Research Summaries and Quote Excerpts

Eric Bagley

June 2020

The basis of Sherin's study is to answer the question in what ways are programming-physics and algebraic physics different, and can programming-physics be a viable tool for teaching complex concepts in physics. He compares it to algebraic physics by outlining the key characteristics and interpretive devices (problem-solving tactics and analytic methods) of both. The outcome of his study is that programming physics highlights different interpretive devices and it can be used as a viable physics teaching method.

> The point here is that programming forces a student to engage with these issues in a way that algebra-physics does not. … it is not too surprising that algebra-physics instruction leaves some undiscovered intuitive cobwebs in this territory, and that the act of programing tends to expose these cobwebs p48
>
> ...programming physics draws more strongly on causal intuitions p50
>
> ...programs might be easier to understand or interpret than equations. 3
>
> It has been a frequent observation that physics students tend to use equations without understanding; … This is seen to have devastating effects on the learning that occurs in physics classrooms, since it means that the student's conceptual understanding of physics is not being positively impacted during the many hours of problem solving that are typical of a physics student's' learning activities. Thus if programs are easier for students to interpret – if students are more likely to use them 'with understanding' – there would be important instructional implications. 3

> [At the time of his study computers were not readily available to everyone. Today they are] 8
>
> ...the act of programming requires the students to explode each instant of the motion into a series of actions that happen through what I will refer to as 'pseudo-time' 34
>
> [programming physics introduced a whole new way of learning that were not readily present in algebraic physics] 41
>
> [any method comes with its challenges and benefits] 43, 44

Software-realized scaffolding to facilitate programming for science learning

Guzdial, Mark

`https://drive.google.com/file/d/Wa_-_t2rbsLYeGb32fiBKlUta5stgTf/view?usp=sharing` Guzdial explains the use of programming to teach science topics. He does this using a tool he created called Emile. Emile incorporates scaffolding to assist students in learning the skill of programming in order to not distract from the greater goal of teaching scientific topics. He says "Students use Emile to program models… and simulations … in kinematics. Emile supports students without previous physics or programming background to successfully create models of kinematics and execute these models as simulations and learn about physics in the process." (p2). He outlines what scaffolding is, how fading works, and what his pedagogy and teaching patterns looked like in his study. Conclusions drawn from the study were that Emile was a successful implementation of software realized scaffolding. It both facilitated in teaching programming concepts while also teaching physics modeling through simulation.

> … we are even more interested in having students learn through programming because we recognize that programming is a good lever for understanding many domains. 1 [Outlines the concept of scaffolding] Programming is an activity that leads to a skill, and teachers have techniques for supporting students engaging in an activity and learning a skill. These techniques are called scaffolding. The goal of scaffolding is (a) to enable students to achieve a process or a goal which would not be possible without

> the support, and (b) to facilitate learning to achieve without the support. A critical component of scaffolding is that it be capable of fading: a gradual or leveled reduction in the support provided to that ability increases with students' learning. As students learn the skill, they need less support, so the scaffolding fades. 2
>
> [Explains the challenges of software scaffolding, and what scaffolding is] 3
>
> [explains three things (communicating process, coaching, and eliciting articulation) that comprise scaffolding.] 3,4,5,
>
> [explains the need for fading and why it is important.] 4

Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study

Papadakis, Stamatios - Kalagiannakis, Michail - Zaranis, Nicholas

`https://drive.google.com/file/d/1XPH7LpQrAevPgi8tlX-R8PtJrQksmaPz/view?usp=sharing` This study states that research has found that "teaching programming to young children has a crucial influence on the development of their cognitive functions." They did this using ScratchJr, a block-based programming software that was developed primarily for young children (aged 5-7). They found that preschoolers and young primary schoolers can code and that programming can be used as a teaching tool for other subject areas.

> … school education sometimes seems trapped in following the path of the 19th-century learning logic (teaching and examination). This is opposed to the modern 'road to knowledge', which concentrates on building foundations for the gradual conquest of abstract concepts as means of learning children to 'learn how to learn'. 188
>
> … proof of the benefits of using developmentally appropriate interactive technology has been well documented. 188

Today, computational thinking is such a fundamental skill for everyone that we should add it to every child's analytical ability along with reading, writing, and arithmetic. 188

… numerous studies have confirmed the benefits generated by the teaching of programming concepts in the development of basic cognitive skills, which are associated, for example, with the mathematical ability and the development of logical thinking in children of preschool and early primary school age. 189

The UK is the first country in the world to mandate computer programming in primary and secondary schools 189 (I should look into what today's statistics are)

[Computational thinking] is important because it allows one to solve problems, to design systems, and understand the potential and limitations of human intelligence and of machines. 189

CT is a skill today's students need to be taught, in order to adequately prepare for the workplace but also to be able to participate effectively in the modern digital world. Compared to traditional instruction or information from textbooks, learning with the use of ICT seems to be a more attractive way of learning that can trigger the interest and motivation of preschoolers (Hwang and Chang, 2011). 191

The major goal of programming courses is to help students learn to solve problems with program design rather than merely memorising the syntax of the programming language and the operations of the programming tools (Wang et al., 2015). 191

The strict syntax of the text-based programming languages such as Logo can finally discourage young children. Alternatively, graphics programming environments can potentially simplify the syntax difficulties, but at the same time frequent use of text in such cases poses an additional challenge for children of this age. 192

[ScratchJr] allows young children (5-7 years) to 'discover' the basic programming concepts by creating projects in the form of interactive stories and games. In the 21st century, the demand for the development of computational and digital skills, as elements of the fundamental literacy, will become increasingly intense. 199

> [this study] provided evidence that even preschoolers and kindergartners can learn to code. 199
>
> As a result, ScratchJr could be implemented in early childhood as a teaching tool, set up in a developmentally appropriate way: by integrating other disciplines, helping children develop cognitive, conceptual, language and collaborative skills (Toh et al., 2016). 199

The relationship between mathematical ability and programming ability of computer science education students.

Pramata Sari, Delsika - Sukmawati, Ati - Zulkarnain, Iskandar

`https://drive.google.com/file/d/1BvhUYgCohtW5Unu-CtCJuMM6xMhpCo2d/view?usp=sharing` The purpose of this paper is to show that students with greater interest and success in mathematics often perform better in learning to program.

A conservative approach to special education reform: mainstreaming through transenvironmental programming and curriculum-based measurement

Fuchs, Douglas - Fuchs, Lynn S - Fernstrom, Pamela

`https://drive.google.com/file/d/1K0xmLMFEztOC-hi2xIkcZPajEeT7xjdv/view?usp=sharing` The purpose of this study was to "evaluate the use of Scratch in school lessons as an introduction to programming for total novices, in a younger age-group at primary school." (129) This goal was made with the fact that no previous programming experience would be expected. It used "student-centered" design principles, (130). The following were there conclusions: 1. An active pedagogical approach using a Visual Programming

Language significantly improves several elements: learning programming concepts, logic, and computational practices. 1. Students and observers point out that working with visual programming through projects provides fun, motivation, enthusiasm, and commitment from the students. 3. Perceived usefulness in these practices and computational concepts addressed obtained considerably higher results with average values greater than 4.5. Therefore, the descriptive analysis makes clear the utility and the possibility of learning sequences, loops, parallelism, and events, and sharing content. Fun and communicative possibilities are present in this educational process. 4. Project-based learning carried out in the intervention enables an active approach. This factor obtained values around 4 so active learning is essential and important in this process. The active approach gets positive results and stands out through a teaching methodology centered on the student to develop projects and creations. 5. With regard to art and history contents, results are a little bit higher than 3; therefore, according to the proposed category, students achieve an understanding and comprehension of the aforementioned concepts that is consistent with Bloom's classical taxonomy. 139

> Due to the aforementioned benefits and positive results obtained in this research, it is recommended to implement a Visual Programming Language in educational settings in 5th and 6th grade in primary education through a cross-curricular implementation. 129
>
> ...computational thinking entails logical analysis of data, modeling and abstractions, and implementing possible solutions. 130
>
> Computational thinking plans and coding in education are growing in several countries due to several advantages. 130
>
> Programming is not only a fundamental skill of computational science and a key tool for supporting the cognitive tasks involved in computational thinking but a demonstration of computational competencies as well. 131
>
> The ability to be a creator rather than just a consumer of technology is increasingly seen as an essential skill in order to participate fully in a digital society. 131
>
> Writing computer programs doesn't require special expertise, just clear and careful thinking. 131

...Teachers who were initially skeptical of implementing computing found computer programs such as Scratch and Etoys to be both valuable and accessible.

Visual Programming Languages integrated across the curriculum in elementary school: a two-year case study using "Scratch" in five schools

Saez-Lopez, Jose-Manuel - Roman-Gonzalaz, Marcos - Vazquez-Cano, Esteban

This paper observed and analyzed students over the course of two years and across different grade levels (5-7th). They incorporated Scratch into a cross-curricular pedagogy that helped students develop computational thinking skills. In conclusion the results indicated the following: an active pedagogical approach using visual programming significantly improves learning programming concepts, logic and computational practices. 2 students and observers point out that working with visual programming through projects provides fun, motivation, enthusiasm, and commitment from the student. 3. Perceived usefulness in these practices and the computational concepts addressed obtained considerably higher results. The descriptive analysis makes it clear the utility and possibility of learning sequences, loops, parallelism, and events and sharing content. 4 project based learning carried out in the intervention enables an active approach. Active approach gets positive results and stands out through a teaching methodology centered on the student to develop projects and creations.

It is recommended to implement a visual programming language in educational settings in 5th and 6th grade in primary educational through a cross-curricular implementation. (abstract)

…students often give up computer science because they think it is confusing and difficult. 129

The purpose of the present study is to evaluate the use of scratch in school lessons as an introduction to programming for total novices, in a younger age-group at primary school. 129

The aforementioned application [scratch] is aimed at engaging

young learners to provide an accessible starting point for learning with limited or no programming background. 130

Computational thinking entails logical analysis of data, modeling and abstractions, and implementing possible solutions. 130

[design based research] is proposed as a strategy to innovate in educational contexts, and allows for a systematic strategy focused on learning. It is a naturalistic approach to understanding the processes of learning through informed exploration, enactment, evaluation, within a local context, and development of design principles. 130

Programming is not only a fundamental skill of computational science and a key tool for supporting the cognitive tasks involved in computational thinking but a demonstration of computational competencies as well. 131

Writing computer programs doesn't require special expertise, just clear and careful thinking. 131

Teachers who were initially skeptical of implementing computing found computer programs such as scratch and Etoys to be both valuable and accessible. 132

Common Core Standards in education

Common Core

`http://www.corestandards.org/Math/Content/3/introduction/` Summary:

How design features in digital math games support learning and mathematics connections

Moyer-Packenham, Lommatsch, Lister, Ashby, Bullock, Roxburgh, Shumway, Speed, Covington, Hartmann, Clarke-Midura, Skaria, Westenskow, MacDonald, Symanzik, Jordan `https://drive.google.com/file/d/1NUgtrbIhg6iSA5RwyY87RL_LltNBXHI_/view?usp=sharing` This paper looked deeper into design components of digital math games. They outline some of the key areas. The main research questions they hoped to answer were "How do interactions with digital math games impact children's proficiency?" and "How do design

features in digital math games support learning and promote mathematics connections?" (318). They conclude that post test scores improved significantly. They identified eight prominent categories of design features present across the 12 math games considered. There are two main categories: Learning support, and math connections. Learning support included: accuracy and feedback, unlimited/multiple attempts, information tutorials and hints, focused constraint, progressive levels, and game efficiency. Mathematics connections included: linked representations and linked physical actions. (321). Each of the categories offered different benefits for the students. This article may be useful in designing the coding environment.

> Research on virtual manipulatives has consistently shown positive learning outcomes for children. 319
>
> Research shows that when children use direct touch devices…, in comparison with indirect touch devices(such as a computer mouse), this direct action encourages higher level strategies, less guessing, better accuracy, and faster completion times. 318
>
> [Use of scaffolding in digital math games] 319
>
> [Design features that provide learning support, accuracy feedback, unlimited/multiple attempts, information tutorials and hints, focused constraint, progressive levels, information tutorials and hints, game efficiency] starting at page 322
>
> "There is a need to integrate cues, hints or guide notes to scaffold their learning process" 323
>
> When a digital game includes design features that support user efficiency, this promotes mathematical flow, a state of completely focused motivation, which impacts learning my mediating enjoyment and performance 325
>
> The features of some apps that supported pattern recognition were accessed by high-achieving students; conversely the features that supported processes and procedures were accessed by low-achieving students. 330
>
> Games that provide an introduction, tutorials, helps and hints can support learning. But designers must make choices about the amount of information that is provided in the tutorial of the game to little information can cause children to be unable to complete

> the task whereas too much information can limit children's efforts to discover strategies and make connections. 330

Using open-response fraction items to explore the relationship between instructional modalities and student's solution strategies

Shumway, Moyer-Packenham, Baker, Westensckow, Anderson-Pence, Tucker, Boyer-Thurgood, Jordan

`https://drive.google.com/file/d/16eEH7sWXZ9zvfmZH5YtPY9kUB5mGG9nP/view?usp=sharing` This paper discusses using virtual manipulative representations as well as physical manipulatives and textbook resources in teaching mathematics to elementary-aged children. The findings suggest that both can be viable teaching tools. They found, however, that the two offered different strengths, indicating that using both provides a more comprehensive learning experience.

> … a combination of instructional modalities are necessary in order for students to develop deep and connected understandings of rational numbers. 127

The building blocks of coding: a comparison of early childhood coding toys

Clark-Midura, Shumway, Lee, Hamilton

Summary:

The Emerging Technology Report on Computational Toys in Early Childhood

Hamilton, Clarke-Midura, Shumway, Lee

Summary:

Coding toys in Kindergarten

Shumway, Clarke-Midura, Lee, Hamilton, Baczuk

Summary:

Coding as a playground: Promoting positive learning experiences in childhood classrooms

Bers, Gonzalez-Gonzalez Armas-Torres

Summary:

Programming in preschool - with a focus in learning mathematics

Palmer

This article summarizes a study carried out in three Swedish Preschools. They analyzed the affects programming and computational thinking had on spacial thinking. The results of the study indicated that the children could learn how to program The students were able to understand symbols, interpret them, and knew what they would do in the programming language. They observed a variety of learning and execution methods from each of the students.

Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade

Sullivan, Bers

This paper descibes a study where they use robotics to teach programming. It seemed to focus more on helping students develop a knowledge of robotics

and the components that make up a robot. In addition to this they conclude that 7 year olds can master the basic concepts of programming.

I wonder if this is the nature of the study and not the nature of younger students. I also had the thought that there may be evidence showing that if you start a student too early on concepts that are too difficult they may struggle and fall behind the rest of the time they work with that subject... But that's just a thought of mine.

Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study

Fessakis, Gouli, Marvroudi

This research paper focuses on a younger audience. Its main objectives is to "Explain complex causal links in real life interventions, describe the real-live context in which the intervention has occurred, describe the intervention itself, explore those situations in which the intervention being evaluated has no clear set of outcomes. (89) It investigates the following questions: are kindergarten children capable of solving problems in the specific programming environment? What kind of problems related to the use of the software do they face? Do children enjoy working with the programming environment? What kind of difficulties related to the solving of problems do children encounter? How do they approach the solution of the problems? What are the main characteristics of children's interaction throughout the problem solving process? What is the teacher's attitude and what is her role? What is the teacher's opinion about the feasibility and the pedagogical added value of the specific computer programming learning activities? In order to achieve these objectives and answer these questions, they utilized a logo like program with a ladybug.In the end they conclude students were able to achieve the end goal of the activity and learn mathematical principles while using programming concepts. Many of the children had initial difficulty with communicating the orientation terms and verbalizing directions. They found, in the end, that students performed one of two types of tasks (planning, or trial and error).

Computer programming is considered an important competence for the development of higher-order thinking in addition to

algorithmic problem-solving skills. 87

The research evidence supports the view that children enjoyed the engaging learning activities and opportunities to develop mathematical concepts, problem solving, and social skills. (abstract)

While trying to 'teach the computer how to solve a problem [interesting way to think about it], 'solvers' have the opportunity to articulate their thoughts, watch the outcomes, clarify their thought process and receive immediate feedback… 87

… students having computer programming experiences scored higher on various cognitive-ability tests compared to students without any prior programming experiences. 87

… contemporary educational systems should provide for a smooth and cross-curricular integration of computer programming in all grades. 87

From the first remark, it follows that learning activities involving programming and targeted at young children must be carefully designed so that they are meaningful and challenging (and thus engaging) but also achievable in order to avoid the discouragement of children. 88

Computer programming environments support autonomous or guided open-ended explorations in the process of which the children participate actively, think and control the computer 88

...there is no published research reporting potential negative effects of the use of programming environments 88

Finally, research conducted on young children, supports the pedagogical value of Logo in the learning of mathematics, in the improvement of thinking skills as well as in the development of problem solving strategies.

It is a widespread belief that computer programming based learning activities facilitate the development of algorithmic reasoning and problem solving capability in general. 96.

Review on teaching and learning of computational thinking through programming: What is next for K-12?

Lye, Koh

This article attempts to analyze current research studies (as of 2014) on computational thinking and similar concepts in the k-12 and higher education categories. Their conclusions indicate that exploration and reporting in this area are limited and they suggest more studies be carried out. In addition to this, they suggest possible studies, research topics, and methods to carry them out.

> Programming...exposes students to computational thinking which involves problem-solving using computer science concepts like abstraction and decomposition. Even for non-computing majors, computational thinking is applicable and useful in their daily lives. P51
>
> [Computational thinking] can be considered to be fundamental for k-12 students because it requires "thinking in multiple abstractions" 52
>
> Computational thinking is in line with many aspects of 21st-century competencies such as creativity, critical thinking, and problem-solving. 52
>
> Some of the outcomes suggested by researches are the ability to think more systematically and the development of mathematical and scientific expertise. 52
>
> With respect to Scratch, Brennan and Resnick (2012) proposed three dimensions of computational thinking: computational concepts, computational practices, and computational perspectives. 52
>
> Traditional programming languages such as Java or C++ have representation that closely resembles the computer's way of thinking. On the other hand, visual programming languages use representation that is closer to human language. 53

Ultimately, these tools [Visual programming languages], become "technology-as-partner in the learning process" and can possibly help k-12 students to extend these computational practices towards enhancing their general problem-solving ability. 53

Students were found to be using programming to learn content such as languages or mathematics. In the learning of languages, Burke (2012 suggested that Scratch offered a "new medium through which children can exercise the composition skills they learned within traditional literacy classrooms while also offering the mutual benefit of introducing coding at earlier ages". 54

[these concepts can benefit at risk students] 56

Reflection was also found to encourage the review of one's own learning performance, thereby engaging the students into thinking-doing [as opposed to just doing] 57

… engaging in self-explanation and peer code review could help the students to test and debug. 57

Without guidance on the cognitive aspects of computational practices and computational perspectives , the programming experience may be non-educativ as students are not actively reflecting on their experience. They could be merely doing it in the trial-and-error mode rather than thinking as they are doing. Hence, when planning for programming in k-12 contexts, care needs to be devoted to these two aspects for supporting computational thinking. In essence, the students ought to be thinking-doing and not just doing. 58

Teaching objects-first in Introductory Computer Science

Cooper, Dann, Pausch

Summary:

Slow Off the Mark

Diana Epstein, Raegen T. Miller

This article analyzes teacher preparation and argues that it needs drastic improvement in the STEM areas. They propose 5 suggestions that aim to improve teacher preparation. 1. Increase selectivity in elementary teaching programs 2. Implement compensation packages that are more attractive to STEM college graduates. 3. Include more Math and science pedagogy preparation in teacher education courses. 4. Require certification and licensing to raise the standard for teachers. and 5. Explore other methods to help teachers improve their related STEM skills and knowledge.

STEM starts early: ...

McCleur et al

This paper performs research to determine why STEM pracitices and teaching methods have not been implemented yet in the Elementary education levels. They find the following: 1. Teachers and parents are enthusiastic to start and participate in STEM learning, but they require additional knowledge to do so effectively. 2. Teachers at this age level need more robust training. 3. Parents and technology help connect school home and other learning environments that promote STEM learning. 4. Research and public policies play a critical role in the presence of STEM learning. 5. More emperical studies are needed to convey accurate understanding of science and policies.

Computational Thinking in K-12: A review of the state of the field

Suchi Grover, Roy Pea

> The value of abstraction as CT's keystone (distinguishing it from other types of thinking) is undisputed. Abstraction is "defining patterns, generalizing from specific instances," and a key to dealing with complexity (Wing, 2011).

If basic literacy in Math and Science can be considered essential for all children to understand how our world works, why should school education not lift the hood on all-pervasive computing devices as well? We believe that those in possession of computational competencies will be better positioned to take advantage of a world with ubiquitous computing.

Also underinvestigated is the idea of computing as a medium for teaching other subjects—dovetailing the introduction of CT at K–12 with transfer of problem-solving skills in other domains