



Volume 23 (2026), pp. 205-228

American Journal of STEM Education:

Issues and Perspectives

eISSN 3065-1190 | Print ISSN: 3069-0072

<https://doi.org/10.32674/aga4kx82>

Balancing Theory and Computational Tools in Undergraduate Engineering Education

Thathsarani D.H. Herath Mudiyansele

Kansas State University, USA

<https://orcid.org/0000-0002-6638-8760>

ABSTRACT

The growing use of computational tools in engineering education creates opportunities for applied learning but also raises concerns that software proficiency may be prioritized over theoretical understanding. This qualitative descriptive study examined how undergraduate students in water resources and hydraulic engineering courses demonstrated theoretical understanding, theory-tool integration, and engineering judgment across laboratory assignments. De-identified submissions collected with institutional review board approval were analyzed using a three-level coding framework. Results showed stronger performance in structured and software supported activities, although successful tool use did not always reflect strong conceptual understanding. Engineering judgment was most evident when assignments explicitly required interpretation, feasibility assessment, and justification. The findings support laboratory designs that connect analytical reasoning, computational tools, verification, and engineering decision making.

Keywords: Computational tools, engineering education, engineering judgment, hydraulic engineering, theoretical understanding, water resources engineering

© Author(s), 2026. Published by Star Scholars Press.

This article is distributed under the Creative Commons Attribution 4.0

International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are

INTRODUCTION

Computational tools now play a central role in engineering modeling, simulation, data analysis, and design. Their use in undergraduate courses allows students to investigate realistic systems, evaluate multiple operating conditions, and complete analyses that would be difficult to perform entirely by hand. The increasing importance of computational thinking in engineering education has also created opportunities for curricular innovation and applied learning (Hidayat et al., 2024). Engineering model-building activities can engage students in abstraction, decomposition, algorithmic thinking, evaluation, and generalization, all of which are relevant to contemporary engineering practice (Lyon & Magana, 2021). Recent systematic review work has further emphasized the need to identify and assess computational thinking explicitly within engineering contexts rather than infer such thinking solely from successful completion of a computational task (Ravan & Huang, 2024).

The growing visibility of computational tools also creates an instructional challenge. Students may be able to construct a model, operate software, and obtain numerical results without clearly understanding the governing equations, assumptions, or physical behavior represented by the model. Engineering competence requires more than procedural proficiency. It includes conceptual understanding, the ability to apply technical knowledge in context, and the judgment needed to determine whether a result is reasonable and useful (Giuliano et al., 2022; Litzinger et al., 2011). Computational tools are therefore most educationally valuable when they extend engineering reasoning rather than replace it.

This challenge is particularly important in engineering laboratory instruction, where students are expected to connect theory, practice, and application, interpret results, evaluate assumptions, and develop professional judgment. Recent research on project and challenge-based engineering education similarly emphasizes learning environments in which students apply disciplinary knowledge to authentic and open-ended problems, while instructional structure and task design shape the kinds of reasoning students are required to demonstrate (Doulougeri et al., 2024; Lavado-Anguera et al., 2024). Although prior research has examined computational thinking, modeling, and simulation in engineering education, less attention has been given to how students' ordinary course submissions reveal the relationship among theoretical understanding, computational tool use, and engineering judgment. Examining student work across different assignment

formats can help identify whether students use computational tools as supports for reasoning or primarily as mechanisms for producing final outputs.

The present study examines student laboratory work from undergraduate water resources engineering and hydraulic engineering courses. The selected assignments represented different levels of analytical and computational support, including theory-focused calculations, spreadsheet-supported analysis, software-based analysis, design activities, and planning presentations. The study investigates how students applied governing principles, connected computational results to physical meaning, and demonstrated judgment through their discussion of assumptions, limitations, reasonableness, and design decisions. The study was guided by the following research questions:

1. How do students demonstrate theoretical understanding in computationally supported and theory-focused laboratory assignments in water resources engineering and hydraulic engineering courses?

2. What evidence of engineering judgment appears in students' interpretation of results, including their discussion of assumptions, limitations, reasonableness, and physical meaning?

3. How can laboratory assignment design better support the integration of computational tool use with conceptual understanding?

LITERATURE REVIEW

Theoretical Understanding and Engineering Judgment

Engineering education has traditionally treated conceptual knowledge as a foundation for professional practice. Sheppard et al. (2009) described engineering preparation as requiring the integration of analytical knowledge, practical experience, and professional responsibility. Litzinger et al. (2011) similarly argued that engineering expertise develops through deep conceptual knowledge, fluent technical performance, and engagement with authentic problems rather than procedural performance alone.

Engineering judgment extends beyond the correct application of equations. It requires students to evaluate evidence, consider context, recognize uncertainty, and make defensible decisions. Giuliano et al. (2022) associated engineering judgment with critical thinking and the ability to evaluate technical work within its practical context. In computational assignments, evidence of judgment may appear when students question an output, assess whether results are reasonable, identify assumptions or limitations, compare alternatives, or justify a design recommendation.

This dimension is important because numerical outputs generated by engineering software may appear authoritative. Without sufficient conceptual understanding, students may accept results without recognizing unrealistic values,

incorrect inputs, inappropriate assumptions, or limitations in the model. Theoretical understanding therefore provides a basis for evaluating computational results rather than simply obtaining them.

The relationship among knowledge, interpretation, and application is also consistent with broader descriptions of STEM literacy. Tenney et al. (2023) framed STEM literacy as involving not only technical content but also the ability to interpret and apply knowledge meaningfully. In engineering education, this includes understanding what a computational model represents, determining whether its assumptions are appropriate, and deciding how its results should influence an engineering decision.

Computational Tools and Conceptual Learning

Computational tools can support meaningful engineering learning by allowing students to represent systems, test relationships, evaluate outcomes, and revise their understanding. Lyon and Magana (2021) found that engineering model-building activities can elicit several computational thinking practices, including abstraction, decomposition, algorithmic thinking, evaluation, and generalization. Similarly, Magana and Silva Coutinho (2017) emphasized that modeling and simulation should remain connected to disciplinary knowledge, experimentation, and engineering judgment rather than being treated only as technical competencies. More recent work has extended this concern to the assessment of computational thinking in engineering education, where evidence of abstraction, decomposition, algorithmic reasoning, evaluation, and related practices must be elicited through appropriate learning and assessment activities rather than assumed from the production of a computational result (Ravan & Huang, 2024).

Computational thinking interventions have also been associated with improved academic achievement in engineering education, although their effectiveness depends on student preparation and instructional design (Hidayat et al., 2024). These findings support the inclusion of computational tools in engineering curricula but do not indicate that tool use alone produces conceptual learning. A four-year multiple-case study in engineering education likewise emphasized the importance of a student-centered balance between technology and teaching practices when considering the learning outcomes associated with digital education (Álvarez Ariza & Hernández Hernández, 2026). The educational value of a computational activity depends on what students are required to explain, interpret, and verify while completing it. Recent work in civil engineering education has similarly proposed pairing interactive simulations with explicit scaffolding and assessment to support difficult conceptual transitions and make student reasoning more visible (Zhang, 2026). Research on newer technology-supported learning environments similarly suggests that the instructional value of

a digital tool depends on how learners interact with it and how understanding is elicited and assessed. For example, immersive and AI-supported environments can provide new forms of interaction and representation, but their educational effectiveness remains closely tied to instructional and assessment design (Luo et al., 2025; Pande & Jepsen, 2025).

Student explanations can provide evidence of whether computational work is connected to underlying engineering principles. Vieira et al. (2019) demonstrated that explanations in computational science and engineering environments can reveal how students connect computational procedures, models, and disciplinary concepts. Moore et al. (2013) similarly emphasized the importance of representational fluency, noting that engineering students use equations, graphs, diagrams, and models to express and refine conceptual understanding. This emphasis remains evident in newer computational learning environments. Recent extended reality research, for example, has examined students' open-ended conceptual explanations rather than relying solely on successful interaction with the technology as evidence of understanding (Feijoo-Garcia et al., 2025).

Specialized engineering software allows students to analyze systems that would otherwise require extensive manual computation. However, automated solution procedures can reduce the visibility of intermediate calculations and governing relationships. A correct model output may therefore demonstrate successful tool operation without necessarily demonstrating that the student understands why the result is physically valid. This distinction is especially important when students are learning professional software while still developing their understanding of the principles implemented by that software. Recent systematic review work on immersive technologies in engineering education has likewise emphasized the need to integrate technological development with explicit pedagogical frameworks and constructive alignment rather than treating the digital environment primarily as a technical intervention (Bezzina & Zammit, 2026). Emerging engineering education environments based on augmented reality, virtual reality, and digital twins extend this same instructional opportunity and challenge. These technologies can make complex systems and otherwise difficult to observe phenomena more accessible and interactive, but their contribution to conceptual learning is strengthened when digital representations remain explicitly connected to physical systems, disciplinary models, and interpretation (Lampropoulos et al., 2025; Zhou et al., 2024).

Computational tools can consequently serve two different educational roles. They may function primarily as tools for producing results, or they may support investigation, comparison, verification, and interpretation. The latter role is more consistent with balanced engineering learning because students remain responsible for understanding the model, evaluating its assumptions, and interpreting its outputs.

Laboratory Learning and Assignment Design

Engineering laboratories provide an appropriate setting for examining the relationship among theoretical understanding, computational tools, and engineering judgment. Feisel and Rosa (2005) identified laboratory instruction as an environment in which students connect theory and practice, interpret data, investigate uncertainty, and develop professional skills. Feisel and Peterson (2002) similarly emphasized inquiry, interpretation, and judgment as important objectives of engineering laboratory experiences. More recent higher education and engineering education research has extended these principles through active, project-based, and challenge-based instructional approaches. These approaches emphasize student engagement with authentic tasks, application of disciplinary knowledge, iterative problem solving, and opportunities to connect theoretical knowledge with realistic decisions. At the same time, the literature indicates that learning outcomes depend substantially on how activities are structured, scaffolded, and aligned with the intended forms of student reasoning (Doulougeri et al., 2024; Lavado-Anguera et al., 2024; Li et al., 2023).

Laboratory assignments can reveal different forms of student understanding depending on how their deliverables are structured. An activity requiring only a completed model, screenshot, or numerical result primarily provides evidence of procedural completion. In contrast, an assignment that requires students to explain governing principles, predict system behavior, verify selected outputs, discuss assumptions, diagnose problems, or recommend modifications provides more direct evidence of conceptual understanding and engineering judgment. This distinction is consistent with recent engineering education research showing that authentic and open-ended learning activities can support knowledge application and complex problem solving, but that the intended forms of reasoning must be deliberately incorporated into the activity and its expected outputs (Doulougeri et al., 2024; Lavado-Anguera et al., 2024).

Experiential and representational approaches to STEM learning further suggest that students benefit from opportunities to connect abstract principles with applied contexts (Glancy & Moore, 2013). In computationally supported laboratories, this connection may be strengthened when students first establish an analytical expectation, use a computational tool to investigate the system, and then return to the underlying theory to interpret or verify the result. Recent applications of immersive and digitally mediated engineering learning similarly illustrate the value of connecting computational representations to physical systems and underlying disciplinary principles rather than treating the digital environment as an isolated learning experience (Lampropoulos et al., 2025; Zhou et al., 2024).

The literature therefore supports the educational value of computational tools while also emphasizing that their effectiveness depends on instructional design. What remains less clear is how students demonstrate the balance among theoretical

reasoning, tool use, and engineering judgment across ordinary laboratory assignments with different structures and expectations. The present study addresses this gap by examining student submissions from theory-focused, spreadsheet-supported, software-based, design-oriented, and planning activities in undergraduate water resources and hydraulic engineering courses.

METHODOLOGY

Research Design

This study used a qualitative descriptive content analysis design to examine course-based student work. The analysis focused on observable evidence in submitted calculations, written explanations, computational outputs, design reports, and presentation materials. The purpose was to identify recurring patterns in how students applied theoretical principles, used computational tools, interpreted results, and demonstrated engineering judgment. The study was not designed to compare course grades or establish causal differences among assignment types.

Participants and Instructional Context

The data were drawn from two undergraduate engineering courses: water resources engineering, offered in Fall 2025, and hydraulic engineering, offered in Spring 2026. Each course enrolled 20 students. The analysis treated the submissions from the two courses separately and did not depend on identifying whether individual students were enrolled in both courses.

Student work was produced during weekly laboratory sessions that generally lasted between two and three hours. During these sessions, students completed the assigned activities in the classroom using personal laptops. Depending on the task, students used Microsoft Excel, EPANET, presentation software, online resources, and web-based reference materials to perform calculations, develop models, analyze data, prepare graphical outputs, and interpret engineering results. Some assignments were completed individually, while others involved group reports, collaborative design work, or presentations.

Institutional Review Board approval was obtained, and student work was collected and analyzed in accordance with the approved protocol. All submissions were de-identified before analysis to protect student confidentiality.

Data Sources and Assignment Selection

The data sources consisted of de-identified student artifacts from six selected laboratory assignment types. These included 20 individual Hardy Cross

submissions, five EPANET group submissions, five comprehensive water-distribution design project submissions, 20 individual runoff-modeling submissions, 20 individual storm-sewer design submissions, and water-crisis presentation materials completed by teams of four to five students.

For each selected assignment, all available submissions that contained sufficient completed work to be evaluated through the normal course grading process were included in the analysis. No submissions were excluded based on grade, quality of performance, or demonstrated level of understanding.

Depending on the assignment, the artifacts included analytical calculations, Excel worksheets, EPANET model outputs and screenshots, plots, design tables, written explanations, engineering reports, and presentation slides. Individual submissions were treated as individual units of analysis. Each group report or presentation was treated as one unit of analysis.

Because the unit of analysis differed across assignments, evidence observed in individual and group submissions should not be interpreted as directly equivalent. Group artifacts reflected the collective work of multiple students and may have benefited from peer collaboration, division of responsibilities, or stronger contributions from particular group members. Consequently, group submissions may mask variation in individual conceptual understanding, whereas individual submissions provide more direct evidence of each student's performance.

The assignments were purposefully selected to represent different relationships between theoretical reasoning and computational tool use. The selected set included theory-focused analytical work, spreadsheet-supported calculations, specialized engineering software, an open-ended design project, structured analytical design activities, and a planning presentation. The aim was to examine a range of instructional approaches rather than every laboratory completed in the two courses.

Several assignments formed an intentional instructional sequence. In hydraulic engineering, students first applied conservation of mass and energy to a closed-loop pipe network using the Hardy Cross method and Excel. They subsequently modeled a comparable network in EPANET, compared software-generated results with the earlier analytical results, and examined how changes in demand and pipe availability affected pressures and flow directions.

The comprehensive water distribution project extended this sequence by requiring students to estimate demand, document assumptions, establish a preliminary network layout, construct an EPANET model, diagnose pressure problems, test modifications, and prepare an engineering recommendation. The runoff modeling and storm sewer assignments in water resources engineering provided comparison activities in which the governing analytical procedures remained visible while Excel supported calculations and plotting. The water crisis presentation emphasized evaluation of alternatives, planning, communication, and

recommendation rather than specialized modeling software. Table 1 summarizes the selected assignments, evidence sources, and classifications.

Table 1
Selected Assignments and Their Classification

Assignment	Course and evidence source	Primary activity and tools	Classification
Hardy Cross pipe-network analysis	Hydraulic Engineering; 20 individual submissions	Apply conservation of mass and energy through iterative flow correction using Excel and analytical calculations	Theory-focused with spreadsheet support
EPANET pipe network modeling	Hydraulic Engineering; five group submissions	Reproduce and modify a previously analyzed network using EPANET and Excel	Integrated theory and software activity
Comprehensive water distribution design project	Hydraulic Engineering; five group submissions	Develop a conceptual system, construct a model, diagnose performance, and recommend modifications using EPANET, Excel, calculations, sketches, and web resources	Design-based theory and tool integration
Runoff modeling	Water Resources Engineering; 20 individual submissions	Apply unit hydrograph theory and generate runoff hydrographs using Excel, plotting, and analytical calculations	Analytical activity with general computational support
Storm sewer design	Water Resources Engineering; 20 individual submissions	Select pipe sizes and slopes through repeated hydraulic calculations using Excel and analytical calculations	Analytical design with general computational support
Water crisis planning presentation	Water Resources Engineering; group presentation materials from teams of four to five students	Develop, compare, and communicate water management alternatives using web resources and presentation software	Engineering judgment and communication

Data Analysis and Coding Framework

The submissions were reviewed according to three analytical dimensions: theoretical understanding, theory-tool integration, and engineering interpretation and judgment. Theoretical understanding referred to the correct application of governing equations, conservation principles, analytical procedures, and relevant physical concepts. Theory-tool integration referred to whether students connected spreadsheet or software results to the engineering principles represented by the calculations or model. Engineering interpretation and judgment included the ability to explain physical meaning, assess reasonableness, identify assumptions and limitations, diagnose system behavior, and justify design decisions or recommendations.

To support consistency in the review, the same predefined coding definitions were applied across all submissions, and classifications were based on observable evidence in the submitted artifacts rather than informal classroom interactions. In addition, the researcher independently recoded a randomly selected subset of approximately 20% of the analyzed submissions using the same coding framework without consulting the original classifications. For each recoded submission, the ratings assigned to each of the three analytical dimensions, theoretical understanding, theory and tool integration, and engineering interpretation and judgment, were compared with the original ratings. Percentage agreement was calculated as the number of matching ratings across the three analytical dimensions divided by the total number of ratings compared, resulting in 92% agreement. Any discrepancies were reviewed against the predefined coding definitions before the final classifications were confirmed. No independent second coder or formal inter-rater reliability procedure was used.

The analytical dimensions were informed by literature on conceptual understanding, computational modeling and theory-tool integration, and engineering judgment (Giuliano et al., 2022; Litzinger et al., 2011; Lyon & Magana, 2021; Vieira et al., 2019). The descriptive evidence levels of limited, developing, and clear were developed for the present study to operationalize these dimensions based on observable features of the submitted work. Table 2 summarizes the definitions used for these three evidence levels. The code-level definitions were developed by translating the three analytical dimensions into observable features of the student artifacts. Limited evidence was used when submissions primarily presented calculations or outputs with substantial errors or little conceptual explanation. Developing evidence represented partial or incomplete application, interpretation, or connection between theory and computational work. Clear evidence was assigned when the relevant engineering principles were applied correctly and the submission explicitly connected calculations or computational outputs to physical interpretation or engineering

decisions. These levels were used as descriptive categories rather than as standardized proficiency levels or numerical grades.

Table 2
Descriptive Coding Levels Used Across the Three Analytical Dimensions

Level	Description
Limited evidence	The submission primarily reported calculations, outputs, or required information, with substantial errors or little conceptual explanation.
Developing evidence	The submission demonstrated partial understanding or interpretation but contained errors, incomplete explanations, or a limited connection between theory and tool use.
Clear evidence	The submission correctly applied the relevant principles and clearly connected calculations or computational outputs to physical interpretation and engineering decisions.

Terms used in the Results and Discussion to describe relative performance or success, such as stronger, more successful, or competent, refer to the observable evidence represented by the coding framework and assignment requirements rather than to course grades. Stronger evidence indicates more complete and correct application of relevant engineering principles, clearer connections between theory and computational outputs, or more developed interpretation and justification, depending on the analytical dimension being discussed.

The primary researcher reviewed the de-identified submissions using these definitions. For each assignment, the predominant level of evidence within each analytical dimension and recurring qualitative patterns across the submissions were recorded. The predominant evidence level was determined from the overall distribution of coded submissions. When one evidence level characterized most submissions, that level was reported as predominant. When substantial variation occurred across two adjacent evidence levels, both levels were reported descriptively. Recurring qualitative patterns were identified when similar strengths, difficulties, or forms of reasoning appeared across multiple submissions rather than in an isolated submission, and these were summarized in the predominant-pattern column of Table 3. The categories were used as descriptive indicators rather than numerical grades. When an analytical dimension was not a central feature of an assignment, it was identified as not being the principal focus rather than being assigned an artificial rating.

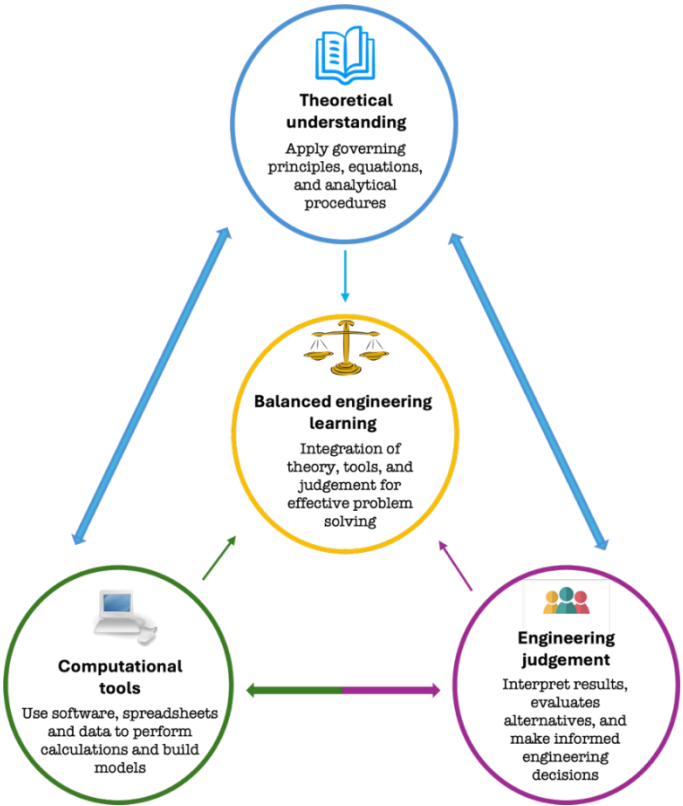
No statistical comparisons were performed because the assignments differed in objectives, complexity, format, and unit of analysis. The analysis instead examined recurring patterns across theory-focused, software-supported, design-based, and communication-oriented assignments. Particular attention was given to

whether students connected computational procedures and outputs to governing principles, physical meaning, assumptions, limitations, and engineering decisions.

Informal student comments made during laboratory sessions were not included in the formal coding. Where relevant, they were used only as contextual instructor observations and were clearly distinguished from findings derived from the submitted work.

Figure 1 illustrates the conceptual relationship among the three dimensions used in the analysis.

Figure 1
Conceptual Progression of the Selected Laboratory Assignments from Foundational Theory to Computationally Supported Engineering Judgment



RESULTS

The descriptive review identified different patterns across theory-focused, software-supported, design-oriented, and communication-based assignments. Table 3 summarizes the predominant evidence observed for each activity. The

combinations of the three evidence levels are used descriptively when submissions within an assignment showed variation across adjacent levels. For example, “Limited to developing evidence” indicates that submissions ranged between the Limited and Developing evidence levels, while “Developing evidence, with some clear evidence” indicates that Developing evidence was predominant but some submissions demonstrated Clear evidence. These descriptions do not represent additional coding levels.

Table 3
Descriptive Summary of Evidence Across the Selected Laboratory Assignments

Assignment	Theoretical understanding	Theory-tool integration	Engineering interpretation and judgment	Predominant pattern
Hardy Cross pipe network analysis	Limited to developing evidence	Limited to developing evidence	Limited evidence	Frequent errors in the iterative procedure and difficulty connecting calculations to system behavior.
EPANET pipe network modeling	Developing evidence, with some clear evidence	Developing to clear evidence	Developing evidence	Software operation was generally stronger than explanation of the underlying hydraulic behavior.
Comprehensive water distribution design project	Developing evidence	Developing to clear evidence	Developing to clear evidence	Strongest work connected assumptions, calculations, model results, and recommendations.
Runoff modeling	Developing evidence	Developing evidence	Developing evidence	Students generally completed the calculations but varied in their discussion of assumptions and limitations.
Storm sewer design	Developing evidence	Developing evidence	Developing evidence	Structured procedures supported completion, although explanations of

				design choices varied.
Water crisis planning presentation	Not the principal focus	Not the principal focus	Developing evidence	Groups followed the required planning framework but introduced limited analysis beyond the assignment outline.

Theoretical Understanding and Theory-Tool Integration

The Hardy Cross assignment showed the most frequent difficulties. Across the 20 individual submissions, recurring errors appeared in flow corrections, sign conventions, loop calculations, and the transfer of corrected values between iterations. The observed errors included cases in which portions of the procedure were completed but a consistent analytical solution was not maintained through convergence. Written explanations of network behavior were also limited compared with the amount of calculation presented.

The five EPANET group submissions were generally more complete. All groups constructed and ran the model, although the quality of interpretation varied. Three of the five groups clearly connected the software results to conservation principles and the earlier Hardy Cross analysis. The remaining groups reported generally acceptable outputs but provided shorter or less complete explanations of flow redistribution and pressure changes. For example, one group reported changes in pipe flow and nodal pressure after modifying the network conditions but did not explain how flow redistribution and head losses within the network produced those changes.

The five comprehensive water distribution project submissions showed developing to clear evidence of theory-tool integration. Stronger reports maintained consistency among demand assumptions, nodal inputs, network layout, modeled pressures, tested modifications, and final recommendations. Other reports completed the required modeling steps but provided less explanation of the physical basis or practical realism of the modifications.

The 20 runoff modeling submissions and 20 storm sewer design submissions were generally more successful than the Hardy Cross work. Students typically completed the structured calculations and produced the required plots or design checks. The depth of explanation varied, particularly in discussions of unit hydrograph assumptions, method limitations, downstream design trends, and the rationale for selected pipe sizes or slopes.

Engineering Interpretation and Judgment

Engineering judgment was most visible in assignments that explicitly required diagnosis, feasibility assessment, or recommendation. In the comprehensive design project, stronger groups used pressure criteria and modeled system behavior to justify modifications to the tank location or operating level. Other groups identified acceptable numerical solutions but provided less discussion of whether the proposed modifications were practically realistic.

In the runoff modeling and storm sewer design assignments, students most often addressed physical meaning, assumptions, and limitations when these elements were directly requested in the assignment. The quality of these explanations varied from brief identification of a limitation to more developed discussion of how the limitation affected the analysis.

The water crisis planning presentations showed developing evidence of engineering judgment and communication. Groups addressed the required nine-step planning process and proposed generally reasonable alternatives. Most presentations, however, remained close to the information and organizational structure supplied in the laboratory sheet, with limited additional evidence or scenario specific analysis.

DISCUSSION

Table 4 synthesizes the key findings across the three areas and their implications for engineering instruction.

Table 4
Synthesis of Key Findings and Implications for Engineering Instruction

Area	Significant finding across activities	Implication for engineering instruction
Theoretical understanding	Structured analytical activities made students' application of governing principles more visible, while successful software operation did not consistently correspond to an equally strong explanation of the underlying theory.	Computational activities should require students to connect outputs to governing principles rather than treating successful model execution as evidence of conceptual understanding.
Engineering judgment	Judgment was most evident when assignments explicitly required students to interpret results, assess feasibility, identify limitations, and justify recommendations.	Engineering judgment should be deliberately elicited through assignment prompts rather than assumed to emerge from realistic or computationally complex tasks.

Laboratory assignment design	Activities that connected analytical reasoning, computational analysis, verification, interpretation, and decision-making provided stronger opportunities for theory-tool integration.	Laboratory activities should be structured so that students move between analytical expectations, computational results, verification, and engineering decisions.
------------------------------	--	---

Theoretical Understanding in Computationally Supported Work

The first research question examined how students demonstrated theoretical understanding in theory-focused and computationally supported assignments. The findings indicate that theoretical understanding was most consistently visible when students applied governing equations through structured and sequential procedures. The runoff and storm sewer assignments divided the analytical work into clearly defined stages, and students generally completed the required calculations and produced the required plots or design checks.

The Hardy Cross activity required students to coordinate continuity, energy balances, assumed flow directions, shared pipes, and repeated corrections simultaneously. The recurrence of errors in these interconnected steps suggests that students experienced greater difficulty carrying this iterative procedure through convergence than completing the more structured analytical activities. This finding suggests that the visibility of theoretical reasoning depends not only on whether an assignment is analytical or computational, but also on the structure and complexity of the task. Complex analytical procedures may expose weaknesses in conceptual understanding that are less apparent in more structured activities.

The more complete procedural execution observed in the EPANET assignment should not be interpreted as evidence that the underlying hydraulic theory was fully understood. All five groups constructed and ran the model, but only three clearly connected the software results to conservation principles and the earlier Hardy Cross analysis. Students were more consistent in completing the software-based modeling requirements than in explaining why pressures and flows changed. This distinction is consistent with research showing that computational models can support disciplinary learning, but student explanations are needed to determine whether tool use is connected to conceptual reasoning (Feijoo-Garcia et al., 2025; Lyon & Magana, 2021; Ravan & Huang, 2024; Vieira et al., 2019). The findings also support the importance of moving among equations, network diagrams, software outputs, and physical interpretations rather than relying on one representation alone (Moore et al., 2013).

This distinction has important implications for computationally supported engineering instruction. Professional software can reduce the procedural burden associated with complex calculations and allow students to investigate systems that would otherwise be difficult to analyze. At the same time, automated solution

procedures can make intermediate reasoning less visible. Consequently, successful completion of a computational activity should not, by itself, be treated as evidence that students understand the governing relationships represented by the model. Laboratory assignments may therefore need to require students to explain selected outputs, connect modeled behavior to governing principles, or independently verify portions of a computational solution.

Informal classroom interactions provided additional context. During the Hardy Cross laboratory sessions, several students asked why the analytical method was necessary when professional software could perform the calculations. These comments were not included in the formal coding and are presented only as contextual instructor observations. The submitted artifacts did not directly assess students' attitudes toward analytical methods relative to software; however, the comments provide additional context for the need to make the relationship between theory and computational tools explicit.

The educational purpose of the Hardy Cross activity is not to suggest that engineers should replace professional software with manual calculations. Rather, analytical understanding provides a basis for recognizing questionable inputs, units, assumptions, boundary conditions, and outputs. The intentionally paired Hardy Cross and EPANET sequence therefore remains valuable, but the connection between the activities should be made more explicit. For example, students could be asked to identify which governing relationships used in the analytical exercise are represented within the software model or to verify a selected software result using an analytical calculation. Such activities would make the relationship between theory and computational practice more visible.

More broadly, the findings suggest that computational tools are most likely to support theoretical understanding when their use is deliberately connected to explanation, verification, and physical interpretation. The issue is therefore not whether analytical methods or computational tools are inherently more effective, but how assignments require students to move between them. In engineering courses, computational proficiency and conceptual understanding should be treated as complementary learning outcomes, and competence in one should not be assumed to demonstrate competence in the other.

With respect to the first research question, evidence of theoretical understanding was most visible when submissions correctly applied governing principles and connected calculations to physical behavior. This evidence was more consistent in the structured analytical activities and less consistent in the complex iterative activity and in software-based work where interpretation was limited. Completion of software tasks did not necessarily provide corresponding evidence of understanding of the principles implemented by the software. These findings support the broader argument of this study that computational tools can strengthen engineering learning when they are integrated with theoretical

reasoning, but tool use alone does not provide sufficient evidence of conceptual understanding.

Engineering Judgment

Based on the coding criteria, engineering judgment was most evident when assignments explicitly required students to diagnose a system, evaluate feasibility, and defend a recommendation. Evidence of judgment was identified through students' interpretation of physical meaning, assessment of reasonableness and limitations, diagnosis of system behavior, and justification of engineering decisions. The comprehensive water distribution project provided a particularly direct opportunity to demonstrate these forms of judgment because students had to interpret pressure criteria, modify the modeled system, and determine whether the resulting design was practical.

This finding is consistent with the view that engineering judgment involves evaluating technical work in context rather than performing calculations alone (Giuliano et al., 2022). Project reports showing clearer evidence of engineering judgment treated the model as evidence within a larger design argument. They connected assumptions and calculations to modeled performance and then used those results to support a recommendation.

Students discussed assumptions, limitations, reasonableness, and physical meaning less consistently when these elements were not directly requested. In the runoff and storm sewer assignments, students provided more complete explanations when the laboratory sheet explicitly required discussion of limitations or design trends; when the required work focused primarily on calculations or design completion, these forms of interpretation were less evident. Similarly, the planning presentations generally followed the required framework but showed limited independent extension beyond it.

These patterns suggest that realistic engineering problems and computational tools alone do not necessarily produce evidence of engineering judgment. If interpretation, feasibility assessment, and justification are intended learning outcomes, they should be explicitly incorporated into assignment prompts and assessment criteria. Computational tools may then serve not simply as mechanisms for producing results, but as sources of evidence that students must interpret and use to support engineering decisions.

With respect to the second research question, the findings suggest that engineering judgment becomes more visible in student work when assignments explicitly require interpretation, feasibility assessment, and justification. This has an important implication for engineering instruction: realistic context alone may not be sufficient to elicit these forms of reasoning. Assignment design must provide students with opportunities and explicit expectations to explain what evidence informed a decision and why a proposed alternative is reasonable.

Implications for Laboratory Assignment Design

The third research question considered how laboratory design can better support the integration of computational tools with conceptual understanding. The findings support a staged approach that links analytical reasoning, computational analysis, verification, interpretation, and decision-making. This staged structure is consistent with recent engineering education literature emphasizing authentic problem contexts, explicit scaffolding, iterative knowledge application, and opportunities for students to explain or justify decisions within project and challenge-based learning environments (Doulougeri et al., 2024; Lavado-Anguera et al., 2024).

The sequence from Hardy Cross calculations to EPANET modeling was useful because it introduced the principles automated by the software. However, the relationship should also be reinforced after the software is used. Laboratory assignments should require students to verify at least one selected computational result using a governing principle, energy balance, continuity relationship, or simplified analytical calculation and explain any differences between the analytical and computational results.

The Hardy Cross assignment could also be shortened to reduce the amount of repetitive calculation while increasing the emphasis on the meaning of each correction. A follow-up EPANET activity could include a deliberate modeling error, such as an incorrect elevation, demand, pipe status, or unit. Students could then use theory and a simplified calculation to diagnose the problem. This would make the professional value of theoretical understanding more visible.

Assignment deliverables should also distinguish tool operation from engineering understanding. Screenshots and completed tables can document that a procedure was performed, but they provide limited evidence of reasoning. Brief prompts can require students to predict system behavior, identify the governing principle behind an observed result, evaluate whether an output is reasonable, state an assumption or limitation, or justify a design modification.

Because the group submissions represented collective rather than individual work, the present analysis cannot determine how participation in group activities affected each student's theoretical understanding or engineering judgment. Group artifacts may reflect collaborative reasoning while also masking differences in individual understanding or contribution. Including a small individual component, such as a verification calculation, reflection, or oral explanation, would provide additional evidence of each student's reasoning while preserving the benefits of collaborative design work.

These implications extend beyond the specific hydraulic and water-resources activities examined here. As computational tools become increasingly common across engineering courses, laboratory assignments should be designed so that

software use remains connected to the underlying principles, interpretation, and decision-making expected in professional practice. Although the present study was conducted in water resources and hydraulic engineering courses, the broader instructional challenge is not discipline specific. Many engineering and STEM courses rely on computational tools, simulation environments, spreadsheets, and specialized software that can produce technically correct outputs without necessarily revealing students' underlying conceptual understanding. The findings therefore contribute to higher education research by showing how assignment design can make theoretical reasoning and engineering judgment more visible through explicit requirements for explanation, verification, interpretation, and justification. These principles may be transferable to other disciplines in which computational tools are used to support analysis and decision-making. Future research could examine whether similar assignment structures produce comparable patterns across other engineering disciplines and computational environments.

The findings addressing the third research question suggest that laboratory assignment design can better support theory-tool integration by requiring students to move visibly between governing principles and computational results. Computational tools were most valuable when students used them to test, interpret, and refine engineering reasoning rather than treating the software output as the final answer.

Overall, balanced engineering learning was most evident when students connected assumptions and governing principles to computational results and then used both to support a defensible engineering decision.

Study Limitations

The qualitative coding was conducted by a single researcher, which introduces the possibility of researcher subjectivity in the interpretation and classification of student work. Although the analysis used predefined coding dimensions and descriptive criteria that were applied consistently across the de-identified submissions, and an intra-rater consistency check was conducted on a subset of the submissions, no independent second-coder analysis was performed. The findings should therefore be interpreted as descriptive patterns within the selected course artifacts rather than as independently validated qualitative classifications. Future studies could strengthen the credibility and confirmability of the analysis through multiple coders, peer debriefing, or formal assessment of inter-rater agreement.

The unit of analysis also varied across assignments. Some artifacts represented individual student work, whereas others represented group reports or presentations. Group submissions may reflect peer collaboration, division of responsibilities, or stronger contributions from particular group members and therefore may mask variation in individual conceptual understanding. The

evidence levels across individual and group assignments should consequently be interpreted as descriptive patterns within each assignment context rather than as directly comparable measures of individual student performance.

Other instructional and contextual factors were not independently evaluated in this study. Differences in instructor guidance, clarity and structure of assignment directions, students' prior knowledge and experience with computational tools, and access to or functionality of personal computing equipment and software may have influenced the evidence observed in the submitted work. These factors make it difficult to attribute the observed patterns solely to assignment design or computational tool use. In addition, the analysis was conducted retrospectively using completed course artifacts. Consequently, the study was limited to evidence contained in the submitted work and could not directly probe students' reasoning through follow-up interviews, think-aloud activities, or additional questions about the decisions reflected in their submissions. Future research could incorporate these approaches and examine similar instructional designs across different instructors, courses, and engineering disciplines.

CONCLUSIONS

This study examined how undergraduate engineering students demonstrated theoretical understanding, theory-tool integration, and engineering judgment across laboratory assignments with different levels of computational support. The findings indicate that computational tools were most educationally valuable when students were required to connect model outputs with governing principles, physical interpretation, verification, and engineering decisions. Successful software operation alone did not necessarily demonstrate strong conceptual understanding, while engineering judgment was most evident when assignments explicitly required students to assess feasibility, interpret results, and justify decisions. Overall, the findings support laboratory designs that deliberately integrate analytical reasoning, computational tools, and engineering judgment rather than treating these elements as separate learning activities.

These findings have broader relevance for higher education as computational and emerging technologies become increasingly embedded in undergraduate instruction. Although the study focused on water resources and hydraulic engineering, the underlying instructional principles may also apply to other engineering and STEM disciplines in which students use computational tools to analyze, model, or design complex systems. The findings may therefore inform instructional design across varied institutional and international contexts where educators seek to balance technological proficiency with conceptual understanding and professional judgment.

AI TOOLS STATEMENT

Generative AI (ChatGPT, OpenAI) was used during manuscript preparation solely to assist with language refinement, clarity, organization, and formatting. It was not used to generate research data, independently code or analyze student work, or generate findings or conclusions. The author reviewed, edited, and validated all AI-assisted content and retains full responsibility for the accuracy, originality, interpretations, citations, and conclusions of the manuscript.

REFERENCES

- Álvarez Ariza, J., & Hernández Hernández, C. (2026). Investigating the impact of Education 4.0 and digital learning on students' learning outcomes in engineering: A four-year multiple-case study. *Informatics*, 13(2), Article 18. <https://doi.org/10.3390/informatics13020018>
- Bezzina, A., & Zammit, J. P. (2026). A comprehensive framework for designing metaverse-based learning for engineering education. *Multimodal Technologies and Interaction*, 10(7), Article 74. <https://doi.org/10.3390/mti10070074>
- Doulougeri, K., Vermunt, J. D., Bombaerts, G., & Bots, M. (2024). Challenge-based learning implementation in engineering education: A systematic literature review. *Journal of Engineering Education*, 113(4), 1076–1106. <https://doi.org/10.1002/jee.20588>
- Feijoo-Garcia, M. A., Zhang, Y., Gu, Y., Magana, A. J., Benes, B., & Popescu, V. (2025). Students' conceptual explanations of neural networks enabled by extended reality learning: A multiple methods approach. *Computer Applications in Engineering Education*, 33(6), e70084. <https://doi.org/10.1002/cae.70084>
- Feisel, L. D., & Peterson, G. D. (2002). A colloquy on learning objectives for engineering education laboratories. *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*, Montreal, Canada, 7.20.1–7.20.12. <https://doi.org/10.18260/1-2--11246>
- Feisel, L. D., & Rosa, A. J. (2005). The role of the laboratory in undergraduate engineering education. *Journal of Engineering Education*, 94(1), 121–130. <https://doi.org/10.1002/j.2168-9830.2005.tb00833.x>
- Giuliano, H. G., Giri, L. A., Nicchi, F. G., Weyerstall, W. M., Ferreira Aicardi, L. F., Parselis, M., & Vasen, F. (2022). Critical thinking and judgment on engineer's work: Its integration in engineering education. *Engineering Studies*, 14(1), 6–16. <https://doi.org/10.1080/19378629.2022.2042003>
- Glancy, A. W., & Moore, T. J. (2013). *Theoretical foundations for effective STEM learning environments* (School of Engineering Education Working Papers, Paper 1). Purdue University. <https://docs.lib.purdue.edu/enewp/1/>

- Hidayat, H., Zuhlendra, Z., Efrizon, E., Delianti, V. I., Dewi, F. K., Mohd Isa, M. R., Harmanto, D., & Tanucan, J. C. M. (2024). Computational thinking skills in engineering education: Enhancing academic achievement through innovations, challenges, and opportunities. *TEM Journal*, 13(4), 3454–3467. <https://doi.org/10.18421/TEM134-78>
- Lampropoulos, G., Fernández-Arias, P., de Bosque, A., & Vergara, D. (2025). Virtual reality in engineering education: A scoping review. *Education Sciences*, 15(8), Article 1027. <https://doi.org/10.3390/educsci15081027>
- Lavado-Anguera, S., Velasco-Quintana, P.-J., & Terrón-López, M.-J. (2024). Project-based learning (PBL) as an experiential pedagogical methodology in engineering education: A review of the literature. *Education Sciences*, 14(6), Article 617. <https://doi.org/10.3390/educsci14060617>
- Li, R., Lund, A., & Nordsteien, A. (2023). The link between flipped and active learning: A scoping review. *Teaching in Higher Education*, 28(8), 1993–2027. <https://doi.org/10.1080/13562517.2021.1943655>
- Litzinger, T. A., Lattuca, L. R., Hadgraft, R. G., Newstetter, W. C., Alley, M., Atman, C., DiBiasio, D., Finelli, C., Diefes-Dux, H., Kolmos, A., Riley, D., Sheppard, S., Weimer, M., & Yasuhara, K. (2011). Engineering education and the development of expertise. *Journal of Engineering Education*, 100(1), 123–150. <https://doi.org/10.1002/j.2168-9830.2011.tb00006.x>
- Luo, J., Zheng, C., Yin, J., & Teo, H. H. (2025). Design and assessment of AI-based learning tools in higher education: A systematic review. *International Journal of Educational Technology in Higher Education*, 22, 42. <https://doi.org/10.1186/s41239-025-00540-2>
- Lyon, J. A., & Magana, A. J. (2021). The use of engineering model-building activities to elicit computational thinking: A design-based research study. *Journal of Engineering Education*, 110(1), 184–206. <https://doi.org/10.1002/jee.20372>
- Magana, A. J., & Silva Coutinho, G. (2017). Modeling and simulation practices for a computational thinking-enabled engineering workforce. *Computer Applications in Engineering Education*, 25(1), 62–78. <https://doi.org/10.1002/cae.21779>
- Moore, T. J., Miller, R. L., Lesh, R. A., Stohlmann, M. S., & Kim, Y. R. (2013). Modeling in engineering: The role of representational fluency in students' conceptual understanding. *Journal of Engineering Education*, 102(1), 141–178. <https://doi.org/10.1002/jee.20004>
- Pande, P., & Jepsen, P. M. (2025). Science lab safety goes immersive: An ecological media-comparison study with gender analyses assessing iVR's learning effectiveness. *Research and Practice in Technology Enhanced Learning*, 20, Article 1. <https://doi.org/10.58459/rptel.2025.20001>

- Ravan, K., & Huang, S. (2024). Assessing computational thinking in engineering education: A systematic review. *Proceedings of the Canadian Engineering Education Association*. <https://doi.org/10.24908/pceea.2024.18632>
- Sheppard, S. D., Macatangay, K., Colby, A., & Sullivan, W. M. (2009). *Educating engineers: Designing for the future of the field*. Jossey-Bass.
- Tenney, K., Stringer, B. P., LaTona-Tequida, T., & White, I. (2023). Conceptualizations and limitations of STEM literacy across learning theories. *Journal of Microbiology & Biology Education*, 24(1), e00168-22. <https://doi.org/10.1128/jmbe.00168-22>
- Vieira, C., Magana, A. J., Roy, A., & Falk, M. L. (2019). Student explanations in the context of computational science and engineering education. *Cognition and Instruction*, 37(2), 201–231. <https://doi.org/10.1080/07370008.2018.1539738>
- Zhang, J. (2026). Scaffolding probabilistic reasoning in civil engineering education: Integrating AI tutoring with simulation-based learning. *Education Sciences*, 16(1), Article 103. <https://doi.org/10.3390/educsci16010103>
- Zhou, Z., Oveissi, F., & Langrish, T. (2024). Applications of augmented reality (AR) in chemical engineering education: Virtual laboratory work demonstration to digital twin development. *Computers & Chemical Engineering*, 188, Article 108784. <https://doi.org/10.1016/j.compchemeng.2024.108784>

Bio

THATHSARANI D.H. HERATH MUDIYANSELAGE, PhD, is an Assistant Teaching Professor in the Department of Civil Engineering at Kansas State University. Her academic interests are broadly in the field of water resources engineering and engineering education, with a particular interest in the effective use of computational tools in undergraduate instruction. Her research background includes computational fluid dynamics, coastal hydrodynamics, and water resources engineering. Email: dilinih@ksu.edu