IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Exploring The Role Of Enhancing Computational Thinking Skills In Mathematics Education: A Systematic Literature Review

[1] Harijana Manohar [2] Prof G. Viswanathappa

[1] Research Scholar, Department of Education, Regional Institute of Education-NCERT, Mysore

[2] Professor in Education, Department of Education, Regional Institute of Education-NCERT, Mysore

ABSTRACT

The word computational thinking has emerged as a significant catalyst in transforming the way pupil engage and comprehend mathematics in the dynamic landscape of education. Despite the conventional realm of mathematical problem solving the computational thinking provides an alternative view point that align with the requirement of digital world. The study investigates computational thinking (CT) as a transformative catalyst in mathematics instruction. The paper conducts a thorough literature review to analyze multiple definitions of CT across disciplines, defines its essential components relevant to mathematics, and studies efficient integration procedures for mathematical problem solving. The study shows that four critical CT components such as decomposition, pattern recognition, abstraction, and algorithms are consistent with mathematics education frameworks and can considerably improve students' problem-solving abilities. The study also investigates pedagogical approaches that use both digital (plugged) and non-digital (unplugged) activities to enhance CT skills. The findings show that problem-based learning is particularly successful for incorporating CT into mathematics education. The study contributes to our understanding of how computational thinking can be systematically integrated into mathematics classrooms to prepare students for 21st-century issues, implying that careful curriculum design and teacher training are required for successful implementation.

Key words: Computational Thinking, Mathematics Education, Pedagogical approaches, Role of Computational thinking skills.

1.Introduction:

The term computational thinking (CT) has evolved and gained popularity as a research subject in the field of education in the twenty-first century. Computational thinking (CT) is one of the talents required in the twenty-first century to tackle problems in today's technological environment. Historically, Seymour Papert coined the term computational thinking in the area of computer science. Seymour Papert is an American computer scientist and mathematician who develops child-friendly activities using computer programming languages. In 1980, the word CT appears without explanation in his book Mindstorms (page 182). Later, in 1996, Seymour Papert explained computational thinking in his paper titled "An Exploration in the Space of Mathematics Education" as the principle of object is to develop concepts through computational thinking, which gives mathematical ideas in the form of operations, representations, and thinking about them, and that development is explained with turtle geometry and logo programming. (Ramanujam R. 2022). However, it did not gain popularity among students for developing computational thinking skills through digital technology for a long time until professor Jennette Wing entered the picture with the communication of the association with respect to computer machinery in 2006 (Wing,2006).

However, Jennette Wing (2006&2008) characterised CT in her key presentation as "computational thinking is an essential skill for everyone, not just computer scientists". She also describes it as a globally applicable "set of attitude and skills" that everyone may acquire. She went on to say that computational thinking encompasses problem solving, system design, and understanding human behaviour by drawing on core notions from computer science. Though computational thinking skills encompass a wide range of mental tools that reflect on the field of computer science through plugged activities using technology, we must also understand the type of thought process used in other disciplines, particularly mathematics and science, to solve complex problems. Computational thinking is no longer confined to computer science; it has found applications in various fields such as the life sciences (Arık & Topçu, 2022), music (Bell & Bell, 2018), mathematics (Weintrop et al., 2016; Benakli et al., 2017), general science (Basu et al., 2017), robotics (Jaipal Amani & Angeli, 2017; Yi Wu & Sheng Su, 2021), as well as across both the sciences and the arts (Lin et al., 2020; Subramaniam S et al., 2022).

More ever, (Seymour Papert,1996) stated that while the computer can assist in obtaining an effective solution, computational representation in mathematics will make it more visible and will also improve creativity and invention, hence aiding in the development of the child's mental process. Computational thinking requires a wide range of cognitive skills, such as breaking down problems, analyzing and abstracting information, developing algorithms, recognizing patterns, thinking iteratively, transforming ideas, minimizing errors while ensuring accuracy, and applying intuitive reasoning. These skills are critical for the development of problem-solving skills, and everyone must learn CT as a fundamental talent in engaging mathematical concepts. CT, like reading, writing, and math, is a crucial talent in the twenty-first century (Liu et al., 2022). Many research and academicians have stated that computational thinking is a necessary 21st century ability that everyone, not only computer scientists, should master (So et al., 2020; Subramaniam S et al.,2022).

2. Rational for the study:

Learning mathematics is one of the major concerns in mathematics education, and students, regardless of gender, believe it is one of the most difficult subjects to learn at the basic education level. However, there are many factors that make mathematics learning difficult, even generating negative thoughts among students to learn mathematics, particularly at the middle stage level (Mohd Rustam & Mohd Rameli 2016). Mathematics and numeracy equip us with essential concepts and skills to understand our environment and the broader world. When taught effectively, mathematics can be enjoyable and may even develop into a lifelong passion (Draft NCFSE, 2023, p. 175). The PISA 2021 framework emphasizes that students should showcase computational thinking abilities when using mathematics to solve problems. It also suggests that countries involved are expected to prioritize integrating computational thinking into their mathematics curricula and teaching practices. emphasizing the significance of computational thinking as it relates to mathematics. As a result, there is a need to focus on developing student thinking processes to solve complex problems through pedagogical practices that promote and aid in the development of various skills such as computational thinking skills, mathematical concept application, and the ability to use multiple problem-solving strategies. According to the National Education Policy (NEP 2020), "it is recognised that mathematics and mathematical thinking will be very important for India's future and India's leadership role in the numerous upcoming fields and professions which involve the thinking process, artificial intelligence, machine learning, data science, etc." (Para 4.25, above). (NCFSE draft, 2023) also said that mathematics is developed by identifying patterns, making conjectures (i.e., proposed truths), and then verifying those conjectures using logical and rigorous reasoning. The process of identifying patterns, making conjectures, and finding proofs or counter-examples frequently requires a tremendous amount of creativity. In classroom discussions, patterns should require creativity to discover and creativity to explain; problems should also require creativity to solve, and in many cases, allow for multiple approaches, some of which the teacher may be unaware of because this is the nature of mathematical knowledge discovery (Draft NCFSE P,178). As a result, it emphasized that mathematics, mathematical thinking, and computational thinking are extremely important for children, as well as nurturing them with appropriate ways to be given more priority throughout the school years.

Huiyan Ye et al. (2023) noted that more research is needed to determine how computational thinking may be effectively applied to promote K-12 mathematics learning. Hichmott et al. (2018) also emphasized the necessity for studies that connect mathematical principles with computational thinking. In light of this, the current paper provides insights into the integration or role of computational thinking skills in mathematics, notably in problem solving.

3. Statement of the problem:

Exploring The Role of Enhancing Computational Thinking Skills in Mathematics Education: A systematic Literature Review

4. Research Objectives:

1. To Examine the definition of computational thinking and its components with reference to mathematics.
2. To Explain the integration of computational thinking skills in solving mathematics problems in mathematics classrooms.
3. To find out pedagogical approaches used in developing computational thinking skills with plugged activities and un plugged activities.

5.Methodology:

The current study used a systematic literature review (SLR) approach to evaluate the concepts and components of computational thinking (CT) in mathematics education. Data was acquired from significant academic databases, such as ERIC, ResearchGate, ACM and Google Scholar, using keywords like "computational thinking," "mathematics education," "CT skills," and "CT components." The inclusion criteria centered on peer-reviewed journal publications and conference papers published between 2015 and 2025, written in English, and directly connected to CT in mathematics teaching. Studies that did not meet these standards were excluded. An initial search generated 215 papers, which were reviewed for relevancy using titles and abstracts. After deleting duplicates and irrelevant records, 82 articles remained for full-text analysis. Following a rigorous examination using predetermined inclusion and exclusion criteria, 38 papers were chosen for the final synthesis. Key information from these papers was collected and evaluated thematically to uncover reoccurring themes, definitions, and CT components. Diverging interpretations were handled through cross-comparison of findings, which improved the synthesis's dependability and comprehensiveness.

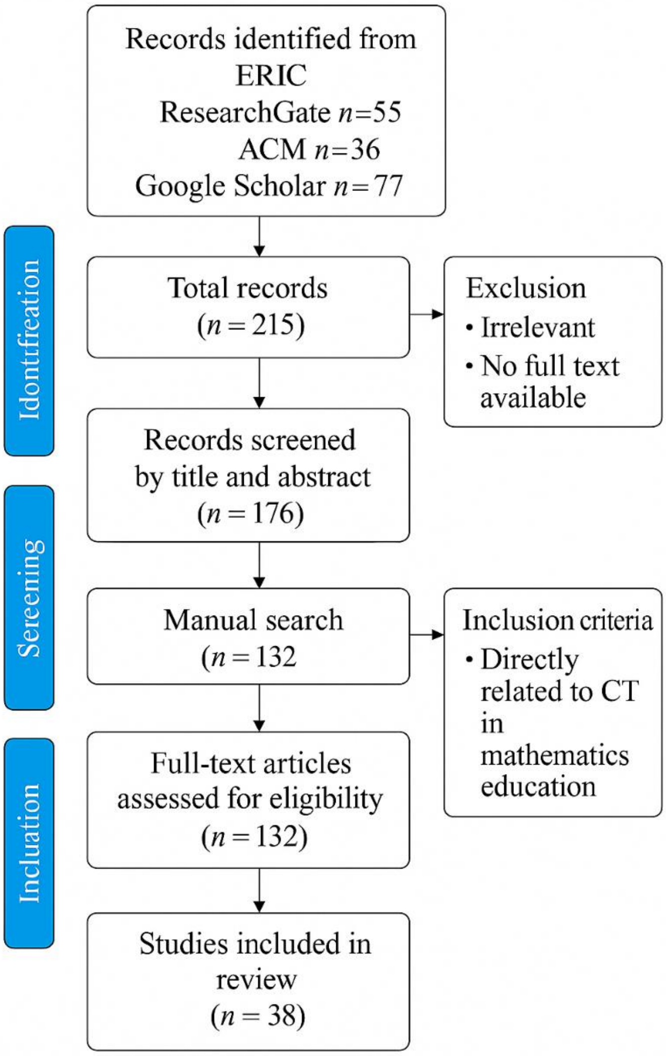

Fig N0: 01 The PRISMA flow diagram of studies identification, screening, included for study.

6.Results:

Objective one: To Examine the definition of computational thinking and its components with reference to mathematics.

In order to attain the first objective of the study namely "To Examine the definition of computational thinking and its components with reference to mathematics" the researcher collected computational thinking definition defined by different researchers. The data collected from different sources such as google scholar, ERIC, research gate, etc.

Computational Thinking: Evaluation and Definition

Seymour Papert (1980), a mathematician and computer scientist renowned for his contributions to artificial intelligence and the development of the Logo programming language, was the first to introduce the term computational thinking (CT) in his seminal work Mindstorms. In the evolving context of the 21st-century digital revolution, Wing (2006) emphasized the significance of computational thinking as a crucial approach for problem-solving, system design, and understanding human behavior, highlighting its role in grasping fundamental concepts in computing. Later, Wing (2008) further refined the concept, defining computational thinking as the cognitive process involved in formulating problems and developing solutions that can be effectively executed by an information-processing agent. This understanding gained wide acceptance, with Cuny and Snyder (2010) endorsing and building upon Wing's conception.

Expanding on the concept, Denning (2017) described computational thinking as the thought process involved in designing, using, and evaluating computational tools, stressing that such processes are integral to the development of cognitive abilities and the performance of various activities. Similarly, Grover and Pea (2018) broadened the scope of computational thinking by defining it as the thought processes involved in formulating problems and expressing their solutions in ways that can be effectively executed by computers, humans, or machines. Their definition placed strong emphasis on leveraging internal cognitive processes within real-world contexts. International organizations like the BBC and the Bowers Institute (US) have also contributed to shaping the understanding of computational thinking. They define it as a universal problem-solving process applicable across various content areas and everyday life contexts. According to their framework, computational thinking involves breaking down complex problems into smaller parts (decomposition), identifying patterns (pattern recognition), focusing on relevant information (abstraction), and formulating step-by-step strategies for solutions (algorithms).

Similarly, the Association for Computing Machinery (ACM) characterizes computational thinking as a method of problem-solving that entails breaking down complex problems into manageable subproblems and applying techniques from computer science to develop effective solutions. Despite growing research interest, there continues to be some confusion and debate over the precise boundaries of computational thinking. Various researchers including Denning (2009), Hemmendinger, Cuny, Snyder, Wing (2010), Yadav et al. (2014), and Weintrop et al. (2016) have attempted to contextualize computational thinking within computer science education, particularly through the use of digital tools, as well as within mathematics and science disciplines. Most definitions consistently highlight the cognitive nature of computational thinking, viewing it as a process aimed at enhancing individuals' cognitive and problem-solving skills. Kilcarslan Sibel et al. (2019) concluded that computational thinking activities are designed to enhance cognitive skills and to support teaching and learning across different school subjects. Nevertheless, debates persist regarding what computational thinking truly entails and whether it necessarily requires programming and coding, or if it can be nurtured independently through non-programming activities. While the majority of studies have linked computational thinking closely to coding and programming practices, fewer investigations have explored its development through mathematics, science, and other academic domains. In synthesizing these perspectives, it becomes evident that although there is no universally agreed-upon definition of computational thinking, a common thread runs through the literature: it is a cognitive process that cultivates essential thinking skills such as logical reasoning and critical thinking. Computational thinking is increasingly recognized as a fundamental skill to reading, writing, and arithmetic that every individual must develop to succeed in the 21st-century world.

Components of Computational Thinking:

The fundamental components of computational thinking are also divergent from researcher to researcher. There are some components related to the context of computer science and some components related to mathematics and science disciplines.

Components	Authors
Decomposition, Abstraction, Pattern recognition, Algorithms	Wing 2008
Abstraction, Algorithms, automation, problem decomposition, Parallelization, simulation	Barr & Stepheson (2011)
Abstraction, Automation, Analysis	Lee et.al, (2011)
Abstraction, Algorithmic thinking decomposition, Evaluation, Generalization	Selby &Woollard (2013)
Decomposition, Abstraction, Pattern recognition, Algorithms	BBC, ACM
Decomposition, Abstraction, Pattern recognition, Algorithms	Bowers Institute of Tech Organizations US

Table 1: Computational thinking components based on researcher to researcher.

From the above table we can observe that, the computational thinking skills include abstraction, algorithmic thinking, cooperativity, creativity, critical thinking, data analysis, debugging, decomposition, problem-solving, and recursive thinking. While components differ based on the researcher to researcher but the essential components are largely uniform across the field were four basic components such as Decomposition, pattern recognition, abstraction, and algorithm. Many organizations like BBC, ACM, and the Bowers Institute in the United States of America (US) also defined as the same. According to the Draft NCFSE 2023 (p. 187), one of the curricular goals for the middle stage highlights the development of computational thinking by encouraging students to tackle complex problems by breaking them down into simpler, manageable parts. To foster computational thinking in mathematics classrooms, the draft recommends that students practice decomposing problems into sub-problems, analyzing sequences of instructions, identifying patterns by comparing similarities and differences across problems, and engaging in algorithmic thinking to create solutions. The Draft (NCFSE,2023 p, 187) recognized that developing basic skills of mathematics and capacities of computational thinking namely decomposition, pattern recognition, abstraction, algorithms is a part of curricular goals for the secondary stage. Where such techniques and skills are important in order to solve mathematics effectively and also, stated that computational thinking means "deals with complex problems and is able to break down them into a series of simple problems that can be solved by suitable procedures/algorithms. Unique components of computational thinking based on widely used components in the mathematics and science discipline are defined as follows.

1. **Decomposition:** Breaking down problems into smaller problems.

   Breaking down problems into smaller parts can make complicated challenges more manageable. This enables other computational thinking elements to be applied more effectively to complex challenges or problems. The solutions to the smaller problems are then combined to solve the original or larger problem. The role of the teacher at this stage is to create an environment for students in the problem-solving scenario and share the complex, multi-step problems. Thereafter, facilitate a conversation with students that will help students to break down larger problems into smaller problems. We can advance students' thinking by making use of early children's natural propensity to explore and play as well as by encouraging problem-solving abilities. While encouraging playful thinking, computational thinking provides structure so that the abilities students are developing can later be applied to more challenging tasks.

2. **Pattern Recognition (**Recognizing patterns, similarities, and determining the sequence)

   Identifying patterns or connections to previously solved problems can make finding solutions easier. Recognizing patterns also helps in organizing, grouping, and simplifying problems, leading to more efficient problem-solving. Pattern recognition invites students to analyze similar objects, steps, and experiences, and identify commonality in the problem in order to solve the problem. By identifying similar objects, steps, and experiences have in common, children can begin to develop an understanding of trends of the problem and thereafter will be able to predict. The role of the teacher at this stage is to facilitate students to identify or recognize patterns that are involved and make awareness of the problem in the context of discipline and make awareness of the real-life world around them.

3. **Abstraction**: Generalization of a problem - focus on the big picture and what is important

   Taking a step back from the specific details of a given problem allows you to create a more generic solution. This requires analysing the problem to remove extra detail and highlight the basic parts. Once completed, begin brainstorming a solution to the problem. Teaching abstraction to the students, they will be able to sort all of the information available and identify specific information which is needed to solve the problem. Doing abstraction is an invaluable skill as students read the larger text and complex problems.

4. **Algorithms:** Step-by-step instructions to solve a problem

When solving a problem, it is important to create a plan for your solution. Algorithms are a strategy that can be used to determine step-by-step instructions on how to solve a problem. Algorithms can be written in plain language, with flowcharts.

Further extending meaning as an Algorithm helps students to develop a solution to the problem, it creates sequential rules to follow in order to solve the problem. Conversations about sequence and developing the foundation of algorithmic thinking will help to achieve desired results. Particularly, in mathematics to solve problems various thinking process plays a major role. In that computational thinking skills are closely related to problem-solving and analytical skills. In order to develop or practice computational thinking skills viz., Decomposition, Pattern recognition, Abstraction, Algorithms in the teaching learning process. These components play a major role to become a problem solver in solving mathematics problems.

Objective Two: To Explain the integration of computational thinking skills in solving mathematics problems.

The integration of computational thinking skills within mathematics domain recognized as more vital approach to enhance the computational thinking skills in problem solving method and to prepare children for the demand of 21st century. Research from various data base highlighted the integration of computational thinking skills multiple ways and it can be inculcated in mathematics and positive impacts observed from the key research papers.

Key findings:

CT as a Tool for Improved Mathematical Problem Solving: CT can be a very effective tool for improving mathematical problem-solving, especially when dealing with complicated issues that call for a number of calculations or the investigation of several possible outcomes. Students can automate calculations in computational environments, which makes it easier for them to test solutions in different scenarios and investigate extreme situations (Berkaliev et al., 2014; Pei et al., 2018; Cui &Ng,2021).

Creation of Heuristics for Solving Mathematical Problems:

Common mathematical problem-solving heuristics can be developed and applied through computational problem-solving activities. For example, the mathematical heuristic of breaking down problems into subgoals is in line with the CT process of breaking down a complex problem into smaller, more manageable subproblems (Lehmann, 2025; Cakiroglu & Mumcu 2020).

Commonly Used CT Skills in Mathematical Contexts:

Research looking at how students solve problems when given programming tasks shows that skills like abstraction, debugging, and algorithmic thinking—which uses loops, conditionals, and sequences—are commonly used. However, depending on the type of mathematical problems, decomposition and pattern recognition may occur less frequently (Cui & Ng, 2021; Cui et al., 2023; Hong et al., 2021; Ng & Cui, 2021).

Integration Flexibility:

Key components of this integration include the ability to translate mathematical ideas and procedures straight into a programming language (translation flexibility) or to select from a variety of CT strategies for resolving an issue that calls for various mathematical approaches integrated approach (Kallia et al., 2021).

Curriculum Integration and Teacher Training:

Including CT in math classes frequently entails creating instructional resources and exercises that particularly promote CT proficiency. Additionally, to properly support this integration, training methods for math teachers must be developed. Effect on Cognitive and Affective Domains: Studies show that incorporating CT into math instruction enhances students' cognitive (such as their ability to solve problems and reason logically) and affective (such as their creativity, confidence, and involvement in the learning process) aspects (Fauzi et al., 2024).

Problem-Based Learning as a Successful Method:

One promising pedagogical strategy for incorporating CT into mathematics education is problem-based learning (PBL). PBL encourages students to use CT skills like decomposition, pattern recognition, abstraction, and algorithm design to solve complex problems by focusing learning around them (Sihotang et al., 2023; Nurasiah et al., 2023). Research indicates that when PBL models are used, students' mathematical computational thinking skills improve (Sihotang et al., 2023).

In conclusion, integrating computational thinking skills into mathematics education has the potential to significantly improve students' problem-solving abilities by providing tools for complex calculations, encouraging the use of mathematical heuristics, and engaging specific CT skills such as algorithmic thinking and abstraction. Research demonstrates the adaptability of this integration across curricula and its favorable influence on both cognitive and affective learning domains, with problem-based learning emerging as a particularly effective pedagogical strategy. While the findings highlight the advantages of this integration, careful curriculum design, teacher training, and rigorous tests for originality are required for successful implementation.

Objective 03: To find out pedagogical approaches used in developing computational thinking skills with plugged activities un plugged activities.

The review of research literature identified numerous significant educational approaches and methodologies used to build computational thinking (CT) skills through both plugged (using digital technology) and unplugged (no digital technologies) activities.

Plugged Activities:

Digital tools and programming environments have considerably improved the integration of computational thinking (CT) into mathematics instruction via a variety of novel ways. One such approach is dynamic mathematical modeling, in which students utilize tools like GeoGebra to create interactive geometric creations that represent algorithmic thinking (Benton et al., 2017). Spreadsheets are another tool that allows students to break down large issues into digestible chunks and create models for mathematical solutions (Weintrop et al., 2016).

Another useful method is to employ visual programming environments to make abstract mathematical concepts more concrete. For example, students use Scratch to visualize and analyze number patterns and sequences, which promotes both creativity and computational logic (Benton et al., 2018). Blockly offers a visual interface for creating algorithms to solve algebraic equations, encouraging structured problem solving (Grover & Pea, 2018). Logo, invented by Papert (1980), is still significant because it allows for the examination of geometric transformations and symmetry using programmable techniques (kafai&Burke,2014).

Furthermore, programming simulations provide a hands-on approach to exploring probability and statistical distributions, enhancing conceptual understanding through iterative experimentation. Digital tools and programming environments have considerably improved the integration of computational thinking (CT) into mathematics instruction via a variety of novel ways. One such approach is dynamic mathematical modeling, in which students utilize tools like GeoGebra to create interactive geometric creations that represent algorithmic thinking (Benton et al., 2017). Spreadsheets are another tool that allows students to break down large issues into digestible chunks and create models for mathematical solutions (Weintrop et al., 2016). Furthermore, programming simulations provide a hands-on approach to exploring probability and statistical distributions, enhancing conceptual understanding through iterative Experimentation (Psycharis&Kallia,2017).

Interactive data analysis exercises connect mathematics and CT by encouraging students to interact with data in meaningful ways. Students learn to analyze and interpret mathematical data by creating interactive dashboards, which provide insights via dynamic representations (Weintrop et al., 2016). They also design methods for sorting, classifying, and recognizing patterns in datasets, which are critical abilities in data science and computational reasoning (Grover et al., 2015). Hands-on computational exploration reinforces learning of statistical measures such as central tendency and dispersion (Lee et al., 2014).

Finally, mathematical problem-solving through programming allows pupils to go beyond traditional methods. They use numerical methods to solve equations that do not have analytical solutions, offering up new areas for mathematical research (Blikstein and Wilensky, 2009). Iterative coding techniques facilitate the examination of complex issues such as limits, convergence, and recursion, resulting in a deeper understanding (di Sessa, 2000). Furthermore, programming helps students to validate hypotheses and investigate edge cases, which fosters mathematical curiosity and precision (Gadanidis et al., 2017). Collectively, these initiatives demonstrate how computational thinking, when integrated into mathematics education via digital tools and programming, may change learning into a more dynamic, exploratory, and problem-solving experience.

Unplugged activities approach:

Unplugged activities are important in incorporating computational thinking (CT) concepts into mathematics teaching, particularly for younger students. Hands-on, screen-free activities like sorting games, logic puzzles, and making flowcharts with index cards make abstract ideas like algorithms, decomposition, pattern recognition, and abstraction more tangible and accessible (Bell et al., 2012; Curzon et al., 2014; Tang et al., 2020). These experiences stress collaboration and problem-solving without the use of digital technologies, and they provide as a solid basis for entering CT through ordinary classroom practices.

Unplugged exercises focus on building algorithmic thinking through familiar mathematics concepts. Activities like creating step-by-step instructions for solving multi-step equations (Curzon et al., 2014) or following precise methods to make geometric shapes with a compass and straightedge (Bell et al., 2009) mimic the logical sequencing important to algorithms. Students can also create flowcharts to illustrate and represent solution strategies for word problems, demonstrating how structured thinking promotes problem solving (Csizmadia et al., 2015).

In parallel, pattern recognition is an important CT talent that is built into number systems. Learners see patterns in numerical sequences and use them to create general rules or formulas (Yadav et al., 2014). Manipulatives assist pupils understand repeating structures in multiplication tables and modular arithmetic. Furthermore, activities like designing and deciphering mathematical cipher systems enable children to recognize underlying patterns and rules (Brackmann et al., 2017).

Another basic CT idea is abstraction, which is essential in mathematical thinking. When students use variables and algebraic notation to describe broad mathematical concepts, they engage in abstraction (Wing, 2008). They also use abstraction while generating generalized formulas from specific cases utilizing inductive reasoning (Kilcarslan Sibel et al., 2019) or modelling real-world issues by focusing only on fundamental elements (Sengupta et al., 2013).

Logical thinking, as seen in geometry, is closely related to abstraction. Geometric proofs help students to think algorithmically by logically ordering stages to a conclusion (Kotsopoulos et al., 2017). Conditional statement activities and investigating their consequences help students comprehend logical structures (Brennan & Resnick, 2012), whereas deductive reasoning helps students develop and justify geometric relationships (Bowles et al. 2018).

Computational mathematics labs are hybrid spaces that blend physical and digital learning to help deeper integrate CT into math instruction. Students, for example, may start using tangible manipulatives to grasp a concept before moving on to programming-based models to expand their understanding (Caldwell & Smith, 2017). Card-sorting tasks are useful for introducing sorting algorithms in an unplugged manner before going on to actual coding. Similarly, creating actual geometric models and digitally recreating them allows students to investigate invariant mathematical features (Benton et al., 2017).

Problem decomposition, a concept that allows students to approach complex mathematical challenges with greater ease, is central to CT. This can include breaking down multi-step word problems into smaller sub-tasks (Román-González et al., 2016), dissecting difficult geometric forms into simpler shapes (Toh et al., 2016), or dividing hard numerical calculations into smaller operations (Shute et al., 2017). Decomposition not only clarifies the problem-solving process, but it also helps students improve their analytical skills.

Furthermore, combining mathematics and CT across fields promotes a more comprehensive knowledge. For example, examining musical patterns and designing algorithms to construct musical sequences mixes mathematics and art (Burke & Kafai, 2012). Similarly, students can create computational models to simulate scientific phenomena with mathematical underpinnings (Weintrop et al., 2016) or use data analysis tools to investigate mathematical relationships in social sciences (Moreno-León & Robles, 2015), demonstrating that CT has applications outside of traditional STEM disciplines.

Finally, project-based mathematical inquiry combines all of these features by immersing students in extended, authentic activities. CT can be meaningfully embedded in math education through projects such as designing and coding geometric art using concepts like symmetry and iteration (Kafai & Burke, 2014), creating interactive games that require algorithmic and logical thinking (Hsu et al., 2018), or developing computational models for solving open-ended optimization problems (Sengupta et al., 2013). These initiatives not only promote deeper learning, but they also demonstrate the creative and multidisciplinary possibilities of computational thinking in the classroom.

7. Major findings of the study

The review indicated that while definitions of computational thinking (CT) have varied since early contributions by Papert (1980) to Wing (2006) and beyond, a consistent understanding has emerged: CT is a cognitive technique designed to solve complicated issues efficiently. Four fundamental components of CT are consistently identified across numerous theoretical frameworks: decomposition, pattern identification, abstraction, and algorithmic reasoning. These features are also represented in the Indian National Curriculum Framework for School Education (NCFSE) 2023, which emphasizes CT as an important educational goal. In the context of mathematics, each component is critical: decomposition allows learners to break down complex problems; pattern recognition aids in the identification of trends and regularities; abstraction facilitates the focus on relevant mathematical features; and algorithmic thinking guides the creation of systematic solutions

The incorporation of CT into mathematics teaching enhances both conceptual understanding and procedural proficiency. It enables students to try out different solution strategies and improves mathematical reasoning by incorporating heuristic methods like trial and error, working backward, and logical sequencing. The capacity to communicate between mathematical ideas and computational representations is an important aspect in this integration, since it increases students' flexibility and depth of comprehension. Among the various pedagogical models, problem-based learning (PBL) is particularly effective in promoting CT. It encourages discovery, inquiry, and creativity while also increasing problem-solving abilities, confidence, and involvement. The review additionally examined at the instructional strategies used to improve CT skills. A variety of plugged-in activities, such as using digital tools like GeoGebra, Scratch, Blockly, and Logo, can help students visualize mathematical concepts and computational processes. In contrast, unplugged techniques like paper and pen activities such as sorting games, logic puzzles, and flowchart exercises to build CT thinking without the use of computers. Furthermore, interactive data analysis tasks and computational mathematics labs provide rich, mixed environments for hands-on learning. Finally, project-based inquiry was discovered to be an especially effective approach, integrating all CT components through long-term, real-world tasks that encourage both individual and collaborative learning.

8. Conclusion:

This study found that computational thinking is an effective Approach for improving mathematics instruction, which is similar with the findings of Weintrop et al. (2016) and Benakli et al. (2017). While our findings support Wing's (2008) assertion that CT is crucial for everyone, not only computer scientists, we discovered greater links between mathematical thinking and computational thinking than previously suggested in the literature. Unlike Papert's (1996) emphasis on computer-based tools, our findings lend support to Grover and Pea's (2018) approach, which emphasizes mental processes above technical implementation. The study builds on Cui and Ng's (2021) work by identifying distinct instructional tactics for both plugged and unplugged activities. Our findings build on Sihotang et al.'s (2023) study on problem-based learning, providing a more comprehensive framework for implementation in a variety of educational environments. This integrated approach to computational thinking in mathematics education provides a viable avenue for acquiring 21st-century abilities and deepening mathematical comprehension.

References:

Arık, S., & Topçu, M. S. (2022). Integrating computational thinking into life sciences. Journal of Science Education and Technology, 31(4), 456–470. https://doi.org/10.1007/s10956-021-09999-9

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48–54. https://doi.org/10.1145/1929887.1929905

Basu, S., Kinnebrew, J. S., & Biswas, G. (2017). Learning physics through computational modeling. Interactive Learning Environments, 25(3), 276–293. https://doi.org/10.1080/10494820.2016.1276080

BBC. (n.d.). Computational thinking resources for education. BBC Bitesize. https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1

Benakli, N., Leis, R., Morse, J., & Steinberg, L. (2017). Integrating computational thinking into undergraduate mathematics curricula. International Journal of Research in Undergraduate Mathematics Education, 3, 167–188. https://doi.org/10.1007/s40753-017-0040-4

Berkaliev, M., Golovachev, A., & Ivanov, D. (2014). Enhancing mathematical problem-solving with computational thinking.

Bowers Institute of Technology Organizations. (n.d.). Computational thinking framework.

Cakiroglu, U., & Mumcu, F. K. (2020). Exploring the role of computational thinking in mathematics education. Education and Information Technologies, 25(1), 1–17. https://doi.org/10.1007/s10639-019-09974-1

Cui, L., & Ng, O. L. (2021). Integrating computational thinking into secondary mathematics classrooms: A case study. Journal of Computers in Education, 8(2), 213–234. https://doi.org/10.1007/s40692-020-00168-5

Cui, L., Ng, O. L., & Wang, D. (2023). Students' computational thinking practices in mathematical contexts. International Journal of STEM Education, 10(1), 12. https://doi.org/10.1186/s40594-023-00345-6

Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists.

Denning, P. J. (2009). Beyond computational thinking. Communications of the ACM, 52(6), 28–30. https://doi.org/10.1145/1516046.1516054

Draft National Curriculum Framework for School Education (NCFSE). (2023). National Curriculum Framework for School Education 2023 (Draft). Ministry of Education, Government of India. https://ncert.nic.in/pdf/ncf/ncfse2023.pdf

Fauzi, A., Farid, S., & Fikri, A. (2024). Impact of computational thinking integration on cognitive and affective domains. Journal of Educational Computing Research, 62(3), 345–362. https://doi.org/10.1177/07356331231234567

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. Computer Science Education, 28(1), 1–14. https://doi.org/10.1080/08993408.2018.1433170

Hichmott, K., Maynard, C., & Chalmers, C. (2018). The need for integration of computational thinking in mathematics education. Australian Mathematics Teacher, 74(1), 3–7.

Hong, J., Chai, C. S., & Wong, B. (2021). Programming tasks and problem-solving strategies in mathematics education. Journal of Educational Technology & Society, 24(2), 45–58.

Jaipal-Jamani, K., & Angeli, C. (2017). Developing elementary school teachers' competencies in computational thinking through practice-based professional development. Computers in Human Behavior, 76, 548–556. https://doi.org/10.1016/j.chb.2017.03.038

Kallia, M., Sentance, S., & Humphreys, S. (2021). Flexibility in integrating computational thinking into mathematics learning. Education and Information Technologies, 26(3), 345–362. https://doi.org/10.1007/s10639-020-10345-6

Lehmann, K. (2025). Developing heuristics for mathematical problem solving through computational thinking.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., & Werner, L. (2011). Computational thinking for youth in practice. ACM Inroads, 2(1), 32–37. https://doi.org/10.1145/1929887.1929902

Lin, Y., Zhang, X., Zheng, L., & Chen, L. (2020). Computational thinking across science and art: An interdisciplinary approach. International Journal of STEM Education, 7(1), 1–14. https://doi.org/10.1186/s40594-020-00218-y

Liu, S., Zhang, D., Wang, Y., & Liu, J. (2022). The role of computational thinking in solving complex problems.

Ministry of Education, Government of India. (2020). National Education Policy 2020. https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf

Mohd Rustam, M. Z., & Mohd Rameli, M. R. (2016). Students' difficulties in learning mathematics at the basic education level.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books.

Papert, S. (1996). An exploration in the space of mathematics education. International Journal of Computers for Mathematical Learning, 1(1), 95–123. https://doi.org/10.1007/BF00191473

Pei, Y., Chai, C. S., Wong, B., & King, R. B. (2018). Students' learning experiences in computational thinking-enhanced mathematics classrooms. Educational Technology & Society, 21(4), 37–50.

Ramanujam, R. (2022). Computational thinking: Foundations and future directions.

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. University of Southampton. https://eprints.soton.ac.uk/356481/

Sihotang, M., Ardiansyah, A., & Widodo, S. A. (2023). The effect of problem-based learning on computational thinking skills. International Journal of Instruction, 16(2), 45–62. https://doi.org/10.29333/iji.2023.1624a

So, H. J., Lee, T. T., & Toh, Y. (2020). Fostering 21st-century competencies through computational thinking. Asia Pacific Journal of Education, 40(1), 1–14. https://doi.org/10.1080/02188791.2019.1572591

Subramaniam, S., Cooper, S., Krumholz, K., & Haynie, K. (2022). Cross-disciplinary computational thinking education: Current trends and challenges.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wu, Y., & Su, S. (2021). Using computational thinking to improve learning in robotics. International Journal of STEM Education, 8(1), 28. https://doi.org/10.1186/s40594-021-00291-7

Ye, H., Yan, J., & Chen, M. (2023). Computational thinking and K–12 mathematics learning: A review of empirical studies. Educational Research Review, 38, 100456. https://doi.org/10.1016/j.edurev.2022.100456