Between computational thinking and literacy in mathematics education

Elicer, Raimundo

Published in:
Proceedings of the 48th Conference of the International Group for the Psychology of Mathematics Education

Creative Commons License
Unspecified

Publication date:
2025

Document Version
Accepted author manuscript, peer reviewed version

BETWEEN COMPUTATIONAL THINKING AND LITERACY IN MATHEMATICS EDUCATION

Raimundo Elicer

Department of Sustainability and Planning, Aalborg University, Denmark

In this essay, I critically engage with arguments regarding the trend of including programming and other computing contents into school mathematics education. I differentiate and compare two perspectives on this regard: computational thinking and computational literacy. Based on Niss' (1996) view on purposes of mathematics teaching, I delineate four aspects for consideration: political, contingent, disciplinary and empirical. My position is threefold. First, computational thinking is more accessible for implementation and assessment purposes. Second, computational literacy is a more radical view on the role of computing, beyond a tool for teaching traditional mathematics. Third, computational literacy has a more robust grounding, which may work as a necessary alternative to the state of the actual.

INTRODUCTION

Countries around the world have been including elements of programming and computing in their curricula (Bocconi et al., 2022) in the past decade. Both in curricula and in practice, mathematics tends to play a major role in these reforms, either by hosting the integrated computational skills or by sharing its role in providing a systematic and formal modelling approach to solving problems in other disciplines. This development is accompanied by a body of educational research that often takes this trend for granted, without a deep scrutiny of its premises. In this report, offer a critical discussion about the reasons and justifications (Niss, 1996) for the integration of computing in mathematics education from two major perspectives: computational thinking (Wing, 2006) and computational literacy (diSessa, 2018). The full version of this discussion has been submitted in Elicer (2025) in Spanish, and the purpose of this publication report is to extend these thoughts to a broader audience.

CLARIFICATION OF TERMS

Computational thinking

Revived by Wing (2006), the term *computational thinking* (CT) characterises the way computer scientists think and solve problems. Wing (2017) clarifies that CT describes the "mental activity involved in formulating problems that admit computational solution" (p. 8), suggesting a focus on the cognitive, human, and individual. It is, then, "a way that humans, not computers, think" (Wing, 2006, p. 35).

Other definitions that have emerged since then (e.g., Shute et al., 2017; Weintrop et al., 2016), describe computational thinking as a collection of skills. While seeking out for

Elicer, R. (2025). Between computational thinking and literacy in mathematics education. In C. Cornejo et al. (Eds.). *Proceedings of the 48th Conference of the International Group for the Psychology of Mathematics Education*, Vol. X, pp. XX-YY. PME.

a closed definition, Weintrop et al. (2016) ended up with a taxonomy of CT practices with four main categories: data, modelling and simulation, computational problem-solving, and systems thinking. Shute et al. (2017) define CT as "the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts" (p. 142). They then compartmentalise computational thinking into six categories: decomposition, abstraction, algorithm design, debugging, iteration, and generalization, which overlap with Weintrop et al.'s (2016).

CT is understood as a fundamental and not a niche skill. It should then be channelled into educational systems, as many countries are doing (Bocconi et al., 2022). Wing (2006) goes further to claim that a person educated in computer science "can go on to a career in medicine, law, business, politics, any type of science or engineering, and even the arts" (p. 35). CT is supposedly a competence for everyone and for everything.

Computational literacy

I draw on diSessa (2018) to characterise *computational literacy* (CL), whose work is grounded in the original Logo team (Papert, 1980; Clements & Sarama, 1997). On his terms, *literacy* should be understood as the "adoption by a broad cultural group—perhaps an entire civilization—of a particular form of infrastructural representational form for supporting intellectual activities" (diSessa, 2018, p. 4). This definition parallels written-word literacy in that reading and writing have been adopted by a large majority of humankind and it is by default the form of developing culture. diSessa distances himself from cognitive and pragmatic terms such as thinking, skills, and competences which can be planned, enacted, and measured individually. In contrast, literacy as understood by diSessa (2018) is a long-term, culturally broad issue, taking place in complex and diffuse social processes. CL is, then, the widespread adoption of computing as a form of representational infrastructure for certain intellectual activities, among others, those associated with mathematics and natural sciences.

Such an adoption is characterized by four foci or Rs (diSessa, 2018): remediation, reformulation, reorganization, and revitalization. *Remediation* implies a radical change in the media that support the representational infrastructure. The contents, order, and methods of teaching-learning would change dramatically due to a *reformulation* of intellectual resources. The intellectual terrain would be *reorganised*, so that using computers simply to "optimise" today's teaching would not bear fruit. The ecology of learning environments would be *revitalised*, facilitating the transition from isolated and impersonal exercises to projects aligned with the disciplines and the students' interests.

FRAMING REASONS AND ARGUMENTS

Niss (1996) defines *reasons* as the "driving force, typically of a general nature, that has actually motivated and brought about the existence (…) of mathematics education in a certain segment" (p. 12). Real reasons tend to be complex, vague, fussy and inertial. Only when explicitly activated, these reasons are called *arguments* or *justifications*, and they may refer to either the technological and socioeconomic development of a society; its political, ideological and cultural preservation and development; or the preparation of individuals with prerequisites for life.

Based on Niss (1996), I propose four aspects to consider when comparing CT and CL perspectives on justifications. At the societal level, I describe the explicit or declared *political* aspect, followed by the perhaps hidden *contingent* aspect that leads to curricular changes. Third, at the individual level, I unfold the *disciplinary* aspect of why and how computing and mathematics can go hand in hand. Fourth, I discuss the goodness of these latter justifications based on *empirical* research**.**

The political: A solution to what?

Special relevance has been attributed to the mathematics curriculum in societies with technocratic aspirations immersed in a predominantly capitalist world (Valero, 2023). It is therefore to be expected that arguments for including notions of computing will be similar to and competing with mathematics as a subject.

For example, we can look at the cases of England, Sweden and Denmark, investigated by Tamborg (2022). In England, the stated problem is the projection of a future labour market with high demand for computer skills. The response was the inclusion of a new *computing* subject with programming and basic software use content, and independent from mathematics. In Sweden, they observed an increasingly digitalised society, to which inclusive and democratic access must be guaranteed through a *digital competence*. Its skills and attitudes permeate existing subjects, such as programming and simulation in mathematics. In Denmark, there was a perceived need to respond to a digitalised and algorithm-dominated society, in which one must play an active role and have a critical stance. This is how the subject *technology comprehension* emerged. Its competence and subject-matter areas were juxtaposed with mathematics, leaving teachers the responsibility of this connection in the classroom (Tamborg et al., 2022).

If politics is considered in the foreground, as Valero (2023) suggests*,* their curricular reforms are historical documents that exemplify underlying ideologies. The three categories defined by Biesta (2009)—*qualification*, *socialisation* and *subjectivation*—can be helpful. England assumes a need for technical qualifications toward a future job. Sweden appeals to socialisation, in the sense of including students in a digitalised democracy. Denmark proposes a subjectivation of students, to produce active and critical types of people. That is to say, the lack of consensus on the way to include computer science notions is not in a vacuum, but rather responds to different political views on the purpose of education as national projects.

The contingent: Agendas and trends

The step between a declared political problem and the implementation of its claimed solution is not necessarily logical or evidence-based (Niss, 1996). Contingent trends and agendas are highly influential, particularly with digital technologies, whose development and insertion into classrooms precede rigorous studies on the matter.

There have been various initiatives to teach programming and coding outside the curriculum, by non-governmental organizations or private tech companies. One of them is vocational preparation (diSessa, 2018), the preparation for a future labour market with a high demand for programmers. This projection of the labour market is, at the very least, questionable, particularly given recent advances in generative artificial intelligence. More to the point, the teaching of programming has become predominant in niche sectors parallel to curricular planning (Williamson et al., 2019), therefore, without a basis thought out at the country level.

On the other hand, when there is centralised deliberation, the manifestation in curricular reforms can be rather chaotic. One of Tamborg's (2022) findings is that the wording of curricular reforms is, to some extent, contingent and random. For example, in Denmark, the strategy of integrating technology comprehension into existing subjects can be considered inorganic (Tamborg et al., 2022). A representative of the mathematics subject described the process as a "bingo where we had to agree which subject takes the different components of the curriculum" (Tamborg, 2022, p. 11). In Sweden, an expert indicated that programming was added to the algebra area because "they did not know where else to put it" (p. 15). The term *program* was replaced by *algorithm* in the text to make it more suitable for mathematics teachers, even though they are not equivalent concepts. Such decisions respond more to the urgency of carrying out a reform than to a solid collection of results from educational research.

The disciplinary: Blurring the disciplinary line

As scholarly disciplines, mathematics and computer science have notable overlaps, which explains why mathematics tends to be the host subject for the inclusion of computational content in the curriculum.

In teaching materials and practices, this integration can take many forms. For example, Elicer and Tamborg (2022) studied Danish tasks, and their analysis produced a model where what characterises a task is the weight attributed to the operational or conceptual aspects coming from each discipline. Bråting and Kilhamn (2022) used a similar distinction between mathematical and programming actions and concepts, resembling the *praxis* (know how) and *logos* (know why) of both subjects (Tamborg et al., 2022). They additionally found integrated tasks, where the disciplinary distinction is blurred.

A well-researched family of tasks where mathematics and computation organically come together is Turtle geometry (Papert, 1980; Clements & Sarama, 1997), named after the robotic turtle programmable by means of the Logo language. Papert (1980) proposed that, by programming a geometric figure, students are motivated—if not forced—to formalise their intuitions. For example, the task of writing code to draw any regular polygon is as much a computational challenge as a geometric one. Generalising the turning (external) angle and the polygon's internal angle go hand in hand. This is also a documented problem for students. While these angles are complementary and therefore interrelated, the external angle is difficult to identify as such even with visual support and, without proper intentionality, students tend to stick with trial-and-error strategies without a desire for generalisation or formalisation (Owens & Highfield, 2015). More generally, the study by Cui and Ng (2021) confirms that many of the difficulties associated with activities that for experts can be synergistic can be attributed to the fact of having to learn programming and mathematics together.

The empirical: What does the evidence say?

For the mathematics education community, fundamental questions are whether, how, and to what extent computation-related activities can support mathematics learning.

There is plenty of evidence of potential synergies between mathematics and computation in educational contexts. Some of them refer to specific areas of mathematics, and date back to the work with Logo. Papert (1980) argues that many mathematical contents make use of thought experiments that can be realised as computational experiments. This general idea is reinforced by Clements and Sarama (1997), who show evidence of how computational environments support or force the externalisation of intuitions and provide instant feedback, thus facilitating the formalisation of mathematical ideas. More recently, the literature review by Barcelos et al. (2018) revalidates the potentials of specific activities, highlighting the role of transfer between code, algebra and other representational media as a high-order skill.

The stated potentials tend to be channelled into the following claim: integrating computing improves mathematics learning. On a larger scale, the evidence does not yet support it. For example, one of the most influential recent research and development projects is *ScratchMaths*. This project focused on continuing the legacy of Logo through the Scratch block programming environment, with an explicit focus on mathematics in grades 1–6. They unfolded pedagogical possibilities for working with Scratch, based on their studies with a focus on teachers. However, they explicitly state that this project does not comprise a student-centred learning model Benton et al., 2017). Furthermore, the evaluation of this project determined that it managed to advance many learning objectives linked to computing, but did not produce a significant improvement in mathematics learning (Boylan et al., 2018).

Frameworks that link computing with mathematics outline its value in supporting mathematical problem solving (Kallia et al., 2021; Pérez, 2018). However, the studies by Cui and Ng (2021) and Elicer and Tamborg (2023) suggest that it is indeed the confluence of computational and mathematical learning that produces difficulties in effective problem solving. In a more recent study, Refvik and Opsal (2023) found no significant differences in mathematical problem-solving PISA items between the cohort of students who took elective programming courses and the control group.

DISCUSSION: THREE THESES

What do CT and CL perspectives bring about in terms of the existing arguments for integrating computational content in mathematics education?

First, pragmatically speaking, taking a CT perspective is more accessible. While the proliferation of theoretical frameworks can be confusing, much progress has been made in demystifying computational thinking as a collection of competencies, skills or practices (e.g., Shute et al., 2017; Weintrop et al., 2016). From a policy point of view, particularly from the perspective of education as a qualification (Biesta, 2009), these elements can be included in the curriculum in a transparent way, and their measurement and comparison between nations becomes feasible. Considering the disciplinary aspect, the reference frameworks mentioned have competencies directly associated with mathematics (Kallia et al., 2021; Pérez, 2018). This new wave of CT in research brings with it a vast collection of resources available to explore in the mathematics classroom. Projects such as ScratchMaths (Benton et al., 2017), new textbooks (Bråting & Kilhamn, 2022) and *Tekforsøget* (Elicer & Tamborg, 2022) are amongst them. Thus, from an empirical perspective, there is a point of departure for planning appropriate teaching activities that may foster the transfer between representations as a benefit of integrating computing into mathematics (Barcelos et al., 2018).

Second, CL, as understood by diSessa (2018), is much more radical than CT. CL anticipates a total *re-mediation* of intellectual activity in the future. Therefore, the inclusion of computing is an inevitable civilisation-wide process that transcends national political projects. In disciplinary terms, the issue is not to use computational applications to teach the same established mathematics curriculum as many projects aim for, but to *reformulate* the entire curriculum. According to Lodi and Martini (2021), this radicality is one of the reasons why Papert's (1980) ideas were not adopted, since the dominating agenda was exploiting technologies to teach what was already taught. This precaution would be validated by the more recent empirical research (Boylan et al., 2018; Cui & Ng, 2021; Elicer & Tamborg, 2023; Refvik & Opsal, 2023).

Third, CL has a more robust grounding than CT. From the political aspect, CL is not an agenda that seeks to adapt to a medium-term labour market as CT that may be uncertain, especially when current developments in artificial intelligence can offload much of the once-required programming skills. The contingencies that revived CT (Lodi & Martini, 2021) are bringing in research and resources that are already *revitalising* learning environments in the sense of CL. For example, the principles of ScratchMaths (Benton et al., 2017) promote the free exploration and exchange of ideas among students, and the legacy of computer science education brings with it the importance of play and investigation of students' own-posed problems (Elicer & Tamborg, 2023). CL also presents an alternative perspective that can explain and guide the disciplinary aspect and available empirical results. The works cited in this paper cast doubt on whether such computing have positive effects on mathematics learning and problem solving, as Wing (2006) would suggest for any discipline. A CL perspective questions this premise, and instead aims at *reorganising* the construction of knowledge. The case of Turtle geometry (computational) can illustrate that perhaps programming tends to be unproductive or confusing when the learning goal is

Euclidean geometry (logical), but it does succeed in developing visuospatial reasoning (Owens & Highfield, 2015).

I call the mathematics education community to take precautions before accepting overblown statements about CT and its role in "optimising" mathematical learning or supporting any other discipline. More importantly, such claims have been made about mathematics, and we know better than that.

References

Barcelos, T. S., Munoz, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics learning through computational thinking activities: A systematic literature review. *Journal of Universal Computer Science*, *24*(7), 815–845.

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, *3*(2), 115–138.

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Jasutė, E., Malagoli, C., Masiulionytė-Dagiė, V., & Stupurienė, G. (2022). *Reviewing computational thinking in compulsory education: State of play and practices from computing education*. Publications Office of the European Union.

Boylan, M., Demack, S., Wolstenholme, C., Reidy, J., & Reaney, S. (2018). *ScratchMaths: Evaluation report and executive summary. Project Report*. Education Endownment Foundation.

Bråting, K., & Kilhamn, C. (2022). The integration of programming in Swedish school mathematics: Investigating elementary mathematics textbooks. *Scandinavian Journal of Educational Research*, *66*(4), 594–609.

Clements, D. H., & Sarama, J. (1997). Research on Logo: A decade of progress. *Computers in the Schools*, *14*(1–2), 9–46.

Cui, Z., & Ng, O.-L. (2021). The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment. *Journal of Educational Computing Research*, *59*(5), 988–1012.

diSessa, A. A. (2018). Computational literacy and "the big picture" concerning computers in mathematics education. *Mathematical Thinking and Learning*, *20*(1), 3–31.

Elicer, R., & Tamborg, A. L. (2022). Nature of the relations between programming and computational thinking and mathematics in Danish teaching resources. In U. T. Jankvist, R. Elicer, A. Clark-Wilson, H.-G. Weigand, & M. Thomsen (Eds.), *Proc. of the 15th Int. Conference on Technology in Mathematics Teaching* (pp. 45–52). Aarhus University.

Elicer, R. & Tamborg, A. L. (2023). A critical case study on the implementation of computational thinking in mathematics education. *Implementation and Replication Studies in Mathematics Education*, *3*(1), pp. 44–74.

Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education*, *23*(2), 159–187.

Kilhamn, C., Bråting, K., Helenius, O., & Mason, J. (2022). Variables in early algebra: Exploring didactic potentials in programming activities. *ZDM*, *54*(6), 1273–1288.

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, *30*(4), 883–908.

Niss, M. (1996). Goals of mathematics teaching. In A. Bishop, M. A. (Ken) Clements, C. Keitel-Kreidt, J. Kilpatrick, & C. Laborde (Eds.), *International Handbook of Mathematics Education* (pp. 11–47). Kluwer.

Owens, K., & Highfield, K. (2015). Visuospatial reasoning in contexts with digital technology. In K. Owens, *Visuospatial reasoning: An ecocultural perspective for space geometry and measurement education* (pp. 275–289). Springer.

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

Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education*, *49*(4), 424–461.

Refvik, K. A. S., & Opsal, H. (2023). Do optional programming courses affect eighth-grade students' mathematical problem solving? *Computers in the Schools*, *40*(3), pp. 244–261.

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

Tamborg, A. L. (2022). A solution to what? Aims and means of implementing informatics-related subjects in Denmark, Sweden, and England. *Acta Didactica Norden*, *16*(4), 1–26.

Tamborg, A. L., Elicer, R., Misfeldt, M., & Jankvist, U. T. (2022). Computational thinking in Denmark from an anthropological theory of the didactic perspective. In C. Fernández, S. Llinares, A. Gutiérrez, & N. Planas (Eds.), *Proc. of the 45th Conf. of the Int. Group for the Psychology of Mathematics Education* (Vol. 4, pp. 91–98). PME.

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.

Williamson, B., Bergviken Rensfeldt, A., Player-Koro, C., & Selwyn, N. (2019). Education recoded: Policy mobilities in the international 'learning to code' agenda. *Journal of Education Policy*, *34*(5), 705–725.

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

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7–14.

Valero, P. (2023). A cultural–political reading of school mathematics curriculum reform. En Y. Shimizu & R. Vithal (Eds.), *Mathematics Curriculum Reforms Around the World, New ICMI Study Series* (pp. 545–548).

This publication report corresponds to the following submitted article:

Elicer, R. (2025). *Entre el pensamiento y la alfabetización computacionales [Between computational thinking and literacy]* [Manuscript submitted for publication]. Department of Sustainability and Planning, Aalborg University.