Jurnal Theorems (*The Original Reasearch Of Mathematics*)
Volume 11, Nomor 1, Januari 2026
URL: http://ejournal.unma.ac.id/index.php/th

e-ISSN: 2528-102X
p-ISSN: 2541-4321

Analysis of Students' Computational Thinking in Solving Numerical Methods Problems Using Microsoft Excel

Resty Yuliana[1], Melinda Yanuar[2], Nur Hidayanti[3], Huswatun Hasanah[2]
[1]UNIBA, Department of Statistics, Indonesia
[2]UNIBA, Department of Mathematics Education, Indonesia
[3]UNIBA, Department of Information Technology Education, Indonesia

***Corresponding Author*[*]:**
Resty Yuliana
Universitas Bina Bangsa, Serang, Indonesia
resty.yuliana@binabangsa.ac.id

Article Info:
Received 16 Dec 2025
Revised 30 Dec 2025
Accepted 20 Jan 2026

How to Cite:
Yuliana, R., Yanuar, M., Hidayanti, N., & Hasanah, H. (2026). Analysis of students' computational thinking in solving numerical methods problem using microsoft excel. Jurnal Theorems (The Original Research of Mathematics, 11(1), 32-49.

ABSTRACT

Computational thinking is an essential skill in the digital age, particularly in Numerical Methods courses that require logical, algorithmic, and technology-based problem solving. However, students often find it difficult to acquire algorithmic thinking and abstraction while using computational tools to apply numerical techniques. This study aims to assess the computational thinking abilities of Universitas Banten Jaya students by examining their algorithmic and abstraction indicators during Numerical Methods exercises in Microsoft Excel. This study, which employed a descriptive quantitative–qualitative methodology, involved 62 students who were enrolled in the Numerical Methods course during the Odd Semester of the 2025–2026 academic year. Data for the midterm exam was collected via students' responses to computational essays that involved the development of algorithms, their application in Microsoft Excel, and the analysis of their results. Qualitative data was evaluated using content analysis, and quantitative data was evaluated using descriptive statistics.The results showed a medium level of computational thinking skill with an average score of 66.29. The achievement scores for the algorithmic and abstraction measures are 67.34% and 67.74%, respectively. These findings suggest that more structured and digitally integrated training is needed to improve students' computational thinking skills.

Keywords: *Computational thinking; Numerical methods; Algorithmic thinking; Abstraction; Microsoft Excel*

INTRODUCTION

The rapid development of digital technology and the increasing demand for 21st-century skills require university graduates not only to master theoretical knowledge but also to possess strong computational problem-solving abilities. Computational thinking (CT) is crucial competency encompassing cognitive skills such as decomposition, abstraction, pattern recognition, algorithmic thinking, and debugging, which facilitate systematic and efficient problem-solving (Lye & Koh, 2014; Shute et al., 2017; Wing, 2006).

Computational thinking has been conceptualized as a problem-solving framework that transcends programming and includes logical reasoning, algorithm design, abstraction, and systematic evaluation applicable across disciplines (Brennan & Resnick, 2012; Denning, 2017;

Selby & Woollard, 2013). In mathematics and STEM education, computational thinking is acknowledged as a fundamental talent that underpins higher-order thinking, mathematical modeling, and problem formulation (Barendsen et al., 2015; Weintrop et al., 2016).

Recent studies have shown that integrating computational thinking into mathematics instruction can enhance students' conceptual understanding and their ability to translate mathematical ideas into executable procedures (Penelitian et al., 2024). However, effective integration requires explicit instructional strategies that align mathematical content with computational thinking components, particularly in courses that inherently involve procedural, iterative, and approximation-based reasoning. One such course is Numerical Methods, which occupies a central position in engineering and applied mathematics curricula.

In Numerical Methods courses, students must develop numerical procedures, formulate iterative algorithms, conduct approximation and error analysis, and analyze convergence behavior. These learning demands are closely related to algorithmic thinking and abstraction, which are fundamental elements of computational thinking (Grover & Pea, 2013). Algorithmic reasoning is central to numerical problem solving because students must translate mathematical formulations into structured computational steps involving iteration and logical decision-making (Lockwood et al., 2016; Mirolo et al., 2021).

General-purpose software, such as Microsoft Excel, is often utilized to facilitate learning because of its accessibility, integrated numerical and logical functions (e.g., IF functions, iterative cell references, scientific notation), and its ability to illustrate iterative processes and numerical convergence. Empirical research indicates that Excel serves as a cognitive tool that aids students in converting mathematical procedures into sequential computing algorithms(Baker & Sugden, 2003; Taufik & Susanti, 2024). Utilizing spreadsheets allows students to openly examine the iterative functioning of numerical algorithms, so potentially enhancing their computational thinking skills.

Despite the growing body of research on computational thinking and the use of digital tools in mathematics education, several methodological challenges and research gaps remain. First, most existing studies focus on computational thinking in general programming contexts or at the K–12 education level, while relatively few investigations address computational thinking in domain-specific higher education courses, such as Numerical Methods, where algorithm design and abstraction are essential learning outcomes. Second, many studies emphasize instructional interventions but do not provide detailed empirical mapping of students' computational thinking abilities, particularly with respect to algorithmic and abstraction indicators during authentic problem-solving tasks. Furthermore, the assessment of computational thinking in higher education remains limited, as many existing instruments are overly broad, programming-oriented, or insufficiently contextualized to specific mathematical tasks (Basu et al., 2014; Guggemos et al., 2024; Korkmaz et al., 2017; Román-González et al., 2017). Consequently, there is a need for domain-specific assessments that can capture students' computational reasoning processes during authentic numerical problem-solving activities.

In addition, the actual learning conditions of students in Numerical Methods courses are rarely discussed in the literature. Preliminary observations in Numerical Methods classes at Universitas Banten Jaya (UNBAJA) indicate that students demonstrate varying levels of computational competence. While some students can follow procedural steps mechanically, many encounter difficulties in (1) independently constructing algorithms, (2) translating mathematical formulas into spreadsheet-based logic, and (3) verifying the correctness and convergence of numerical solutions. These conditions suggest that students' computational thinking abilities, particularly in algorithmic reasoning and abstraction, require systematic analysis and deeper understanding.

Rectifying these deficiencies is essential for both theoretical and practical perspectives. This study theoretically enhances the existing research on computational thinking by offering empirical proof of the manifestation of algorithmic thinking and abstraction within a specific area, specifically Numerical Methods utilizing Microsoft Excel. This study further substantiates the notion that computational thinking ought to be analyzed not merely as a generic cognitive ability but also as a contextual competence that arises through engagement with mathematical tasks and computational instruments.

From a practical perspective, the findings of this study are expected to offer valuable insights for Numerical Methods lecturers, particularly in the design of learning activities, the selection of appropriate instructional tools, and the development of assessment instruments that explicitly target computational thinking skills. In addition, the results may inform the development of evaluation frameworks and performance-based tasks aimed at assessing students' abilities to construct algorithms and perform abstraction in spreadsheet-based numerical problem solving.

Based on this background, the purpose of this study is to analyze the computational thinking abilities of students at Universitas Banten Jaya (UNBAJA) in solving Numerical Methods problems using Microsoft Excel, with a specific focus on two computational thinking indicators: algorithmic thinking and abstraction. By examining students' work and problem-solving processes, this study seeks to provide a clearer understanding of how computational thinking operates in higher-education numerical mathematics contexts.

METHODS

This study employed a qualitative descriptive approach supported by descriptive quantitative analysis to obtain a comprehensive depiction of students' computational thinking abilities when solving Numerical Methods problems using Microsoft Excel. The qualitative descriptive approach was selected to elucidate phenomena as they manifest in instructional contexts by detailing student actions, problem-solving strategies, and the emergence of specific cognitive behaviours, without the inclusion of experimental intervention (Villamin et al., 2025). The study was performed at Universitas Banten Jaya (UNBAJA). This study did not aim to test causal relationships or the effectiveness of specific instructional interventions, but rather to describe and analysed students' computational thinking profiles as they naturally emerged during numerical problem-solving activities using Microsoft Excel.

The research participants comprised 62 undergraduate students enrolled in the Numerical Methods course during the odd semester of the 2025/2026 academic year, all of whom had successfully completed the midterm assessment (UTS). A total sampling technique was applied, whereby all students who met the inclusion criteria were selected as research participants. The inclusion criteria were that students were officially registered in the course and submitted complete responses to all midterm examination questions.

The primary data source of this study was students' responses to the Numerical Methods midterm examination. The midterm exam consisted of open-ended problem-solving exercises that required students to use Microsoft Excel to apply numerical methods. These tasks were designed to assess students' computational thinking in a domain-specific context by requiring algorithm construction, abstraction, and interpretation of results. The use of open-ended, performance-based tasks aligns with recommendations for valid computational thinking assessment, as such tasks allow for the evaluation of students' reasoning processes rather than final answers alone (Brennan & Resnick, 2012; El-Hamamsy & others, 2025). Each of the five exam questions required students to create numerical algorithms, use spreadsheet formulas for iterative calculations, convert mathematical expressions into computational representations, and interpret the numerical findings. Instead, then only testing procedural knowledge, these challenges were purposefully created to evaluate students' computational thinking skills in a domain-specific context. Two computational thinking indicators: algorithmic thinking, which is measured by Questions 1, 3, and 5, and abstraction, which is measured by Questions 2 and 4, were used to construct the exam questions.

The test instrument was subjected to an expert assessment process that included two mathematics education lecturers and one lecturer in numerical techniques to guarantee content **validity**. The experts assessed how well the learning objectives of the Numerical Methods course,

Jurnal Theorems (*The Original Reasearch Of Mathematics*)
Volume 11, Nomor 1, Januari 2026
URL: http://ejournal.unma.ac.id/index.php/th

e-ISSN: 2528-102X
p-ISSN: 2541-4321

the exam items, and the computational thinking indicators aligned. Revisions were made based on their input to enhance the questions' connection to computational thinking components, clarity, and cognitive demand.

Data collection was carried out during the official midterm examination schedule of the Numerical Methods course. The examination was administered in a computer laboratory over a duration of 120 minutes. Students worked individually under supervised conditions and were instructed to solve the numerical problems given by using Microsoft Excel only, without access to other software or internet resources. At the end of the session, students submitted their Excel files containing formulas, iterative procedures, and numerical results, as well as brief written explanations of their solution steps where required.

Data analysis was conducted in two complementary stages, namely quantitative descriptive analysis and qualitative descriptive analysis. To have a general idea of the students' computational thinking skills, the mean and standard deviation of their midterm exam scores were computed during the quantitative stage. Based on the score categorization criteria put forward by (Nainggolan et al., 2024), students' computational thinking levels were divided into three categories: high, medium, and low. This was done to make interpretation easier. As shown in Table 1, this categorization considers each student's score in relation to the class mean and standard deviation.

Table 1. Classification of Students' Scores

Score	Interpretation
$n \geq \bar{x} + s$	High ability
$\bar{x} - s < n < \bar{x} + s$	Medium
$n \leq \bar{x} - s$	Low ability

Notes:

n : student's scores

$\bar{x}$: the mean student's scores

s : the standard deviation of students' test scores

Based on this classification, the quantitative results were used to identify the distribution of students across different levels of computational thinking ability. These results subsequently guided the qualitative analysis, in which students' responses from each category were examined to obtain a more in-depth understanding of their computational thinking processes. By linking statistical classification with qualitative examination, this study was able to explore not only the overall level of students' computational thinking abilities but also the qualitative characteristics associated with high, medium, and low computational performance.

e-ISSN: 2528-102X
p-ISSN: 2541-4321

In addition to the classification analysis, a percentage-based analysis was conducted to determine the level of achievement for each computational thinking indicator. The percentage of achievement was calculated using the following formula:

$$P = \frac{F}{A} \times 100\%$$

Notes:

P : the percentage of scores for computational thinking ability indicators

F : the total score of the indicator obtained

A : the maximum possible indicator score

The resulting percentages were then interpreted using predefined criteria to describe students' achievement levels for each indicator.

Table 2. Percentage of Indicator Scores

Category	Score
Very high	$80\% \leq \%NI$
High	$60\% \leq \%NI < 80\%$
Medium	$40\% \leq \%NI < 60\%$
Low	$20\% \leq \%NI < 40\%$
Very low	$\%NI < 20\%$

The percentage results were interpreted according to the criteria proposed by (Permatasari, 2024) and were used to describe students' levels of achievement in the two computational thinking indicators examined in this study, namely **algorithmic thinking** and **abstraction**.

FINDINGS

This section presents the findings of students' computational thinking abilities based on descriptive statistical analysis of the Numerical Methods midterm examination (UTS) scores and qualitative analysis of students' responses to the test items. The results describe students' overall levels of computational thinking ability as well as their performance across the algorithmic thinking and abstraction indicators.

Students' Computational Thinking Ability Based on Descriptive Statistics

Students' Numerical Methods midterm examination scores were analyzed using descriptive statistics to describe their overall computational thinking performance. The results are summarized in Table 3.

Table 3. *Students' Computational Thinking Abilities*

Number of Students	Maximum Score	Minimum Score	Mean	Standard Deviation
62	100	5	66,29	25,59

The results indicate substantial variation in students' computational thinking abilities. Similar variability has been reported in previous studies, suggesting that differences in exposure to structured algorithmic tasks and digital tools significantly influence students' computational problem-solving performance (Bocconi et al., 2016; Guggemos et al., 2024). The highest score was 100, while the lowest score was 5, reflecting considerable differences in students' mastery of numerical methods and computational skills. The mean score of 66.29 suggests that, on average, students' computational thinking abilities fall within the medium category.

Although some students demonstrated strong computational thinking abilities, others experienced significant difficulties, as indicated by the relatively high standard deviation (25.59), which reflects wide variability in individual performance. This uneven distribution highlights the need for instructional strategies that provide greater support for students with lower computational thinking skills, particularly in applying numerical procedures using Microsoft Excel.

Based on the classification criteria adapted from (Nainggolan et al., 2024), students' computational thinking abilities were categorized into high, medium, and low levels. The distribution of students across these categories is presented in Table 4.

Table 4. Students' Computational Thinking Abilities by Category

Category	Score Criteria	Number of Students	Percentage
High ability	$x \geq 91{,}88$	9	14,51%
Medium	$40{,}69 < x < 91{,}88$	48	77,42%
Low ability	$x \leq 40{,}69$	9	14,51%

The results show that most students (77.42%) were classified in the medium ability category, while equal proportions of students (14.51%) were classified as having high and low computational thinking abilities.

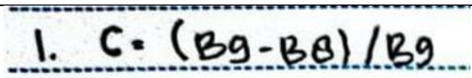
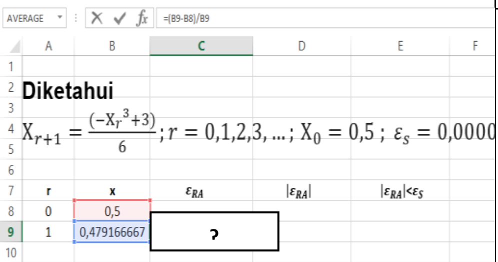
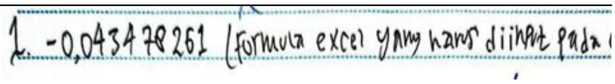
Analysis of Students' Responses on Algorithmic Thinking Indicator

To obtain deeper insights into students' computational thinking skills, a qualitative analysis was conducted based on computational thinking indicators. The first indicator examined was algorithmic thinking, which was assessed through Questions 1, 3, and 5.

e-ISSN: 2528-102X
p-ISSN: 2541-4321

Analysis of students' responses to Question 1 revealed that students in the high and medium ability categories were generally able to construct correct and systematic Excel formulas. whereas students in the low ability category frequently provided incomplete or incorrect formulas. A summary of students' responses is presented in Table 5.

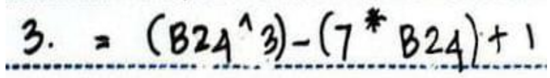
Table 5. Analysis of Students' Responses by Category for Question 1

Question 1	**Students' Responses in the High Ability Category**
Given the formula $\varepsilon_{RA} = \frac{a_{r+1}-a_r}{a_{r+1}}$, the Excel formula that should be entered in **cell C9** (ε_{RA}) in the following error analysis table is …	
	Students' Responses in the Medium ability Category
	
	Students' Responses in the Low Ability Category
	

In Question 1, students were required to construct systematic steps to derive the appropriate Excel formula for determining the error value based on the given error formula. ($\varepsilon_{RA} = \frac{a_{r+1}-a_r}{a_{r+1}}$). Students in the high and medium ability categories were able to correctly formulate the Excel expression for relative approximate error, written as $(B9 - B8)/B9$. In contrast, students in the low ability category tended to provide incorrect formulas or isolated numerical values that did not represent the intended computational procedure.

Further analysis through Question 3 (Table 6) showed that students with high computational thinking ability were able to correctly formulate the Excel expression required to compute the function value. Students in the medium category produced partially correct formulas with missing components, while students in the low category demonstrated difficulties in understanding the required computational procedures.

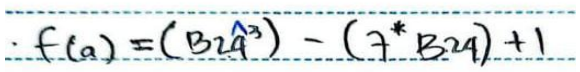
Table 6. Analysis of Students' Responses by Ability Category on Question 3

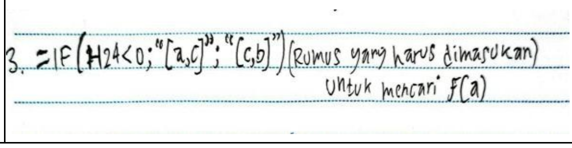
Question 3	**Students' Responses in the High Ability Category**
f the root of the function is to be determined using the bisection method on the interval [1, 2]in the following cells, then the formula	
	Students' Responses in the Medium ability Category

e-ISSN: 2528-102X
p-ISSN: 2541-4321

<table>
<tr><td>that should be entered to compute $f(a)$is …

	A	B	C	D	E
23	r	a	b	c	$f(a)$
24	0	1	3	2	

</td><td>

Students' Responses in the Low Ability Category

</td></tr>
</table>

In Question 3, students were required to construct systematic steps to determine the Excel formula that needed to be entered to compute the value of $f(a)$. Students with high computational thinking ability were able to write the formula correctly, namely $= (B24^{\wedge}3) - (7^{*}B24) + 1$, whereas students in the medium category were able to formulate the expression but with incomplete components. In contrast, students in the low ability category frequently produced incorrect formulas or isolated numerical values that did not reflect the intended computational procedure.

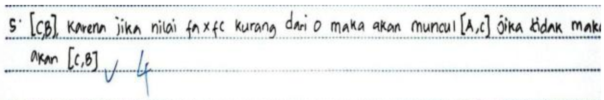
Question 5 examined students' ability to apply conditional logic using an IF function in Excel. As shown in Table 7, students in the high and medium ability categories were generally able to identify the correct output and explain the systematic steps involved. In contrast, students in the low ability category showed limited understanding of conditional operations, resulting in incorrect responses.

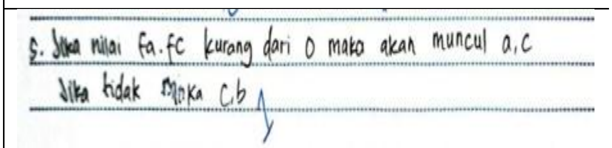
Table 7. Analysis of Students' Responses by Ability Category on Question 5

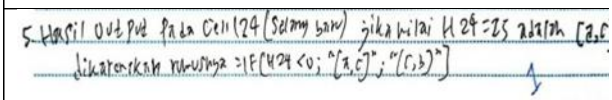
<table>
<tr><th>Question 5</th><th>Students' Responses in the High Ability Category</th></tr>
<tr><td>If the following bisection method calculation table is given:

I24 fx =IF(H24<0;"[a,c]";"[c,b]")

	A	B	C	D	E	F	G	H	I
23	r	a	b	c	f(a)	f(b)	f(c)	f(a).f(c)	Selang Baru
24	0	1	3	2	-5	7	-5	25	

What is the output displayed **in cell I24** (New Interval) if the value in cell **H24 is 25**? Explain the systematic steps used to determine the result.</td><td>

Students' Responses in the Medium ability Category


Students' Responses in the Low Ability Category

</td></tr>
</table>

Jurnal Theorems (*The Original Reasearch Of Mathematics*)
Volume 11, Nomor 1, Januari 2026
URL: http://ejournal.unma.ac.id/index.php/th

e-ISSN: 2528-102X
p-ISSN: 2541-4321

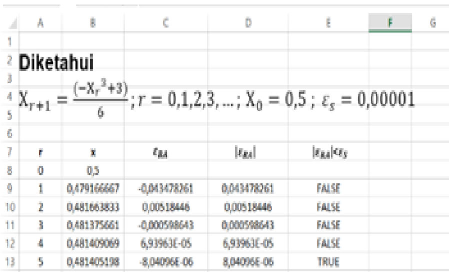
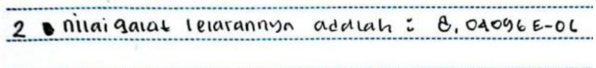
In Question 5, students were required to apply systematic reasoning to determine the output of an IF formula based on the Excel expression entered, namely =IF (H24<0,"[a, c]","[c, b]"). Students in the high and medium computational thinking ability categories were able to correctly identify the possible output values and provide accurate explanations of the underlying reasoning. In contrast, students in the low ability category demonstrated limited understanding of the conditional logic embedded in the IF formula, resulting in incorrect outputs and explanations.

Overall, the findings indicate clear differences in students' algorithmic thinking abilities across categories. Students in the high ability category consistently demonstrated systematic reasoning and accurate procedural implementation, whereas students in the low ability category struggled to construct and apply complete computational procedures.

Analysis of Abstraction Thinking Indicator

The abstraction indicator was examined via Questions 2 and 4. In Question 2, students were tasked with identifying pertinent information to ascertain the iterative error value from an Excel output table. The research (Table 8) demonstrated that students from all ability categories successfully identified the required value, suggesting that this task necessitated relatively fundamental abstraction skills.

Table 8. Analysis of Students' Responses by Ability Category on Question 2

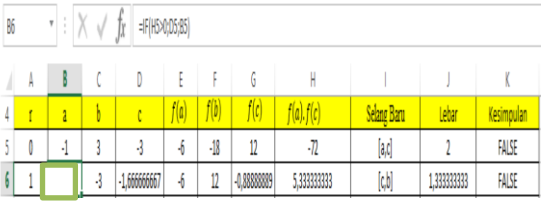
<table>
<tr><th>Question 2</th><th>Students' Responses in the High Ability Category</th></tr>
<tr><td rowspan="5">Based on the following error analysis table, the value of the iterative error is …

Diketahui
$X_{r+1} = \frac{(-X_r^3+3)}{6}$; $r = 0,1,2,3,\ldots$; $X_0 = 0{,}5$; $\varepsilon_s = 0{,}00001$
<table>
<tr><th>r</th><th>x</th><th>ε_{RA}</th><th>$|\varepsilon_{RA}|$</th><th>$|\varepsilon_{RA}|<\varepsilon_s$</th></tr>
<tr><td>0</td><td>0,5</td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>0,479166667</td><td>-0,043478261</td><td>0,043478261</td><td>FALSE</td></tr>
<tr><td>2</td><td>0,481663833</td><td>0,00518446</td><td>0,00518446</td><td>FALSE</td></tr>
<tr><td>3</td><td>0,481375661</td><td>-0,000598643</td><td>0,000598643</td><td>FALSE</td></tr>
<tr><td>4</td><td>0,481409069</td><td>6,93963E-05</td><td>6,93963E-05</td><td>FALSE</td></tr>
<tr><td>5</td><td>0,481405198</td><td>-8,04096E-06</td><td>8,04096E-06</td><td>TRUE</td></tr>
</table></td><td></td></tr>
<tr><td>Students' Responses in the Medium ability Category</td></tr>
<tr><td></td></tr>
<tr><td>Students' Responses in the Low Ability Category</td></tr>
<tr><td></td></tr>
</table>

In Question 2, students were required to identify relevant data to determine the iterative error value using Microsoft Excel. Students across all computational thinking ability categories—high, medium, and low—were able to correctly identify the location of the iterative error value, which was found in cell D13 (8.04096E−06).

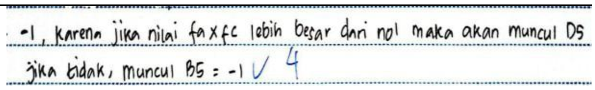
e-ISSN: 2528-102X
p-ISSN: 2541-4321

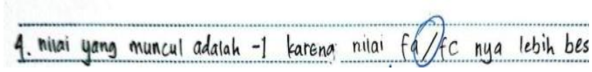
Conversely, Question 4 necessitated that students generalize the output of an IF formula utilized in the False Position Method technique. Table 9 illustrates that students in the high ability category successfully generalized the output and offered suitable rationale. Students in the middle category provided accurate responses with less precise explanations, whereas students in the low category struggled to pick pertinent material or generalize the computational pattern.

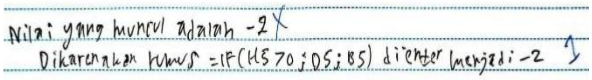
Table 9. Analysis of Students' Responses by Ability Category on Question 4

<table>
<tr><th>Question 4</th><th>Students' Responses in the High Ability Category</th></tr>
<tr><td>Given the following table of computations using the False Position Method method:


	A	B	C	D	E	F	G	H	I	J	K
4	r	a	b	c	f(a)	f(b)	f(c)	f(a).f(c)	Selang Baru	Lebar	Kesimpulan
5	0	-1	3	-3	-6	-18	12	-72	[a,c]	2	FALSE
6	1		-3	-1,666666667	-6	12	-0,88888889	5,333333333	[c,b]	1,333333333	FALSE

If the formula =IF(H5>0, D5, B5) is entered into cell B6 (value of a at iteration 1), what value will appear after pressing Enter?
Explain your answer.</td>
<td>

Students' Responses in the Medium ability Category


Students' Responses in the Low Ability Category

</td></tr>
</table>

In Question 4, students were required to generalize the Excel output generated by an IF formula and identify relevant information from the computation table. Students in the high ability category demonstrated accurate generalization supported by appropriate reasoning. Students in the medium category produced correct but less detailed explanations, whereas students in the low category were unable to generalize the output or select relevant information, leading to incorrect answers.

Summary of Computational Thinking Indicators

A summary of students' computational thinking performance across the algorithmic thinking and abstraction indicators is presented in Table 10. Based on Table 10, students' computational thinking skills across both indicators exhibited varying levels of achievement. For the algorithmic thinking indicator, the highest achievement was observed in Question 1 (68.14%), while the lowest was in Question 3 (58.87%), indicating greater difficulty in organizing more complex computational procedures. Overall, the algorithmic thinking indicator achieved a score of 67.34%, which falls within the **medium category**.

e-ISSN: 2528-102X
p-ISSN: 2541-4321

Table 10. Students' Computational Thinking Skills by Indicator

Question	Algorithmic Thinking Indicator	Question	Abstraction Thinking Indicator
Question 1	68,14%	Question 2	80,24%
Question 3	58,87%	Question 4	55,24%
Question 5	67,34%		
Overall Percentage	**64,78%**	**Overall Percentage**	**67,74%**

For the abstraction indicator, the highest achievement occurred in Question 2 (80.24%), reflecting students' ability to identify relevant information and determine the core of the problem. Conversely, the lowest achievement was recorded in Question 4 (55.24%), indicating challenges in generalization and pattern recognition when analysing iterative data in the False Position Method. Overall, the abstraction indicator attained a score of 67.74%, also classified as medium.

These findings suggest that students' computational thinking skills, as measured by both indicators, are predominantly at a medium level, which will be further analysed in the next section in relation to established computational thinking frameworks and previous research.

DISCUSSION

This study's findings offer significant insights into students' computational thinking skills regarding Excel-based numerical approaches. This part analyses the outcomes by evaluating students' performance in algorithmic thinking and abstraction indicators, correlating them with recognized computational thinking frameworks and prior research. The findings demonstrate that students' computational thinking skills varied not just within indicators but also across varying degrees of proficiency, highlighting disparities in their conceptualization, structuring, and generalization of computational processes.

Algorithmic Thinking

The results illustrate that students who have mastered the algorithmic thinking indicator—defined as the ability to design logical, sequential, and systematic problem-solving steps—are able to translate mathematical problems into appropriate computational procedures. Students in the high and medium ability categories consistently demonstrated this capability by constructing correct Excel formulas, organizing computational steps, and applying conditional logic where required. In contrast, the failure of some students, particularly those in the low ability category, to produce correct or complete algorithms indicates that they have not yet fully internalized computational thinking as a problem-solving strategy.

These findings are consistent with contemporary computational thinking frameworks, which emphasize algorithmic thinking as a core dimension alongside decomposition and abstraction (Büscher, 2025a). Importantly, algorithmic thinking in this context extends beyond traditional programming environments. The present study shows that algorithmic thinking is equally essential in numerical and mathematical tasks that rely on computational tools such as Microsoft Excel, where students must explicitly define procedures, logical conditions, and calculation flows.

Recent studies further suggest that assessment tasks requiring algorithm design—rather than merely producing numerical answers—are effective in revealing differences in levels of computational thinking ability (Büscher, 2025b)This was evident in the current findings, where students' success in formulating correct Excel expressions reflected not only their technical familiarity with Excel syntax but also their capacity to structure solutions algorithmically. Thus, the ability to design and explain Excel-based algorithms represents a meaningful indicator of computational thinking within mathematics education in the digital era.

In tasks involving conditional logic, such as those using the IF function, students with high and medium ability levels were able to identify possible output outcomes and articulate logical explanations for their reasoning. This indicates that they understood not only how to write formulas but also how decision structures and conditional flows operate within computational systems. Conversely, students in the low ability category often produced random responses or isolated numerical values without procedural or logical justification, suggesting difficulty in translating mathematical logic into computational representations.

The findings demonstrate that students with higher computational thinking ability are able to translate mathematical problems into structured computational procedures. This result reinforces the view that algorithmic thinking in mathematics involves more than procedural fluency; it requires the ability to design, explain, and evaluate systematic solution strategies using computational representations (Büscher, 2025a; Lockwood et al., 2016).

Students' difficulties in applying conditional logic through Excel's IF function further indicate challenges in translating mathematical logic into computational structures. These differences align with findings by (El-Hamamsy & others, 2025), who emphasize that computational thinking assessment must include logical representation and decision-making components.

Abstraction

With respect to abstraction, students demonstrated stronger performance in identifying relevant information than in generalizing computational patterns. Abstraction is a multi-level cognitive process involving information selection, complexity reduction, and the extraction of general rules governing computational behaviours (Lyon et al., 2022; Maharani et al., 2023).

Tasks requiring generalization, such as interpreting IF-based decision rules in the False Position Method, posed greater challenges. Students with high computational thinking ability were able to extract invariant patterns and justify their reasoning, whereas students with lower ability struggled to generalize relationships. These findings are consistent with prior research highlighting abstraction as a critical yet challenging component of computational thinking in mathematics education (Azizah et al., 2022; Fatma Dewi Mardianto et al., 2024). However, abstraction tasks that required generalization, as in Question 4, posed greater challenges. In this question, students were required to formulate a generalization of the Excel output generated by an IF formula applied in one of the solution steps of the False Position method. This task demanded that students interpret conditional logic, identify invariant relationships across iterations, and infer a general rule governing the update of interval endpoints. The ability to derive such generalizations from a computational process represents a higher level of abstraction in computational thinking, as it involves reducing complexity and extracting essential structures from problem representations (Mirolo et al., 2021).

Students with high computational thinking ability were able to extract general rules governing the computational process and justify their answers accurately. Medium-ability students often reached correct conclusions but provided explanations that were incomplete or imprecise, indicating partial abstraction. In contrast, students in the low ability category struggled to generalize patterns or select relevant information, resulting in incorrect answers and weak justifications.

These findings reinforce the notion that abstraction is a complex process involving more than data selection. As defined by (Lyon et al., 2022), abstraction includes identifying relevant information, reducing complexity, and extracting general patterns or rules from available data. When students successfully determine the output of an IF formula based on sign changes within the False Position Method, they are engaging in higher-level abstraction by filtering essential data and drawing generalized logical relationships. The observed variation in performance therefore demonstrates that abstraction is a highly determining indicator of computational thinking in understanding Excel-based numerical approaches.

Comparison of Indicators

A comparison of overall achievement levels across indicators shows that abstraction slightly outperformed algorithmic thinking. While students were generally able to identify relevant information, they encountered more difficulty when required to organize complex computational procedures or generalize patterns from iterative processes. This suggests that lower-level abstraction skills may develop earlier than advanced algorithmic structuring and generalization abilities.

In the context of Excel-assisted numerical methods instruction, these findings highlight the importance of designing learning activities that explicitly integrate both indicators. Although Excel provides a user-friendly computational environment, effective use of the software requires students to engage deeply with computational logic, procedural structure, and pattern recognition. Without targeted instructional support, students may rely on surface-level formula manipulation rather than developing robust computational thinking skills.

CONCLUSION AND SUGGESTION

Based on the analysis of the Midterm Examination (UTS) scores in the Numerical Methods course and an in-depth evaluation of students' responses, this study concludes that the computational thinking skills of UNBAJA students in solving numerical problems using Microsoft Excel are classified at a medium level. The average score of 66.29, with a standard deviation of 25.59, indicates considerable variability in students' abilities, with equal proportions of high- and low-performing students, each accounting for 14.51%. Regarding performance across indicators, the algorithmic thinking indicator achieved a score of 67.34%, while the abstraction indicator reached 67.74%. Overall, the findings suggest that students possess an adequate foundational level of computational thinking; however, further reinforcement is required, particularly in advanced algorithmic processes and higher-level abstraction. The use of Microsoft Excel demonstrates potential as an effective medium for developing computational skills, as it enables students to translate mathematical concepts into digital procedures. This study is limited to a descriptive analysis based on midterm examination data and focuses only on two computational thinking indicators, namely algorithmic thinking and abstraction, without examining other dimensions such as decomposition, pattern recognition, or debugging. Nevertheless, additional instructional support—such as step-by-step explicit instruction, progressive algorithmic exercises, and iterative pattern analysis— is essential to enhance the development of students' computational thinking skills.

REFERENCE

Azizah, N., Suryadi, D., & Herman, T. (2022). Computational thinking in solving sequence and series problems. *Journal of Mathematics Education*, *13*(2), 145–158.

Baker, J., & Sugden, S. (2003). Spreadsheets in Education The First 25 Years. *Spreadsheets in Education*, *1*(1), 18–43. https://www.researchgate.net/publication/27827008_Spreadsheets_in_Education_-The_First_25_Years%0Ahttps://www.researchgate.net/profile/Stephen-Sugden/publication/27827008_Spreadsheets_in_Education_-The_First_25_Years/links/00b7d519519ac05257000000/Spreads

Barendsen, E., Mannila, L., Demo, B., Grgurina, N., Izu, C., Mirolo, C., Sentance, S., Settle, A., & Stupuriene, G. (2015). Concepts in K-9 computer science education. In *ITiCSE-WGP 2015 - Proceedings of the 2015 ITiCSE Conference on Working Group Reports*. https://doi.org/10.1145/2858796.2858800

Basu, S., Kinnebrew, J. S., & Biswas, G. (2014). Assessing student performance in a computational-thinking based science learning environment. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8474 LNCS*, 476–481. https://doi.org/10.1007/978-3-319-07221-0_59

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education [Desarrollando el pensamiento computacional en la Educación obligatoria]*. https://doi.org/10.2791/792158

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

Büscher, C. (2025a). Assessing algorithmic and abstraction dimensions of computational thinking. *Computers & Education*, *195*, 104722.

Büscher, C. (2025b). Differences in Students' Computational Thinking Activities when Designing an Algorithm for Drawing Plane Figures. *International Journal of Science and Mathematics Education*, *23*(2), 365–386. https://doi.org/10.1007/s10763-024-10465-3

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33–39. https://doi.org/10.1145/2998438

El-Hamamsy, L., & others. (2025). Designing valid assessment tasks for computational thinking. *Educational Technology Research and Development*, *73*(1), 89–110.

Fatma Dewi Mardianto, N., William Iskandar Muda Ps, J. V, Estate, M., Percut Sei Tuan, K., & Deli Serdang, K. (2024). Systematic Literature Review: Penerapan Berpikir Komputasi Dalam Pembelajaran Matematika Yahfizham Universitas Islam Negeri Sumatera Utara. *Journal of Student Research (JSR)*, *2*(4), 41–55. https://doi.org/10.55606/jsr.v2i4.3082

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Guggemos, J., Seufert, S., & Sailer, M. (2024). Measuring computational thinking: Challenges and perspectives. *Computers & Education*, *196*, 104726.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569. https://doi.org/10.1016/J.CHB.2017.01.005

Lockwood, E., Asay, A., DeJarnette, A. F., & Thomas, M. (2016). Algorithmic thinking: An initial characterization of computational thinking in mathematics. *38th Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*, 1588–1595.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. https://doi.org/10.1016/J.CHB.2014.09.012

Lyon, J. A., Magana, A. J., & Vieira, C. (2022). Abstraction and pattern generalization in computational problem solving. *Journal of Engineering Education*, *111*(2), 356–378.

Maharani, S., Pramudya, I., & Kuswardi, Y. (2023). Abstraction skills in computational thinking-based mathematics learning. *Journal on Mathematics Education*, *14*(1), 97–112.

Mirolo, C., Izu, C., Lonati, V., & Scapin, E. (2021). Abstraction in Computer Science Education: An Overview. *Informatics in Education*, *20*(4), 615–639. https://doi.org/10.15388/INFEDU.2021.27

Nainggolan, R., Sihotang, H., & Simarmata, J. (2024). Classification of students' problem-solving abilities using descriptive statistics. *Journal of Mathematics Education Research*, *8*(1), 21–34.

Penelitian, J. H., Kepustakaan, K., & Pendidikan, B. (2024). *Jurnal Kependidikan: 10*(2), 640–653.

Permatasari, D. (2024). Interpretation of computational thinking indicator achievement levels. *Indonesian Journal of Mathematics Education*, *5*(2), 66–78.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/J.CHB.2016.08.047

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. *ITiCSE Conference Proceedings*, 5–8.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158.

Taufik, M., & Susanti, R. D. (2024). Solving numerical method problems with mathematical software: Identifying computational thinking. *Pedagogical Research*, *9*(3). https://doi.org/10.29333/pr/14583

Villamin, A., Cruz, J. P., & Santos, R. (2025). Qualitative descriptive research in education. *International Journal of Educational Research*, *121*, 102115.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of*

Science Education and Technology, *25*(1), 127–147.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.