Chapter 23
Emergence of Computational/Algorithmic Thinking and Its Impact on the Mathematics Curriculum

Djordje M. Kadijevich, Max Stephens, and Abolfazl Rafiepour

The first ever ICMI study, undertaken in 1985, was entitled *The influence of computers and informatics on mathematics and its teaching* (Churchhouse et al., 1986). The contributing authors, mainly mathematicians operated by and large from a European and North American perspective, mostly focused on using computers to model some advanced mathematical ideas. (Papers that focused on teaching and learning were published in a separate supplementary publication.) Despite that, there was the emergence of an international perspective, however limited, on the relevance of computers and informatics to the teaching and learning of mathematics, particularly in the mathematics curriculum of the senior high school that could just make use of spreadsheets and some graphing packages. This was because high powered computer software programs were realised few years later: the first Wolfram Mathematica in 1988; graphing calculators with Computer Algebra Systems (CAS) in the late 1990s.

In 2006, the ICMI Study 17, hosted in Vietnam, returned to the same theme as "Technology Revisited" (Hoyles & Lagrange, 2010). The contributing authors were mostly from education, and, understandably, there was a stronger focus of the papers was on the teaching and learning. This ICMI study adopted a broader

D. M. Kadijevich (✉)
Institute for Educational Research, Belgrade, Serbia
e-mail: djkadijevic@ipi.ac.rs

M. Stephens
University of Melbourne, Melbourne, Australia
e-mail: m.stephens@unimelb.edu.au

A. Rafiepour
Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran
e-mail: drafiepour@gmail.com

Y. Shimizu, R. Vithal (eds.), *Mathematics Curriculum Reforms Around the World*, New ICMI Study Series, https://doi.org/10.1007/978-3-031-13548-4_23

international perspective, but its conclusions were still cautiously stated. They concluded that some national governments had moved ahead, but generally the position was described as one of limited adoption of technology in the teaching and learning of mathematics, with differences occurring even within different states of the same country. Despite a broader international participation in the study, one can say that the issue had not moved outside the concerns of those mathematics educators who remained its chief protagonists, while other mathematicians and mathematics educators appeared less convinced that the use of digital technologies had an important and indispensable role in school mathematics.

Professional and Societal Perspectives: Connections Between Internationalisation and Globalisation

The so-called fourth industrial revolution, based upon widespread use of data analytics including big data, rapid refinements in artificial intelligence and its applications, and near universal access to high-speed Internet supporting cloud storages, has created economic and social conditions that require an increasing supply of ICT skilled workers. Up to 40 million new positions would need to be filled globally by digitally competent workers, creating an urgent need for young people to leave schools and training institutions digitally literate (WEF, 2018).

As a result, the international debate about curricular issues no longer takes place largely within an educational/academic community. By 2018, major international forums and agencies have taken up these issues. Economic 'think tanks', such as the World Economic Forum and inter-governmental agencies such as the Asia-Pacific Economic Cooperation (APEC, 2018), have advocated strongly that educational systems and school mathematics need to respond promptly to the digital revolution. To this end, a prize for promoting the use of digital technology in education is offered by the Islamic Educational, Scientific and Cultural Organisation (ICESCO, 2019), for example.

These interventions are directly relevant to our theme and help us to draw out important connections between internationalisation and globalisation; in particular, showing how global economic and social conditions influence the framing of agendas for internationalisation of the school mathematics curriculum. ICMI Study 1 and ICMI Study 17 are instances of internationalisation, trying to bring together different perspectives from the participating countries and seeking to reach a measure of agreement about what should take place. However, participation in these studies was limited largely to educators and mathematicians, with a still muted role for governments and international agencies.

To understand economic and social conditions that require an increasing supply of ICT skilled workers and to prepare to take part in the fourth industrial revolution in their future work, students need to learn to apply computational thinking (i.e. thinking based on computations) skilfully. The global trends presented above,

connecting professional and societal perspectives, are, for example, evident in two directions regarding the rationales for including this thinking in compulsory education: one deals with enabling students to solve problems as an information-processing agent, whereas a second concerns a greatly needed preparation of qualified workforce (Bocconi et al., 2016; Rafiepour, 2018). Clearly, there is no hard and fast separation between these two rationales, each obviously influencing each other.

The emergence of computational thinking is as a clear instance of globalisation and internalisation in education as a consequence of: (1) an increasing reliance on digital technology, whose applications often combines local and global contexts; (2) growing use of algorithmic techniques to deal with various real-world challenges, many of which go beyond local contexts; (3) raised parental and societal expectations concerning a better education of children, involving out-of-school coding and programming activities that are available globally. These issues are considered in the preceding chapter by Stephens, Kadijevich, Niss, Azrou and Namikawa.

Using Technology in Mathematics Education: Computational and Algorithmic Thinking

Technology has been integrated in mathematics curriculum in many countries worldwide. About 90% of countries that participated in TIMSS 2015, for example, reported initiatives for this integration (Mullis et al., 2016). However, there is lack of a solid knowledge of the way in which the integration could affect the content taught and enhance its teaching and learning (Cai & Howson, 2012). Not only have questions (such as, "Would frequent use of computers increase achievement?" and "Is the quality of computers use more important to learning outcomes than the quantity of computers use?") generated inconsistent answers, but also several findings supported just an infrequent use of computers in the classroom (Kadijevich, 2015).

To contribute to the development of this knowledge, research may primarily focus on the way in which the use of computers and other digital tools can mediate the learning of mathematics in a productive way (Drijvers, 2018). Because the mediation in question is based upon problem solving with technology (computers and other digital tools), apart from mathematical reasoning, it involves the above-mentioned computational thinking, i.e. thinking based on computations, often related to the application of tools and techniques from computer science.

The term *computational thinking* (CT) was first used by Papert (1980) in his book, *Mindstorms: Children, computers, and powerful ideas* to describe specific thinking that children applied in learning mathematics (i.e. Turtle geometry) through LOGO programming. CT was later examined by Wing (2006), who viewed it as a fundamental personal ability like reading, writing, and arithmetic. The Royal Society (RS, 2011) described this ability as enabling persons to recognise aspects of

computations in various problem situations, and to deal with those aspects, by applying tools and techniques from computer science.

Algorithmic thinking (AT), on the other hand, is one form of mathematical reasoning, required whenever one has to comprehend, test, improve, or design an algorithm – "a precisely described routine procedure that can be applied and systematically followed through to a conclusion" (The Concise Oxford Dictionary of Mathematics, 4th edn, p. 11). This procedure, whereby a mathematical problem is usually solved, processes some numeric, symbolic or geometric data. To deal with algorithms successfully, AT calls for distinct cognitive abilities, including abstraction (making general statements summarising particular examples) and decomposition (breaking a problem down into sub-problems).

CT deals with solutions in representations that could be efficiently processed by information-processing agents (Wing, 2011). As these agents are mostly computers nowadays, we assume that it is precisely the application of automation that separate AT from CT. However, mathematicians may prefer to use term AT even when computer tools are used (see Lockwood et al., 2016, for this preference).

Chapter Outline

In the rest of this chapter – based upon Kadijevich (2019b), Stephens and Kadijevich (2020) and Rafiepour (2018) – CT, as a broader notion, is examined first in detail, by summarising research findings regarding defining, cultivating and assessing it. This examination is followed by a section on CT/AT, discussing different educational priorities and practices regarding CT/AT, its relevance to mathematics education, and emerging implications for this education. The chapter ends with a summary of the findings presented and suggests directions for further research.

Computational Thinking

Despite its widespread use, a widely accepted definition of CT is lacking (Mouza et al., 2017). It has been defined in terms of its main facets, dimensions, concepts, practices, perspectives, etc. For example, core CT facets may be abstraction (data collection and analysis, pattern recognition, modelling), decomposition, algorithms (algorithm design, parallelism, efficiency, automation), iteration, debugging and generalisation (Shute et al., 2017). As regards CT dimensions, there may be three: its concepts (e.g. data, operators, loops), its practices (e.g. abstracting, modularising, debugging), and its perspectives (e.g. questioning, connecting) (Brennan & Resnick, 2012; cf. Kafai & Burke, 2013). In a high school STEM context, CT may comprise four categories of practices, namely: data practices (e.g. collecting, visualising), modelling and simulation practices (e.g. building and using computational models), computational problem-solving practices (e.g. programming,

troubleshooting) and system-thinking practices (e.g. defining systems, managing complexity) (Weintrop et al., 2016).

To simplify matters in defining CT, we may just focus on its basic steps or stages used in problem solving, such as decomposition, pattern recognition, abstraction, and algorithmic thinking, recognised by Hoyles and Noss (2015) as main thinking skills required. These stages, as equally important, may be considered as CT cornerstones (see Fig. 23.1).

However, the processes of abstraction and pattern recognition overlap because pattern recognition may be viewed as abstraction and generalisation (Scantamburlo, 2014). In addition, pattern recognition may be an overall goal of CT, like in troubleshooting or managing system complexity. Finally, the use of technology to automate solutions is missing in this four-step model. It may be thus better to assume that basic CT steps (stages) are decomposition, abstraction, algorithmisation and automation, which may be advanced in a complex, nonlinear way by going back and forth between (not only neighbouring) stages (Fig. 23.2). In a preliminary empirical study, Kadijevich (2019a) shows that this cycle is not only relevant to different subject areas, such as mathematics and science, but also relevant to distinctive learning

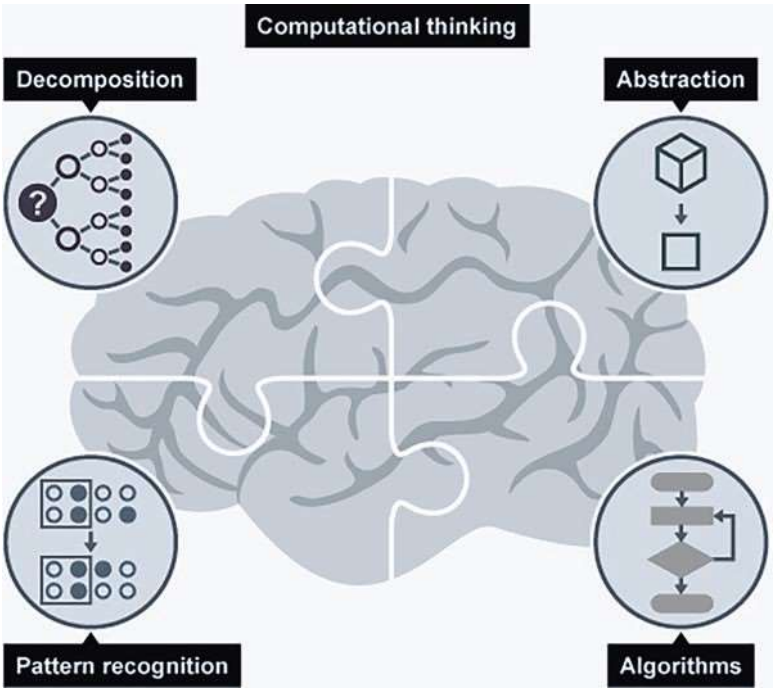

Fig. 23.1 Four cornerstones of CT. (Source: https://www.bbc.com/bitesize/guides/zp92mp3/revision/1)

tasks commonly given in these subject areas, such as data visualisation and spreadsheet modelling.

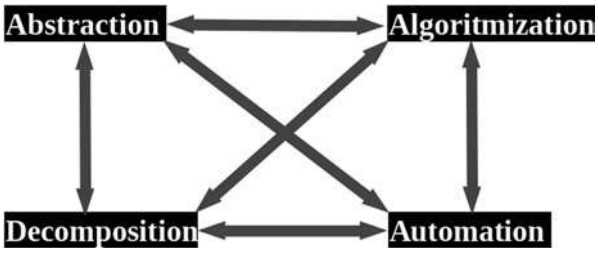

Fig. 23.2 CT cycle. (Kadijevich, 2018a, p. 76)

Cultivating and Assessing CT

Because research is scarce, knowledge about the integration of CT in K–12 education is limited (Voogt et al., 2015). However, to cultivate such thinking, rich computational environments should be used, and students encouraged to develop digital artefacts; in these environments by progressing along a use-modify-create learning path (Lee et al., 2011). Less experienced or novice students should be encouraged to progress along an understand-debug-extend trajectory, i.e. from understanding developed 'artefact' via debugging this 'artefact' to extending it (see Brennan & Resnick, 2012). For an appropriate integration, Mouza and colleagues (2017) discourage narrow use of digital tools (e.g. just concepts mapping tools) promoting just one or two CT components (e.g. problem decomposition).

The learning paths mentioned above may be recognised in CT pedagogy proposed by Kotsopoulos and colleagues (2017), which assumes that the various conceptual or digital objects in mathematical classes make use of four overlapping activities: unplugging (not using computers); tinkering (taking objects apart and changing/modifying their components); making (constructing new objects); remixing (appropriating of objects or their components to produce new objects). As examples of these activities, consider, respectively, sorting mathematical expressions, modifying spreadsheet content, developing interactive geometry presentation, and combining and modifying existing interactive reports to visualise data with dashboard (a set of interactive reports).

A lack of standard CT definition has resulted in diverse measurement of this construct, making comparing results of research studies difficult. Furthermore, CT assessment in classrooms is challenging, requiring real-time assessments that monitor students' progress (Shute et al., 2017). Such assessments could be based on the analysis of students' project portfolios regarding 'artefacts' they develop through progressing along a learning path (e.g. Brennan & Resnick, 2012; Lee et al., 2011), possibly resulted from the application of a suitable pedagogical framework (Kotsopoulos et al., 2017). This analysis should focus on CT features (e.g. stages or components) and their relations aimed to be promoted. To assess CT-based instruction, a technology integration rubric may be used, whose criteria evaluate choosing and applying digital tools and CT components respecting curriculum goals and

instructional strategies, simultaneously aligning content, pedagogy, and technology (Mouza et al., 2017).

Implications for Algorithmic Thinking (AT)

By accepting the position that the main stages of a learning cycle describing AT are decomposition, abstraction and algorithmisation, it may be said that, as indicated above, CT occurs whenever AT is supported by automation, i.e. the use of computational tools and environments. This means that approaches to cultivating and assessing CT summarised above may be applied to cultivating and assessing AT. For example, a suitable learning path could use the following trajectory: from understanding developed algorithm via debugging this algorithm to extending or improving it, focusing of AT features (e.g. stages or components) and their relations aimed to be promoted. Furthermore, to assess AT-based instruction, a AT integration rubric may be used, whose criteria evaluate choosing and promoting AT components with respect to curriculum goals and instructional strategies, simultaneously aligning content and pedagogy.

Computational/Algorithmic Thinking

While the above arguments present a case for a clear distinction between CT and AT, their use in practice reflects different interests and priorities. The term 'computational thinking' rightly draws attention to the underlying logical and mathematical processes that are fundamental to computer science and should be contrasted with facility or familiarity in using digital machines. These processes have been introduced to students in the New Zealand program *computer science unplugged* (https://protect-au.mimecast.com/s/SToXCJypvAfqwq6QRhYchXg?domain=csunplugged.org), to give a well-known international example.[1]

Instead of using the term 'computational thinking', recent educational documents in UK (Stephens, 2018) and Argentina (Sadosky Foundation, 2018) mostly use words 'algorithms' and 'algorithmic'. Algorithms and algorithmic thinking are the preferred terms in the *Australian Curriculum: Digital Technologies* (ACARA, 2016) across all years of schooling. A priority to algorithms is given in the French curriculum, *Algorithmique et Programmation* – a domain of both the mathematics and the technology curricula, and thus taught by the teachers of these two disciplines (Ministere de l'Education Nationale, 2016) where *Scratch* is the main

[1] If the reader accepts the definition of CT assumed in this chapter, the *csunplugged* approach, which does not rely on the use of computers i.e. automation, may be viewed as means that primarily promotes AT.

programming language. Later, in high school, algorithmics is also taught in mathematics using *Python*.

In other countries, curriculum documents emphasise coding and programming in basic education. In Finland, for example, there is a clear emphasis on the cross curricular uses of programming, including the use of programming languages, with specific attention to computer-less programming in the early years (PMO, 2019). Similarly, the announcement by the Japanese government to introduce programming in primary and secondary schools from 2020 has a clear focus on programming across the curriculum. In fact, the published materials refer repeatedly to 'programming thinking' as distinct from learning to program a machine (Stephens, 2018).

Relevance to Mathematics Education

The term 'computational thinking' has been used extensively by computer science specialists, who carried out many studies that link CT and computer science topics, mostly programming (e.g. Hickmott et al., 2018). Consequently, CT has become a critical curricular component in computer science (informatics) education (e.g. Webb et al., 2017). It has not had a similar status in mathematics education. The reason may be that studies explicitly linking it and learning mathematics are rather rare (Hickmott et al., 2018), mostly dealing with areas that are traditionally connected to programming (e.g. numbers and operations, algebra, geometry).

In mathematics education, the main task of technology is to mediate the learning of mathematics in productive ways (Drijvers, 2018), including relating procedural and conceptual mathematical knowledge (Kadijevich, 2018b). AT may be critical to developing these knowledge types and relating them. For example, procedural knowledge may be developed through implementing procedures, especially through designing procedures and algorithms, which could result in knowledge that is rich in connections (e.g. Lockwood et al., 2016). On the other hand, AT may be used to develop conceptual knowledge and a deeper conceptual understanding if a special case of a formula, or an algorithm in general, is used as a means for asking advanced questions about the result obtained by applying it (Abramovich, 2015). Research has supported the position that, in digital environments techniques could be used as a means to relate procedures and concepts (e.g. Artigue, 2010).

If CT/AT is to have an enlarged role in the mathematics curriculum, we must continue to ask how these forms of thinking build upon, connect with, and enhance the way students think about and do mathematics. The following examples are intended to make it clear that our emphasis is on *mathematical thinking* – not on following or memorising routines, and still less on equating algorithmic thinking with coding. Productive examples might include: using the language of algorithms to exemplify mathematical concepts and procedures (e.g. starting with multiplication and division); drawing on appropriate mathematical knowledge to construct algorithms (e.g. to model a particular problem and to allow for a solution); using the

language of mathematics to explain the key steps of a given algorithm (e.g. a simulation); using the language of mathematics to identify or improve the variables and parameters required to use a given algorithm (e.g. in data practices); critically examining solutions to improve on an existing algorithm; identifying mathematical variables and parameters in order to use a given algorithm (e.g. in data analysis); using an algorithmic application to solve a mathematical problem in order to identify its mathematical structure and to generalise the solutions (e.g. computational problem solving); using an algorithm to deepen understanding of mathematical patterns and relationships.

Emerging Implications for Mathematics Education

In the remainder of this chapter, we tend to use the combined term CT/AT, mindful that some readers may be accustomed to separate uses of these terms. As yet there appears to be no international consensus in these matters. We refer to an increasing trend for CT/AT to be included in the compulsory years (basic education) for all students (Stephens, 2018). CT/AT has the potential to play an important role in problem solving and modelling in the school mathematics curriculum at all stages, where, for example, iteratively developed (deterministic or probabilistic) solutions can be expressed in forms resulted from the application of CT/AT (e.g. a spreadsheet model that determines the profitability of a small business; Kadijevich, 2012). In STEM contexts, CT/AT can develop a synergy between mathematical modelling, computer programming, and engineering design (López-Leiva et al., 2019). The use of CT/AT in STEM contexts should also be considered from the primary school years.

Data analysis, based upon the use of interactive displays for example (Kadijevich, 2019b) is a simple instance of data science, defined as the science of obtaining useful information from data by using various computational methods and tools. Data science reflects the unprecedented growth in the availability of data in most areas of human activity. CT/AT is an essential support for steps in data science learning cycle, such as ask/frame questions, locate/accumulate/evaluate data, analyse data, and interpret data (Gould et al., 2017). Data science latter is an emerging and important area of statistics education, supporting students to acquire data and to use them to make informed decisions in their daily lives (see, for example, *International Data Science in Schools Project* (IDSSP) at: http://www.idssp.org/).

Professional mathematicians apply computation in their disciplinary practice to support various aspects of their work, involving experimentation, approximation, conjecture testing, and visualisation. These areas are now increasingly recognised as important features of mathematical reasoning within school mathematics. Although classroom practice should be different from disciplinary practice, the latter should inform the former and help designing it (Lockwood et al., 2019). Students may use CT/AT to define (construct) objects, identify their possible properties (of algebraic, geometric, or statistical nature) and verify these properties (a number of studies reported in Hoyles & Lagrange, 2010, for example, may be re-examined in

that way). The identification and application of geometric properties of shapes, for example, underpins the application of CT/AT in computer design and art, and allows these potentialities to be explored in two and three dimensions much earlier than traditional school geometry has allowed. Like mathematicians who apply computation to find approximate solutions to intractable problems, students may use CT/AT to approximate solutions of mathematical models that cannot (easily) be solved in the context of school mathematics (for examples of such problems, see Kenderov, 2018).

Conclusion

CT originated from learning mathematics with technology. While CT is critical curricular component in computer science (informatics) education (e.g. Webb et al., 2017), CT lacks a similar status in mathematics education. Apart from areas that are traditionally connected to programming (e.g. numbers and operations, algebra, geometry), further research, including curriculum development, is needed to explore other areas of mathematics suitable for technology supported problem solving, such as functions, probability, and statistics explored through modelling, simulations, and data analysis, respectively (Hickmott et al., 2018). Such exemplars of problem-solving utilising CT/AT should aim at developing and interconnecting procedural and conceptual mathematical knowledge.

CT/AT has changed the nature of some contemporary researches in mathematics domains. These are now recognised internationally. For example, computer-based proofs (e.g. four colour theorems) are now accepted in mathematics. New domains of research related to mathematics and computation have become possible, such as Bioinformatics. In this regard, the European Mathematical Society (EMS, 2011) recognised an emerging way of engaging in mathematical research: "Together with theory and experimentation, a third pillar of scientific inquiry of complex systems has emerged in the form of a combination of modelling, simulation, optimization and visualization" (p. 2). Weintrop et al. (2016) try to address CT/AT as a sophisticated and overarching concept through a literature review and interviews with experts who use CT/AT in their professional lives. Accordingly, they have developed a taxonomy of CT/AT which bears a close relation with the third pillar of scientific inquiry mentioned above. Their taxonomy contains four main categories: data practices, modelling and simulation practices, computational problem-solving practices, and systems thinking practices. For each category, they explain how contemporary activities used by mathematicians and scientists are related to CT/AT, arguing that these four areas can be viewed as a future "roadmap for what CT instruction should include in the classroom" (p. 128). For school education, however, we may well need smaller mini road-maps showing how students are introduced to and are led to explore each of these areas. These road maps will be needed to guide the next stages to embed CT/AT in the school mathematics curriculum.

Although AT and CT may in mathematics education denote similar entities, mathematics educators may prefer to use (privilege) the former term to distinguish its place in the school mathematics curriculum from components of the computer science or digital technologies curriculum. The place of algorithms in mathematics has a long history long before the use of computers. Whatever one calls this thinking – computational, algorithmic, programming or even computational algorithmic thinking – the emphasis in mathematics education should be placed on *mathematical thinking* supported by suitable technology. Incorporating CT/AT in the school mathematics curriculum will require important decisions to be taken by national and local curriculum agencies. These will vary from country to country and are outlined in the following chapter. International cooperation and sharing among researchers and educators will be vital, taking special care about defining CT/AT precisely, cultivating this thinking accordingly with a focus on mathematical reasoning, and assessing the contribution of CT/AT to this reasoning appropriately.

Acknowledgement The research done by the first author was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia (contract no. 451-03-68/2022-14/200018).

References

Abramovich, S. (2015). Mathematical problem posing as a link between algorithmic thinking and conceptual knowledge. *The Teaching of Mathematics, 18*(2), 45–60. http://elib.mi.sanu.ac.rs/files/journals/tm/35/tmn35p45-60.pdf

ACARA. (2016). *Digital technologies*. Australian Curriculum, Assessment and Reporting Authority. http://docs.acara.edu.au/resources/Digital_Technologies_-_Sequence_of_content.pdf

APEC. (2018). *Project DARE (data analytics raising employment)*. Asia-Pacific Economic Cooperation. https://www.apec.org/Press/News-Releases/2018/1109_dare

Artigue, M. (2010). The future of teaching and learning mathematics with digital technologies. In C. Hoyles & J.-B. Lagrange (Eds.), *Mathematics education and technology: Rethinking the terrain. The 17th ICMI study* (pp. 463–476). Springer.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education*. European Union, European Commission, Joint Research Centre.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*. AERA. https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Cai, J., & Howson, A. (2012). Toward an international mathematics curriculum. In K. Clements, A. Bishop, C. Keitel, J. Kilpatrick, & F. Leung (Eds.), *Third international handbook of mathematics education* (pp. 949–974). Springer.

Churchhouse, R., Cornu, B., Howson, A., Kahane, J.-P., van Lint, J., Pluvinage, F., Ralston, A., & Yamaguti, M. (Eds.). (1986). *The influence of computers and informatics on mathematics and its teaching*. Cambridge University Press.

Drijvers, P. (2018). Tools and taxonomies: A response to Hoyles. *Research in Mathematics Education, 20*(3), 229–235.

EMS. (2011). *Position paper on the European Commission's contributions to European research*. European Mathematical Society.

Gould, R., Bargagliotti, A., & Johnson, T. (2017). An analysis of secondary teachers' reasoning with participatory sensing data. *Statistics Education Research Journal, 16*(2), 305–334.

Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education, 4*(1), 48–69.

Hoyles, C., & Lagrange, J.-B. (Eds.). (2010). *Mathematics education and technology: Rethinking the terrain. The 17th ICMI study*. Springer.

Hoyles, C., & Noss, R. (2015). Revisiting programming to enhance mathematics learning. In *Paper presented at math + coding symposium*. University of Western Ontario.

ICESCO. (2019). *Announcement of second edition prize for open digital educational resources*. Islamic Educational, Scientific and Cultural Organization. https://www.icesco.org/en/2019/04/22/announcement-of-the-2nd-edition-of-the-isesco-prize-for-open-digital-educational-resources/

Kadijevich, D. (2012). Examining errors in simple spreadsheet modeling from different research perspectives. *Journal of Educational Computing Research, 47*(2), 137–153.

Kadijevich, D. (2015). A dataset from TIMSS to examine the relationship between computer use and mathematics achievement. *British Journal of Educational Technology, 46*(5), 984–987.

Kadijevich, D. (2018a). A cycle of computational thinking. In B. Trebinjac & S. Jovanović (Eds.), *Proceedings of the 9th international conference on e-learning* (pp. 75–77). Metropolitan University. https://protect-au.mimecast.com/s/78-iCMwvygsq3qw6ghYjdaj?domain=elearning.metropolitan.ac.rs

Kadijevich, D. (2018b). Relating procedural and conceptual knowledge. *The Teaching of Mathematics, 21*(1), 15–28. http://elib.mi.sanu.ac.rs/files/journals/tm/40/tmn40p15-28.pdf

Kadijevich, D. (2019a). A cycle of computational thinking and its relevance: An empirical study. In S. Jovanović & B. Trebinjac (Eds.), *Proceedings of the 10th conference on e-learning* (pp. 136–138). Metropolitan University. https://protect-au.mimecast.com/s/fXP7CNLwzjF030OMAFgiZ1m?domain=metropolitan.ac.rs

Kadijevich, D. (2019b). Cultivating computational thinking through data practice. In D. Passey, R. Bottino, C. Lewin, & E. Sanchez (Eds.), *Empowering learners for life in the digital age* (pp. 24–33). Springer.

Kafai, Y., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan, 95*(1), 61–65.

Kenderov, P. (2018). Powering knowledge versus pouring facts. In G. Kaiser, H. Forgasz, M. Graven, A. Kuzniak, E. Simmt, & B. Xu (Eds.), *Invited lectures from the 13th International Congress on Mathematical Education* (pp. 289–306). Springer.

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education, 3*(2), 154–171.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 33–37.

Lockwood, E., DeJarnette, A., Asay, A., & Thomas, M. (2016). Algorithmic thinking: An initial characterization of computational thinking in mathematics. In M. Wood, E. Turner, M. Civil, & J. Eli (Eds.), *Proceedings of the 38th annual meeting of the orth American chapter of the International Group for the Psychology of Mathematics Education* (pp. 1588–1595). PME-NA.

Lockwood, E., DeJarnette, A., & Thomas, M. (2019). Computing as a mathematical disciplinary practice. *Journal of Mathematical Behavior, 54*, 100688.

López-Leiva, C., Pattichis, M., & Celedón-Pattichis, S. (2019). Modelling and programming of digital video: A source for the integration of mathematics, engineering, and technology. In B. Doig, J. Williams, D. Swanson, R. B. Ferri, & P. Drake (Eds.), *Interdisciplinary mathematics education. ICME-13 monographs* (pp. 135–153). Springer.

MEN. (2016). *Algorithmique et programmation*. Ministère de l'Éducation Nationale. http://cache.media.eduscol.education.fr/file/Algorithmique_et_programmation/67/9/RA16_C4_MATH_algorithmique_et_programmation_N.D_551679.pdf

Mouza, C., Yang, H., Pan, Y.-C., Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology, 33*(3), 61–76.

Mullis, I., Martin, M., & Loveless, T. (2016). *20 years of TIMSS: International trends in mathematics and science achievement, curriculum, and instruction*. TIMSS & PIRLS International Study Center, Boston College.

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

PMO. (2019). *Comprehensive schools in the digital age*. Prime Minister's Office.

Rafiepour, A. (2018). Iran school mathematics curriculum: Past, present and future. In Y. Shimizu & R. Vithal (Eds.), *School mathematics curriculum reforms: Challenges, changes and opportunities. Proceedings of the twenty-fourth ICM1 study conference* (pp. 467–474). International Commission on Mathematical Instruction.

RS. (2011). *Shut down or restart? The way forward for computing in UK schools*. The Royal Society. https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

Sadosky Foundation. (2018). *Ciencias De La Computacion: Para El Aula – Manual para docentes*. Author. http://program.ar/descargas/cc_para_el_aula-2do_ciclo_primaria.pdf

Scantamburlo, T. (2014). *Philosophical aspects in pattern recognition research*. Unpublished doctoral dissertation. Venice, Italy: Department of informatics, University Ca' Foscari of Venice. http://dspace.unive.it/bitstream/handle/10579/4639/phdthesis_TeresaScantamburlo.pdf?sequence=1

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

Stephens, M. (2018). Embedding algorithmic thinking more clearly in the mathematics curriculum. In Y. Shimizu & R. Vithal (Eds.), *School mathematics curriculum reforms: Challenges, changes and opportunities. Proceedings of the twenty-fourth ICM1 study conference* (pp. 483–490). International Commission on Mathematical Instruction.

Stephens, M., & Kadijevich, D. (2020). Computational/algorithmic thinking. In S. Lerman (Ed.), *Encyclopedia of mathematics education* (pp. 117–123). Springer.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715–728.

Webb, M., Davis, N., Bell, T., Katz, Y., Reynolds, N., Chambers, D., & Sysło, M. (2017). Computer science in K–12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies, 22*(2), 445–468.

WEF. (2018). *The future of jobs report*. Centre for the New Economy and Society, World Economic Forum. https://www3.weforum.org/docs/WEF_Future_of_Jobs_2018.pdf

Weintrop, D., Beheshti, E., Horn, M., Orno, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classroom. *Journal of Science Education and Technology, 25*(1), 127–141.

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

Wing, J. (2011). Research notebook: Computational thinking – What and why? *The Link Newsletter, 6*, 1–32. https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf