 1978).
48. M. Borrego, J. Karlin, L. D. McNair, and K. Beddoes, “Team Effectiveness Theory From Industrial and Organizational Psychology Applied to Engineering Student Project Teams: A Research Review,” *Journal of Engineering Education* 102, no. 4 (2013): 472–512, <https://doi.org/10.1002/jee.20023>.
49. G. Hofstede, *Cultures and Organizations: Software of the Mind* (McGraw-Hill, 1997).
50. B. Brock-Utne, “Education for All: In Whose Language?,” *Oxford Review of Education* 27, no. 1 (2001): 115–134, <https://doi.org/10.1080/03054980125308>.
51. E. Wenger, *Communities of Practice: Learning, Meaning, and Identity* (Cambridge University Press, 1998).
52. B. Oakley, R. M. Felder, R. Brent, and I. Elhajj, “Turning Student Groups Into Effective Teams,” *Journal of Student Centered Learning* 2, no. 1 (2004): 9–34.
53. L. K. Michaelsen and M. Sweet, “The Essential Elements of Team-Based Learning,” *New Directions for Teaching and Learning* 2008, no. 116 (2008): 7–27, <https://doi.org/10.1002/tl.330>.
54. B. Wächter, “The Bologna Process: Developments and Prospects,” *European Journal of Education* 39, no. 3 (2004): 265–273, <https://doi.org/10.1111/j.1465-3435.2004.00181.x>.
55. United Nations, *Global Sustainable Development Report 2019: The Future Is Now—Science for Achieving Sustainable Development* (United Nations, 2019).
56. World Economic Forum, *The Future of Jobs Report 2020* (World Economic Forum, 2020).
57. J. H. F. Meyer and R. Land, “Threshold Concepts and Troublesome Knowledge (2): Epistemological Considerations and a Conceptual Framework for Teaching and Learning,” *Higher Education* 49, no. 3 (2005): 373–388, <https://doi.org/10.1007/s10734-004-6779-5>.
58. J. Piaget, *Science of Education and the Psychology of the Child* (Orion Press, 1970).

Appendix A: Computational Thinking Model and Operationalisation

Rationale for Dimension Selection

The final CT model comprises five dimensions adapted from Shute et al. [19]:

- Abstraction: The ability to identify essential variables, relationships and constraints in complex electrical systems while suppressing irrelevant detail.

- Decomposition: The ability to break down a circuit or system into manageable sub-components (e.g., nodes, loops or functional modules).
- Algorithmic thinking: The ability to formulate ordered, logical procedures for solving problems, including the systematic application of electrical laws.
- Automation: The ability to express solution logic in a repeatable form supported by computational tools, while maintaining conceptual control over the process.
- Generalisation: The ability to transfer solution strategies to novel circuit configurations or operating conditions.

These dimensions collectively represent a coherent operationalisation of CT that is appropriate for undergraduate electrical engineering contexts.

Operationalisation and Indicators

For each CT dimension, observable indicators, task types and scoring criteria were defined to support reliable assessment:

- Abstraction: Identification of relevant circuit elements and variables; justification of modelling choices.
- Decomposition: Logical segmentation of circuits into analysable sub-units.
- Algorithmic Thinking: Coherent sequencing of analytical steps and correct application of electrical laws.
- Automation: Consistent and appropriate use of simulation tools to execute predefined solution logic.
- Generalisation: Successful adaptation of solution strategies to modified or novel circuit problems.

Scoring rubrics were developed iteratively and applied consistently across pre- and post-intervention assessments. Inter-rater agreement exceeded accepted thresholds, supporting the reliability of the CT measurement.