



HAL
open science

Students think algorithmically with differential equations

Sharon Calor, Sonia Palha, Laura Kubbe

► **To cite this version:**

Sharon Calor, Sonia Palha, Laura Kubbe. Students think algorithmically with differential equations. Twelfth Congress of the European Society for Research in Mathematics Education (CERME12), Feb 2022, Bozen-Bolzano, Italy. hal-03748502

HAL Id: hal-03748502

<https://hal.science/hal-03748502>

Submitted on 9 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Students think algorithmically with differential equations

Sharon Calor¹, Sonia Palha¹ and Laura Kubbe¹

¹Amsterdam University of Applied Sciences, Faculty of Education, Amsterdam, the Netherlands;
s.m.calor@hva.nl, s.abrantes.garcez.palha@hva.nl, l.kubbe@hva.nl

Keywords: Algorithmic thinking, mathematics, in-service teachers, secondary school, higher education.

Aim and rationale of the study

In the past decade, computational thinking (CT) has been a hot topic in educational research and practice (Grover & Pea, 2013; Selby & Woollard, 2013). CT might have a positive influence on learning mathematics (Barcelos et al., 2018). The aspects of CT include algorithms and algorithmic thinking (AT) (Shelby & Woollard, 2013).

AT is a skill that is not easy to develop for students in secondary and higher education. In-service teachers should be better prepared to teach in ways that involve AT. There is a lack of instructional materials to implement AT in in-service teaching trainer programs. The aim of our research is to develop and investigate tasks that enhance in-service teachers' knowledge about AT.

The rationale of our study relies on a system of assessments originally used in Computer Science Education by Grover et al. (2015, pp 209). It describes eight ideas for a high school curriculum involving algorithmic problem solving, two of which involve AT. These are 1) algorithms and pseudocode and 2) algorithmic flow of control, particularly sequence, serial execution, and loops. We understand AT as 1) understanding how an algorithm works, 2) being able to describe an algorithm as pseudocode, 3) knowing how simple loops work, and 4) understanding how commands are executed in sequence.

Method

Tasks for AT in Dynamical Systems course

In the Dynamical Systems course of our in-service teacher training program, students learn to find exact solutions for differential equations. However, sometimes exact solutions do not exist, and solutions need to be approximated with numerical algorithms. To introduce AT into higher teacher education, we replaced an optional part of the Dynamical Systems course (fractals) with the subject of numerical methods. It is important that students understand numerical algorithms. Students should also be able to implement the numerical algorithms in a computer program and test and debug the program. In other words, students must learn CT skills, in particular learn to think algorithmically, to move from a mathematical model (the differential equation) to a reliable approximate solution. The assignment involves Euler's method to solve a first-order initial value problem. The goal is to make students understand that the steps made with Euler's method form an algorithm, which can be described with pseudocode that includes loops.

We developed a number of tasks that aim to elicit AT. The first design principle regards the context in which the AT takes place. Because we wanted to investigate AT in its natural context (in-service

teacher training programs), we chose a content topic that was part of the higher education program and a subject that involved the use of algorithms in a meaningful way. Therefore, we chose a course in differential equations (Dynamical Systems). The second design principle concerns the nature of the tasks. We developed tasks consisting of activities that require AT, such as creating a pseudocode involving serial execution and/or loops (Grover et al., 2015). In this study, we investigated how one in-service teacher solved these tasks and engaged in AT in the first design cycle.

Analysis

We investigated how the task (the finishing of the pseudocode in Euler's method) can elicit AT through task-based interviews and the thinking out loud method (Schellings et al., 2006). In the thinking out loud method, the student is asked to constantly articulate his or her thoughts. The students were selected by the teacher of the course on the basis of their mathematical knowledge level (two with an average mathematical knowledge level and two with a higher-than-average mathematical knowledge level, and in each pair of students, one student was chosen at random). Each student was interviewed individually by one of the authors of this paper, and the interviews were video recorded and then transcribed. Analysis of the transcripts was based on the categories of Grover et al. (2015). Here, we report on the results of one student.

Preliminary results and further research

Analysis of this student's think-aloud protocol while solving the task revealed aspects of AT. The results showed that the student was able to explain the Euler algorithm and create the pseudocode (including loops), which are forms of AT according to our analytical framework. The assignments therefore seem to have potential for the development of AT. However, there are aspects that need to be improved. The next steps are to further develop our coding scheme for AT and to investigate how our research can contribute to teachers' work on algorithms in their practice.

References

- Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199–237. <http://dx.doi.org/10.1080/08993408.2015.1033142>.
- Schellings, G., Aarnoutse, C., & van Leeuwe, J. (2006). Third graders think-aloud protocols: Types of reading activities in reading and repository texts. *Learning and Instruction*, 16, 549–568. <https://doi.org/10.1016/j.learninstruc.2006.10.004>
- Selby, C. C., & Woollard, J. (2013). Computational Thinking: The Developing Definition. University of Southampton (E-prints). <http://eprints.soton.ac.uk/id/eprint/356481>
- Barcelos, T.S., Muñoz-Soto, Villarroel, R., Merino, E., & Silveira, I. (2018). Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *Journal of Universal Computer Science*. 24 (7), 815–845. <http://dx.doi.org/10.3217/jucs-024-07-0815>