Education Inquiry

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/zedu20

A review of research on teaching of computer programming in primary school mathematics: moving towards sustainable classroom action

O.E. Holo, E.N. Kveim, M.S. Lysne, L.H. Taraldsen & F.O. Haara

To cite this article: O.E. Holo, E.N. Kveim, M.S. Lysne, L.H. Taraldsen & F.O. Haara (2022): A review of research on teaching of computer programming in primary school mathematics: moving towards sustainable classroom action, Education Inquiry, DOI: 10.1080/20004508.2022.2072575

To link to this article: https://doi.org/10.1080/20004508.2022.2072575

Published online: 13 May 2022.

EDUCATION INQUIRY
https://doi.org/10.1080/20004508.2022.2072575

Routledge
Taylor & Francis Group

REVIEW ARTICLE

A review of research on teaching of computer programming in primary school mathematics: moving towards sustainable classroom action

O.E. Holo, E.N. Kveim, M.S. Lysne, L.H. Taraldsen and F.O. Haara

Faculty of Education, Arts and Sports, Western Norway University of Applied Sciences, Postboks 133, 6851 Sogndal, Norway

ABSTRACT
This review article focuses on the teaching of computer programming in primary school mathematics, as reported by 37 empirical studies. Our findings show that a variety of research methods are used, which often combine qualitative and quantitative approaches. Furthermore, we found the focus of attention to vary from emphasis on the teacher and the teacher's knowledge regarding computer programming, to how use of tools and devices can influence learning outcomes, and the actual teaching and learning in the classroom. We found the implications for teaching congregate in three categories: the relation between mathematics and computer programming, tools and devices, and didactical approaches. Based on the discussion of our findings, we suggest two new possibilities for research on the teaching of computer programming in primary school mathematics: focus on computer programming and specific mathematical topics, and focus on students' in-depth learning in a way that makes it possible to explore connections between computer programming and mathematics and use of computational thinking to solve mathematical problems.

KEYWORDS
computer programming; mathematics; primary school; teaching; classroom action

1 Introduction

Since the 1980s, computer science and computer programming have been a part of school curricula throughout Europe, but its emphasis has come in waves (Benton, Hoyles, Kalas, & Noss, 2017). To some extent, the responsibility for educating digitally literate students has been left to especially interested teachers (Yadav, Gretter, Hambrusch, & Sands, 2016). This is now changing, leaving teachers insecure when faced with new situations and demands (Sentance & Csizmadia, 2017; Sentance & Waite, 2018). Influenced by stakeholders such as the European Commission, and through developmental programmes such as Shaping Europe's Digital Future and research programmes such as Horizon 2020, computer programming is now included in revisions of school curricula, albeit in various ways. Some countries have chosen to implement a separate subject on computer programming (e.g. the UK and Poland), some work with it as an interdisciplinary competence (e.g. Iceland and Portugal), others

CONTACT O.E. Holo odd.eivind.holo@hvl.no Faculty of Education, Arts and Sports, Western Norway University of Applied Sciences, Postboks 133, 6851 Sogndal, Norway

integrate it into the existing school subject of mathematics (e.g. Spain and France) (Balanskat & Engelhardt, 2015; Kaufmann & Stenseth, 2020). In the Nordic context, Denmark has tried out both treating it as a separate subject, and as integrated in school subjects (Børne- Og Undervisningsministeriet, 2021), while Norway, Sweden, and Finland have all made computer programming part of the mathematics curriculum in their revised or proposed school curricula (Bocconi, Chioccariello, & Earp, 2018; Opetus- Ja Kulttuuriministeriö, 2018; Skolverket, 2019; Utdanningsdirektoratet, 2020).

In 2013, a committee appointed by the Norwegian Ministry of Local Government and Modernisation stated that there had been too much focus in school on learning how to use computers for communicating, writing texts and other humanistic aspects, and too little emphasis on algorithms, mathematics and technology (Kommunal- Og Moderniseringsdepartementet, 2013). It stated that, to ensure growth in digital value creation, it is necessary to educate future generations to be able to create, and not just consume (Kommunal- Og Moderniseringsdepartementet, 2013). This point of view is supported both by research (Fortunati, Manganelli, & Ferrin, 2020) and various stakeholders, who argue that the society needs more emphasis on learning and innovation skills, in-depth approaches to learning, complex problem-solving skills, and mastery of core subjects such as mathematics and science, from its citizens in the future (e.g. Battelle for Kids, 2015; National Research Council, 2012; World Economic Forum, 2016).

Computer programming is defined from various angles in the literature. Some sources define it using a historical approach in an educational context (e.g. Benton et al., 2017; Waite, Curzon, Marsh, & Sentance, 2020), while some loosely define it as synonymous with coding (e.g. Benton, Saunders, Kalas, Hoyles, & Noss, 2018; Mason & Rich, 2020). Others imply that computer programming is something more than what coding is, as coding is understood as the actual of putting together computer codes, and see it as an essential part of computing and computer science (e.g. Arfé, Vardanega, & Ronconi, 2020; Kalelioğlu, 2015). One definition, which we employ in this article, is that computer programming is the creation and coding of steps that allow a problem situation to be solved by a computer (Israel, Pearson, Tapia, Wherfel, & Reese, 2015). This fits well with the common understanding of computer programming as an approach that may foster computational thinking skills (e.g. Dağ, 2019; Piedade, Dorotea, Pedro, & Matos, 2020; Ríos Félix, Zatarain Cabada, & Barrón Estrada, 2020), and more precisely, skills related to computational thinking, such as problem solving, critical thinking, creative thinking, algorithmic thinking, and reflective thinking (Durak, Yilmaz, & Yilmaz, 2019; Erümit, 2020; Fokides, 2017; Kong & Wang, 2019). Wing (2006): 35) suggested that computational thinking is "conceptualizing, not programming. Computer science is not computer programming. Thinking like a computer scientist means more than being able to program a computer". Various models of computational thinking have been added since (e.g. Brennan & Resnick, 2012; Weintrop et al., 2016), but it remains that computational thinking is a set of cognitive skills that is essential to a computer scientist and is very useful in a variety of logical disciplines, such as mathematics. In Norway, for instance, the Ministry of Education has decided to use this understanding of computational thinking as basis for the curricular understanding. This is further used as a motivation for how and why computer programming should be part of the mathematics curriculum (Utdanningsdirektoratet, 2022). This understanding is based on the Barefoot Computing Organisation's interpretation and illustration of concepts and approaches relevant for the computational thinker (Barefoot, 2022). The connection between computer programming and mathematics is shown through the attention to problem solving and algorithmic thinking within both areas, which are core elements in mathematics and emphasised in mathematics classrooms all over the world.

An important insight from research is that curricular integration of computer programming with mathematics is a complex and challenging endeavour for both teachers and students (e.g. Dickes, Farris, & Sengupta, 2020; Humble, Mozelius, & Sällvin, 2020; Rich et al., 2019; Yadav et al., 2016). There are a number of reasons for this, including teachers' lack of technical skills in computer programming (e.g. Dağ, 2019; Kaufmann & Stenseth, 2020), uncertainty about how to integrate computer programming and mathematics (Stigberg & Stigberg, 2020; Vinnervik, 2020), and the difference in opinion regarding whether computer science should be considered a foundational discipline instead of a subject in mathematics (Nardelli, 2019). Nevertheless, computer programming has now become part of the mathematics curriculum in many countries. Students are therefore entitled to quality teaching of the relations between computer programming and mathematics, to meet central expectations of in-depth learning, development of computational thinking skills, and use of computer programming in exploring and solving problems. In Norway, this focus is expected to begin in first grade. The teacher will find competence aims related to algorithmic thinking at the second-, third-, and fourth-grade levels, and competence aims specifically related to computer programming from the fifth-grade level (Utdanningsdirektoratet, 2020). This early attention to computer programming in school is supported by research (Arfé et al., 2020; Fokides, 2017; Hassenfeld, Govind, De Ruiter, & Umashi Bers, 2020), and in the years to come, we expect that a kind of common attention to teaching of computer programming in primary school mathematics will become visible in schools. In review articles focusing on computer programming and mathematics education, like for instance (Forsström & Kaufmann, 2018) and (Hickmott, Prieto-Rodriguez, & Holmes, 2018), we find a general attention to the connection of these two areas in school. With a more concentrated focus in this review article, we aim to contribute to an increase in the targeted development of contributions based on research that will scaffold primary school mathematics teachers in the relocation from paying attention to computer programming, to sustainable classroom action. We do this by focusing on areas of research interest in peer-reviewed articles based on empirical projects where use of computer programming is associated with the teaching of primary school mathematics. Given these conditions, we focus on the following three questions:

1. What methods have so far been used to examine teaching of computer programming in primary school mathematics?
2. What is the focus of attention in research on the teaching of computer programming in primary school mathematics?
3. What are the implications for the teaching of computer programming in primary school mathematics?

2 Method

2.1 *Review parameters*

To answer our research questions, we conducted a literature review with a systematic search and selection of articles written in English. The selection process consisted of three phases. In the first phase, we tested several terms connected with computer programming, such as “computational thinking”, “programming”, and “coding”, and combined them with “mathematics education” using search engines, such as Oria and Google. Through this inductive, preliminary phase, we obtained an overview of possible and relevant keywords to use in our formal search. We decided to use the following keywords: “teach* programming”, “mathematics”, and “primary school” or “elementary school”.

In the second phase, we entered a rigid and deductive search stage. The keywords found in the first phase were used to search the ERIC, Scopus, ScienceDirect, ISI Web of Science, and Springer Link databases. These five databases were chosen for the broad influence they have within education research. The searches were limited to peer-reviewed articles written in English and published between January 2013 and September 2020 (the searches were completed at the beginning of October 2020). We decided to leave out articles published before 2013 for two reasons: on the international level, the implementation of a greater focus on computer programming in curricula gained momentum throughout Europe from 2013 (Balanskat & Engelhardt, 2015); and on the Nordic level, a 2013 Norwegian government report argued for the necessity of moving away from a user perspective on information and communications technology skills, and move towards development of understanding, creation and the progress of such technology (Kommunal- Og Moderniseringsdepartementet, 2013).

In the third phase, maintaining a rigid and deductive approach, we excluded articles that did not include empirical studies, did not focus on primary school/elementary school students or teachers/pre-service teachers as informants, did not mention “mathematics”, or were not published in peer-reviewed journals. Then, we removed any articles that, after closer inspection, did not meet the selection criteria we decided on in the first phase. This included articles that focused on special education or did not pay attention to mathematics in school. After completion of the three phases, 37 articles remained to form the basis of this review. Based on the description of the three phases of the selection process for the articles in this review, the search should be easy to reproduce. The requirement for articles to be written in English, be based on empirical studies, focus on educational purposes, and be published in peer-review journals, provides assurance that the articles in this review represent the worldwide research publications on the use of computer programming with a focus on the teaching of primary school mathematics.

2.2 *Analysis*

We started the process of analysis by dividing the articles between all five authors, for a first round of reading. We used a summary table for each article, in which we each outlined setting (number and type of informants), method and data sources, the focus of attention in the article, and implications for teaching. The first author then combined these details into a single table (our protocol of analysis) and checked for inconsistencies or missing information. To strengthen the reliability and validity of the analysis of articles, the second author then went through all 37 articles and the protocol to check that the presented information reflected the content of each article. When the second author's inspection identified inconsistencies, minor adjustments were made by agreement between the first and second authors. The revision stage did not require any major adjustments.

The three research questions defined the criteria for analysis. For each article, we identified the methods used for data collection and the type of informants. Further, the focus of attention in each article was identified and described according to recurrent topics. Another criterion for analysis was therefore to identify the main question(s) asked by the researchers and determine what main fields of attention the current research could be grouped in. Third, each article provides implications of presented results, discussion, and conclusions on some level. Because of this, a third criterion for analysis was to identify from each article the experienced opportunities and challenges related to the teaching of computer programming in primary school mathematics. Through this we would gain an overall impression of the research's progress and the implications of its contributions to date. This allowed us to outline implications that were offered by the reviewed articles, both insights from the interventions and consequences this have for teaching computer programming in primary school mathematics. This analytical approach is reflected in the structure of the presentation of results in this article.

3 Results

3.1 *Methods used in the reviewed studies*

The studies included in this review employed a variety of qualitative and quantitative approaches to answer their research questions. Studies defined their design as an intervention study (Arfé et al., 2020; Chevalier, Giang, Piatti, & Mondada, 2020), case study (Benton et al., 2018; Çakiroğlu et al., 2018), or design research (e.g. Benton et al., 2017; Carlborg, Tyrén, Heath, & Eriksson, 2019). However, they used a variety of methods for data collection within these frameworks. An overview of the types of methods used in this research is provided in Table 1.

Table 1. Types of methods.

Type of method	Number of occurrences
Interview or conversationa	18
Assessment	15
Questionnaire and/or survey	14
Observationb	12
Student work or artefact	8
Written reflections	3
Feedback	1

*Structured and semi-structured interviews, focus group interview, group conversation, and informal conversation.
**Observation, video recordings, and field notes from researchers.

Most studies reviewed here combined two or more methods for data collection. This was often a combination of quantitative and qualitative methods, in which the participants' perspectives were expressed. Also in quantitative surveys, where a scale is used, it was often the participants' report of their own conceptual and procedural knowledge and skills, as well as their confidence, that was expressed (e.g. Durak et al., 2019; Kong, Lai, & Sun, 2020). It is noticeable that interviews or conversations with participants were the most common methods, which highlights the importance of the participants' perspective. Also when only one method was employed, we found several examples of qualitative methods, such as interview (e.g. Kucuk & Sisman, 2018; Yadav et al., 2016), group conversation (Vinnervik, 2020), or written reflection (Humble et al., 2020). Observation was also a much-used method in this research material (e.g. Kim et al., 2015; Stigberg & Stigberg, 2020), which indicates a large focus on process. This can also be seen in the collection and analysis of student work as a customary method (e.g. Benton et al., 2017; Fokides, 2017). Student work can also form the basis for assessment, which was the second most common method (e.g. Kaplan, Pavlov, & Myradov, 2020; Yallihep & Kutlu, 2020). The focus on assessing learning outcome shows that several articles focused on the effect of an intervention or a new approach.

The studies included in this review were limited to empirical studies involving primary school students and/or primary school teachers as participants or informants. This allowed for considering both the student and teacher perspectives when answering the research questions posed in this article. Fifteen of the studies focused solely on students (e.g. Çakiroğlu et al., 2018; Lambić, Đorić, & Ivakić, 2020), while 10 studies involved both students and their teachers (e.g. Dickes et al., 2020; Mladenović, Boljat, & Žanko, 2018). It is interesting to note that most of these studies revolved around upper primary school level (approximately grades 5–7 or ages 10–13), while only seven articles focused on the first years of schooling. This may suggest that research on computer programming in the first years of primary school is not a large concern, even though some studies suggest that starting early will enhance students' understanding of the concept and subsequent development of computational thinking (Arfé et al., 2020; Fokides, 2017; Hassenfeld et al., 2020). Twelve studies included pre-service and in-service teachers in primary school. Two of the studies including pre-service teachers had participants with a particular interest in computer science (Dağ, 2019; Piedade et al., 2020), while three studies focused on pre-service teachers with no previous knowledge of computer programming (Kim et al., 2015; Kucuk & Sisman, 2018; Uzumcu & Bay, 2020). The studies involving in-service teachers were aimed both at teachers who wanted to teach computer programming (e.g. Kong et al., 2020; Vinnervik, 2020), and teachers who were already teaching computer programming (e.g. Kalelioglu & Sentance, 2020; Yadav et al., 2016).

3.2 *Focus of attention*

The 37 articles that form the basis for this review represent several research topics but based on their focus of attention they can be clustered into three major areas of interest.

First, some research projects were concerned with the teacher and the teacher's knowledge, skills, and attitudes regarding computer programming, and how to positively influence such aspects (e.g. Humble et al., 2020; Pérez-Marin, Hijon-Neira, &

Martin-Lope, 2018). We see this as a natural starting point for research on the teaching of computer programming because we understand that it is difficult for the teacher to plan and conduct quality teaching if the content to be taught and the meaning of it are insufficient.

Second, it is important to note that several research projects were concerned with the choice of tools and devices. Some investigated use of tools such as Scratch (e.g. Fidai, Capraro, & Capraro, 2020; Kalelioğlu & Gülbahar, 2014), while others examined the use of devices such as micro:bit or robots (e.g. Kucuk & Sisman, 2018; Noh & Lee, 2020). These projects had similar aims; they wanted to investigate how teachers could use tools or devices in their teaching to enhance the students' learning of computer programming (Çakiroğlu et al., 2018; Carlborg et al., 2019; Kalelioglu & Sentance, 2020; Psycharis & Kotzampasaki, 2019; Ríos Félix et al., 2020).

Third, we see that research is concerned with teaching and learning programming in the primary school mathematics classroom, although representing two perspectives: Effects and classroom activities. In some projects it is apparent that the concern is with impact and measuring the effects of teaching computer programming. (Iskrenovic-Momcilovic, 2019) examined how to organise the students' way of working to increase their learning outcome, and (Fokides, 2017) examined how the role of the teacher affected the students' learning outcome. The relation to learning outcome is also shown in articles that attend to major educational areas, such as motivation and self-efficacy (Dağ, 2019; Durak et al., 2019), or major mathematics education area, such as development of problem-solving skills (Carlborg et al., 2019), excellence of various computer programming tools and devices (Kalelioglu & Sentance, 2020), and development of algorithmic thinking (Erümit, 2020).

Within the classroom activity perspective, the surveyed research focuses on investigation and observation of teachers' and students' classroom activity. Pérez-Marin et al.'s (2018) investigation into the language metaphors teachers may use with young students when introducing computer programming showed that computer programming can be identified as cooking, and program-sequences as recipes. (Stigberg & Stigberg, 2020) investigated how young students describe the relationship between programming and mathematics, and (Mladenović et al., 2018) examined whether teachers can make didactical choices in primary school education to prevent students' misconceptions regarding algorithmic loops. (Chevalier et al., 2020) worked on how to structure the problem-solving process so that the students could postpone the computer programming process and avoid the "trial and error" phase for as long as possible. (Dickes et al., 2020) studied how primary school students and teachers can work together to investigate how computer programming can be applied to the content of primary school mathematics. They showed that through exploiting teaching approaches that are well-known within problem solving and socio-cultural learning environments, the teachers and students can develop computer programming skills together by reframing computer programming as mathematisation, in this particular case through designing measures of change. In fact, (Dickes et al., 2020) demonstrated an explicit connection between a specific mathematical topic and computer programming. Such a relationship is also apparent in a few other articles. For instance, (Benton et al., 2018) saw the students' computer programming and development of algorithms in relation to place value. However, we found that most articles that connected computer programming to teaching and learning activities by identifying a relation to mathematics did this through broader concepts such as logic, computational thinking, problem-solving, and reflective thinking (e.g. Durak et al., 2019; Erümit, 2020; Piedade et al., 2020).

3.3 *Implications for teaching*

In this article, we have surveyed the methods used and the focus of attention in research, showing that a variety of quantitative and qualitative approaches are applied, which address the teaching of computer programming in primary school mathematics. Based on these impressions, one might perceive the research field to have little structure and no specific aim, but in our opinion, quite the opposite seems to be the case. We see all this research as valuable for reaching a higher goal in primary school mathematics: to make the teaching of computer programming a manageable and relevant part of such teaching. Based on the retention of such a goal within a research field that is currently rather complex, combined with necessary attention to life in school, we classify the implications of the current research into three areas: *relations between mathematics and computer programming, the importance of tools and devices*, and *didactical approaches*.

According to (Durak et al., 2019) and (Erümit, 2020), use of computer programming, and the subsequent exploring of mathematics with the help of computer programming, may help students improve their algorithmic thinking, problem-solving skills, and reflective thinking. One of the participants in (Kalelioğlu, 2015) research even stated that "programming helped create a love of mathematics" (p. 207). On the other hand, it would not be very productive at this stage to go to the research to find exact relations between computer programming and specific mathematical topics. Our review shows that research does not give much priority to this area. So far, it seems that the area of school mathematics is applied mainly as a context for learning computer programming. The broad attention to learning within specific mathematical topics and to using computer programming to explore mathematical content has yet to come.

Although various software is being trialled, block programming offered by Scratch has so far proved to be a kind of highway to the introduction of computer programming in primary schools (Benton, Kalas, Saunders, Hoyles, & Noss, 2018; Çakiroğlu et al., 2018; Durak et al., 2019; Erümit, 2020; Iskrenovic-Momcilovic, 2019; Rich et al., 2019). We note that some devices, such as micro:bit, have been given more attention than others (Carlborg et al., 2019; Kalelioglu & Sentance, 2020). Scratch and micro:bit are examples of tools and devices that have already been chosen as main entries in initial attempts to implement computer programming in primary school in some contexts, for instance in Norway and the UK (BBC, 2015; Vitensenterforeningen, n.d.). Support for these choices can be found in articles that are part of this review (Humble et al., 2020; Mladenović et al., 2018). In addition, (Kong & Wang, 2019) asserted that the use of devices may enhance young students' interest in computer programming, and that this may be explained by the visual effect, for example, the effect that a robot gives when executing an algorithm. An important extension of such an observation is that (Noh & Lee, 2020) found that the use of robots significantly improved the students' computational thinking and creativity. On a more general level, (Erümit, 2020) asserted that no matter how powerful a computer programming tool is, it is not the tool that influences the students, but how it is used. Using a computer programming tool does not in itself make the students see the connection to mathematical ideas or develop problem solving skills; it is necessary for a competent teacher to select suitable tasks and to scaffold the students during their learning processes (Benton et al., 2018).

The above reservations from (Erümit, 2020) and (Benton et al., 2018) may offer an impression of the implications from research on what to do in the classroom to be somewhat divergent. This is strengthened by (Kalelioglu & Sentance, 2020) claim that there is no such thing as a most powerful form of representation of subject knowledge. Instead, they encourage teachers to use a variety of representations and scaffolding methods in the teaching of computer programming. Such vague implications do not offer much guidance about the role of the teacher or about how to teach computer programming in primary school mathematics. However, several of the reviewed articles try in some way to develop a pedagogical or didactical framework for teaching and learning related to computer programming (e.g. Benton et al., 2017; Chevalier et al., 2020; Dağ, 2019; Kaplan et al., 2020). The main implications so far seem to be that teachers have profited from participating in computer programming courses (e.g. Kim et al., 2015; Kong et al., 2020; Kucuk & Sisman, 2018), and that teachers need to be supported with a solid infrastructure that scaffolds their acquisition of computer programming knowledge (Stigberg & Stigberg, 2020), for instance through participatory designed teacher workshops (Haara, Engelsen, & Smith, 2020), and daily practice support.

Research studies also show us that there is no need to wait for the students to reach upper primary school before they start to work with computer programming, although there is a lack of consensus in terms of what aspects to give priority to in the early stages of introduction. Some suggest that unplugged programming is appropriate as a general approach to computer programming in the lower grades, regardless of subject (Humble et al., 2020; Uzumcu & Bay, 2020), while others recommend that the "plugged" programming in school starts as early as possible (Fokides, 2017; Kalelioğlu, 2015). On a more concrete level, our review also reveals that there are viable ways for the teacher to start in the classroom. Several research articles in this review investigated how the students' resources may be emphasised in the teaching. (Mladenović et al., 2018) and Pérez-Marin et al. (2018) have both done research projects on how to avoid challenges in learning and how to develop programming algorithms together with upper primary school students, the former related to block programming (Scratch), and the latter to development of familiar language to be used as metaphors for computer programming concepts. Iskrenovic-Momcilovic (2019) claimed that pair programming on individual computers has a positive effect on students' learning, while Dağ (2019) and Kucuk and Sisman (2018) concluded that teachers should connect computer programming to examples from daily life and include real-life problems to make computer programming more relevant and interesting for students. Dickes et al. (2020) and Ríos Félix et al. (2020) encouraged teachers to orchestrate the investigation of what computer programming is, as a joint exploration between the teacher and the students. Finally, Durak et al. (2019) reported that students without any prior knowledge of computer programming are more open-minded and confident about taking on computer programming challenges than students with some experience in computer programming. This is a familiar pattern within knowledge development, because more knowledge also makes one more aware of obstacles and possible pitfalls. In any case, the power of student resources and student enthusiasm reported in the research may be something to pursue in the introductory teaching of computer programming.

4 Discussion and conclusions

In the introduction of this article, we Highlighted the expectations teachers must meet when implementing computer programming in school, particularly in Nordic regions, where it is introduced as part of school mathematics. We acknowledge the urgency of this implementation, and the widespread attention to making teaching of computer programming a manageable and relevant part of teaching in primary school mathematics, when the decision to make computer programming part of school mathematics (in fact) has been made. For this to happen efficiently in school, however, some issues need to be addressed.

4.1 *State of affairs*

We see that a wide variety of research approaches and methods have been used, and that the research community is not limited in its approaches, questions, or investigations regarding research on teaching computer programming in primary school mathematics. This is promising, because we do not yet know much about this field and there are many possibilities to consider and investigate, both for the researcher and for the teacher. So far, we have learned that various tools and devices are being trialled, the necessity of teacher competence has been highlighted, and there is focus on how to teach computer programming in primary school. This is important, because schools and teachers need to be given the opportunity to critically choose what to do in their teaching, and which tools and devices they want to use. These are fundamental areas that need to be addressed, and although such processes have begun, they are far from complete. We therefore assume that this type of research will continue with strength.

We have not yet found much research on computer programming being directly linked to learning within specific mathematical topics. This might be because, until now, the research community has been focusing on the more fundamental areas highlighted above, instead of mapping the appropriate mathematical content for linking with computer programming. The direct link between computer programming and learning within specific mathematical topics is an area of research that will probably develop once the common attention to teaching computer programming related to school mathematics has started to materialise through classroom action. On the other hand, we note that more general concepts related to mathematics – such as algorithmic thinking, problem solving, and computational skills – have already been highlighted. This is a crucial link between mathematics and computer programming, as shown by the argument for computer programming to be related to the future need to educate citizens who can create digitally and solve problems with help from computers (Fortunati et al., 2020; Kommunal- Og Moderniseringsdepartementet, 2013). This aligns in general with the attention to 21st century skills in school, and specifically with the expectation of using computer programming in school mathematics to explore mathematical connections and problems. This is a challenging endeavour that both schools and the research community have yet to investigate, and which may be an important contribution to school research on in-depth learning in mathematics. Dağ (2019) and Kucuk and Sisman (2018) concluded that teachers ought to connect computer programming to examples from daily life and real-life problems. In this way, computer programming can be considered as a real option and a tool that may be used to explore and solve problems.

4.2 *Recommendations for future research*

Schools, teachers, and students need the research community to continue its work on teaching computer programming in primary school mathematics. One aim for future research should be to help teachers fulfil the expectations from stakeholders who call for students' improved digital competence. Another aim should be to investigate possibilities and challenges related to learning mathematics through exploration, assisted by computational thinking skills. In our opinion, the research community is on the right track, but it is necessary to increase the emphasis on intervention studies and empirical classroom studies, in collaboration with teachers. The articles by Dickes et al. (2020) and Ríos Félix et al. (2020) are examples of studies where the teacher and students are active participants. Such studies will simultaneously bring the research on teaching computer programming in primary school mathematics forward and scaffold mathematics teachers first-hand in the shift from paying attention to computer programming, to sustainable classroom action. In our experience, this kind of research is rare within computer programming, and would mean increased emphasis on qualitative research, based on studies of best-practice and design-based research and action research projects involving teachers who develop teaching arrangements and are responsible for mathematics teaching. At the same time, it is important to be aware that a vast collection of teaching arrangements is now being developed at schools to meet the expectations of implementing computer programming in the primary school mathematics classroom. One example is the development of an algorithm together with first-grade students, where the teacher has to carry out the algorithm commands from the students to move through a trail from the classroom door to get hold of a "joint biscuit treasure" (lots of fun and observations of the power of algorithms for both students and teacher, but yet undocumented by the involved teacher). Another example is Taraldsen and Myhra's (2019) report on their work with the relationships between length, time, and velocity using Sphero balls together with a fifth-grade teacher and students. Such development projects ought to give rich possibilities for empirically based research, with emphasis on the learning environment, learning outcome, and teacher role. On the Norwegian and Danish venue, for instance, the national implementation of micro:bit (Danmarks radio, n.d.; Vitensenterforeningen, n.d) should provide such an opportunity.

We suggest two major new possibilities related to the content of research on teaching computer programming in primary school mathematics, which also are two crucial areas for mathematics teachers and students in the classrooms where computer programming is expected to be part of learning mathematics. First, we need research that focuses on the relation between computer programming and specific mathematical topics. Second, we need research that focuses on students' in-depth learning in a way that makes it possible to explore connections between computer programming and mathematics and use of computational thinking to solve mathematical problems. In our opinion, the time is ripe for a broad, qualitative research investigation of these two areas, to help teachers understand and respond accurately and confidently to questions of why it is considered appropriate to make computer programming part of the mathematics curriculum.

The research community on teaching computer programming in primary school mathematics has made a promising start, but we have only just begun working our way from paying attention to computer programming, to sustainable classroom action. We believe that the continuation of this journey will be easier for teachers and primary school students through the joint use of research and development forces. This will prioritise research projects that involve primary school mathematics teachers who are curious about how they can integrate computer programming and specific mathematical topics, are already developing ideas about how this can be done, and are willing to make this available for researchers. The future possibilities for research on teaching computer programming in primary school mathematics are abundant, despite the fact that there is no algorithm to follow.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Odd-Eivind Holo Assistant professor in mathematics education Faculty of Education, Arts and Sports, Western Norway University of Applied Sciences, Sogndal, Norway.

Eskil Nore Kveim Assistant professor in STEM education ViteMeir, Science Center of Sogn og Fjordane.

Mari Skjerdal Lysne Assistant professor in English education Faculty of Education, Arts and Sports, Western Norway University of Applied Sciences, Sogndal, Norway.

Lene Hayden Taraldsen Assistant professor in mathematics education Faculty of Education, Arts and Sports, Western Norway University of Applied Sciences, Sogndal, Norway.

Frode Olav Haara Associate professor in mathematics education Faculty of Education, Arts and Sports, Western Norway University of Applied Sciences, Sogndal, Norway.

References

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, *148*, Article 103807. doi:10.1016/j.compedu.2020.103807.

Balanskat, A., & Engelhardt, K. (2015, October). *Computing our future. Computer programming and coding: Priorities, school curricula and initiatives across Europe. Update 2015*. Brussels, Belgium: European Schoolnet. http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03

Barefoot, (2022). *Computational thinking focus in the Barefoot project*. https://barefootcas.org.uk/

Battelle for Kids. (2015). *Partnership for 21st century learning: 21st century student outcomes*. http://www.p21.org/our-work/p21-framework
BBC. (2015, July 6). *BBC and partners unveil the landmark BBC micro:bit*. https://www.bbc.co.uk/mediacentre/mediapacks/microbit
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. doi:10.1007/s40751-017-0028-x.
Benton, L., Kalas, I., Saunders, P., Hoyles, C., & Noss, R. (2018). Beyond jam sandwiches and cups of tea: An exploration of primary pupils' algorithm-evaluation strategies. *Journal of Computer Assisted Learning*, *34*(5), 590–601. doi:10.1111/jcal.12266.
Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International Journal of Child-Computer Interaction*, *16*, 68–76. doi:10.1016/j.ijcci.2017.12.004.
Bocconi, S., Chioccariello, A., & Earp, J. (2018, January). *The Nordic approach to introducing computational thinking and programming in compulsory education*. European Schoolnet. https://www.itd.cnr.it/doc/CompuThinkNordic.pdf
Børne- Og Undervisningsministeriet. (2021). *Erfaringerne fra forsøg med teknologiforståelse i folkeskolen er landet* [*Experiences from piloting understanding of technology in compulsory school*]. https://www.uvm.dk/aktuelt/nyheder/uvm/2021/okt/211004-erfaringerne-fra-forsoeg-med-teknologiforstaaelse-i-folkeskolen-er-landet
Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada (Vol. 1, p. 25).
Çakiroğlu, Ü., Suiçmez, S. S., Kurtoğlu, Y. B., Sari, A., Yildiz, S., & Öztürk, M. (2018). Exploring perceived cognitive load in learning programming via Scratch. *Research in Learning Technology* Article 1888, *26*. doi:10.25304/rlt.v26.1888
Carlborg, N., Tyrén, M., Heath, C., & Eriksson, E. (2019). The scope of autonomy when teaching computational thinking in primary school. *International Journal of Child-Computer Interaction*, *21*, 130–139. doi:10.1016/j.ijcci.2019.06.005.
Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education*, *7*(1), Article 39. doi:10.1186/s40594-020-00238-z
Dağ, F. (2019). Prepare pre-service teachers to teach computer programming skills at K-12 level: Experiences in a course. *Journal of Computers in Education*, *6*(2), 277–313. doi:10.1007/s40692-019-00137-5.
Danmarks radio. (n.d.). *ultra:bit*. https://www.dr.dk/skole/ultrabit
Dickes, A. C., Farris, A. V., & Sengupta, P. (2020). Sociomathematical norms for integrating coding and modeling with elementary science: A dialogical approach. *Journal of Science Education and Technology*, *29*(1), 35–52. doi:10.1007/s10956-019-09795-7.
Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, *10*(2), 173–197. doi:10.30935/cet.554493.
Erümit, A. K. (2020). Effects of different teaching approaches on programming skills. *Education and Information Technologies*, *25*(2), 1013–1037. doi:10.1007/s10639-019-10010-8.
Fidai, A., Capraro, M. M., & Capraro, R. M. (2020). "Scratch"-ing computational thinking with Arduino: A meta-analysis. *Thinking Skills and Creativity*, *38*, 100726. doi:10.1016/j.tsc.2020.100726. Article 100726
Fokides, E. (2017). Students learning to program by developing games: Results of a year-long project in primary school settings. *Journal of Information Technology Education: Research*, *16*, 475–505. doi:10.28945/3893.
Forsström, S. E., & Kaufmann, O. T. (2018). A literature review exploring the use of programming in mathematics education. *International Journal of Learning, Teaching and Educational Research*, *17*(12), 18–32. doi:10.26803/ijlter.17.12.2.

Fortunati, L., Manganelli, A. M., & Ferrin, G. (2020). Arts and crafts robots or LEGO® MINDSTORMS robots? A comparative study in educational robotics. *International Journal of Technology and Design Education*. Advance online publication. doi:10.1007/s10798-020-09609-7.

Haara, F. O., Engelsen, K. S., & Smith, K. (2020). Moving from traditional to responsive mathematics classrooms: A proposition of an intervention model. *Teacher Development*, *24*(3), 399–414. doi:10.1080/13664530.2020.1763443.

Hassenfeld, Z. R., Govind, M., De Ruiter, L. E., & Umashi Bers, M. (2020). If you can program, you can write: Learning introductory programming across literacy levels. *Journal of Information Technology Education: Research*, *19*, 65–85. doi:10.28945/4509.

Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping of studies on computational thinking in K-12 mathematics classrooms. *Mathematics and Programming*, *4*, 48–69. doi:10.1007/s40751-017-0038-8.

Humble, N., Mozelius, P., & Sällvin, L. (2020). Remaking and reinforcing mathematics and technology with programming – Teacher perceptions of challenges, opportunities and tools in K-12 settings. *International Journal of Information and Learning Technology*, *37*(5), 309–321. doi:10.1108/IJILT-02-2020-0021.

Iskrenovic-Momcilovic, O. (2019). Pair programming with scratch. *Education and Information Technologies*, *24*(5), 2943–2952. doi:10.1007/s10639-019-09905-3.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*, 263–279. doi:10.1016/j.compedu.2014.11.022.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, *52*, 200–210. doi:10.1016/j.chb.2015.05.047.

Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, *13*(1), 33–50.

Kalelioglu, F., & Sentance, S. (2020). Teaching with physical computing in school: The case of the micro:bit. *Education and Information Technologies*, *25*(4), 2577–2603. doi:10.1007/s10639-019-10080-8.

Kaplan, A., Pavlov, D., & Myradov, M. (2020). Features of using KODU game lab in teaching programming. *Mathematics and Informatics*, *63*(1), 9–23.

Kaufmann, O. T., & Stenseth, B. (2020). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*. Advance online publication. 10.1080/0020739X.2020.1736349.

Kim, C., Kim, D., Yuan, J., Hill, R. B., Doshi, P., & Thai, C. N. (2015). Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching. *Computers & Education*, *91*, 14–31. doi:10.1016/j.compedu.2015.08.005.

Kommunal- og moderniseringsdepartementet. (2013) *NOU 2013: 2. Hindre for digital verdiskapning* [*NOU 2013: 2. Obstacles to digital value creation*]. https://www.regjeringen.no/contentassets/e2f0d5676e144305967f21011b715c16/no/pdfs/nou201320130002000dddpd

Kong, S.-C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, *151*, Article 103872. doi:10.1016/j.compedu.2020.103872

Kong, S.-C., & Wang, Y.-Q. (2019). Nurture interest-driven creators in programmable robotics education: An empirical investigation in primary school settings. *Research and Practice in Technology Enhanced Learning*, *14*(1), Article 20. doi:10.1186/s41039-019-0116-1

Kucuk, S., & Sisman, B. (2018). Pre-service teachers' experiences in learning robotics design and programming. *Informatics in Education*, *17*(2), 301–320. doi:10.15388/infedu.2018.16.

Lambić, D., Đorić, B., & Ivakić, S. (2020). Investigating the effect of the use of code.org on younger elementary school students' attitudes towards programming. *Behaviour & Information Technology*. Advance online publication. doi:10.1080/0144929X.2020.1781931

Mason, S. L., & Rich, P. J. (2020). Development and analysis of the elementary student coding attitudes survey. *Computers & Education*, *153*, Article 103898. doi:10.1016/j.compedu.2020.103898.

Mladenović, M., Boljat, I., & Žanko, Ž. (2018). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, *23*(4), 1483–1500. doi:10.1007/s10639-017-9673-3.

Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM*, *62* (2), 32–35.

National Research Council. (2012). *Education for life and work: Developing transferable knowledge and skills in the 21st century*. Washington DC, United States of America: National Academies Press. doi:10.17226/13398

Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, *68*(1), 463–484. doi:10.1007/s11423-019-09708-w.

Opetus- ja kulttuuriministeriö. (2018). *Tasa-arvoisen peruskoulun tulevaisuus* [*Ministry of Education and Culture: The Future of Equal Basic Education*]. http://urn.fi/

Pérez-Marin, D., Hijon-Neira, R., & Martin-Lope, M. (2018). A methodology proposal based on metaphors to teach programming to children. *IEEE Revista Iberoamericana de Tecnologias Del Aprendizaje*, *13*(1), 46–53. doi:10.1109/RITA.2018.2809944.

Piedade, J., Dorotea, N., Pedro, A., & Matos, J. F. (2020). On teaching programming fundamentals and computational thinking with educational robotics: A didactic experience with pre-service teachers. *Education Sciences*, *10*(9) Article 214, 214. doi:10.3390/educsci10090214.

Psycharis, S., & Kotzampasaki, E. (2019). The impact of a STEM inquiry game learning scenario on computational thinking and computer self-confidence. *EURASIA Journal of Mathematics, Science and Technology Education*, *15*(4), Article em1689. doi:10.29333/ejmste/103071

Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2019). Coding in K-8: International trends in teaching elementary/primary computing. *TechTrends*, *63*(3), 311–329. doi:10.1007/s11528-018-0295-4.

Ríos Félix, J. M., Zatarain Cabada, R., & Barrón Estrada, M. L. (2020). Teaching computational thinking in Mexico: A case study in a public elementary school. *Education and Information Technologies*, *25*(6), 5087–5101. doi:10.1007/s10639-020-10213-4.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495. doi:10.1007/s10639-016-9482-0.

Sentance, S., & Waite, J. (2018). Computing in the classroom: Tales from the chalkface. *It - Information Technology*, *60*(2), 103–112. doi:10.1515/itit-2017-0014.

Skolverket. (2019). *Läroplan för grundskolan, förskoleklassen och fritidshemmet* [*Curriculum for compulsory school, preschool and after-school activity*]. https://www.skolverket.se/getFile?file=4206

Stigberg, H., & Stigberg, S. (2020). Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy Futures in Education*, *18*(4), 483–496. doi:10.1177/1478210319894785.

Taraldsen, L. H., & Myhra, K. S. (2019). Programmering med Spheroballer [Programming with Sphero balls]. *Tangenten – Tidsskrift for Matematikkundervisning*, *3*, 2–7.

Utdanningsdirektoratet. (2020). *Læreplanverket* [*National curriculum*]. https://www.udir.no/laring-og-trivsel/lareplanverket/

Utdanningsdirektoratet. (2022). *Algoritmisk tenkning [Computational thinking]*. https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/

Uzumcu, O., & Bay, E. (2020). The effect of computational thinking skill program design developed according to interest driven creator theory on prospective teachers. *Education and Information Technologies*, *26*(1), 565–583. doi:10.1007/s10639-020-10268-3.

Vinnervik, P. (2020). Implementing programming in school mathematics and technology: Teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education*. Advance online publication. doi:10.1007/s10798-020-09602-0.

Vitensenterforeningen. (n.d.). *super:bit skole* [*super:bit school*]. https://www.vitensenter.no/superbit/

Waite, J., Curzon, P., Marsh, W., & Sentance, S. (2020). Difficulties with design: The challenges of teaching design in K-5 programming. *Computers & Education*, *150*. doi:10.1016/j.compedu.2020.103838, Article 103838.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

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

World Economic Forum. (2016). *The 10 skills you need to thrive in the Fourth Industrial Revolution*. https://www.weforum.org/agenda/2016/01/the-10-skills-you-need-to-thrive-in-the-fourth-industrial-revolution/

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education*, *26*(4), 235–254. doi:10.1080/08993408.2016.1257418.

Yallihep, M., & Kutlu, B. (2020). Mobile serious games: Effects on students' understanding of programming concepts and attitudes towards information technology. *Education and Information Technologies*, *25*(2), 1237–1254. doi:10.1007/s10639-019-10008-2.