Fourth conference of the International Network for Didactic Research in University Mathematics

INDRUM 2022

October 19-22

Hannover

Leibniz University Hannover

PROCEEDINGS

http://indrum2022.sciencesconf.org/

INDRUM 2022 is an ERME topic conference

erme european society for research in mathematics education

INDRUM

khd m kompetenzzentrum hochschuldidaktik mathematik

Leibniz Universität Hannover

INDRUM2022 PROCEEDINGS

Fourth conference of the International Network for Didactic Research in University Mathematics

19-22 Oct 2022

Hannover

Germany

Editors:	María Trigueros, Berta Barquero, Reinhard Hochmuth, and Jana Peters
Publisher:	University of Hannover and INDRUM
Place:	Hannover, Germany
Year:	2022
ISSN:	2496-1027 (online)

INDRUM 2022 was held with the support of the Faculty of Mathematics and Physics, Leibniz University Hannover, and the Centre for Higher Mathematics Education (khdm).

INDRUM2022 was an ERME Topic Conference.

ERME Topic Conferences (ETC) are conferences organised on a specific research theme or themes related to the work of ERME as presented in associated working groups at CERME conferences. Their aim is to extend the work of the group or groups in specific directions with clear value to the mathematics education research community.

http://erme.site/

Toward a framework for integrating computational thinking into teaching and learning linear algebra

Melih Turgut

NTNU – Norwegian University of Science and Technology, Trondheim, Norway, melih.turgut@ntnu.no

This paper introduces an emerging framework for integrating computational thinking into the teaching and learning of linear algebra. To achieve this, we refer to the notions of three teaching principles of linear algebra, theory of instrumental genesis and computational thinking. Through the emerging framework, we present a vignette involving a set of activities using GeoGebra's specific commands, tools and functions. We approach the case of systems *of linear equations and limit ourselves to a linear algebra course whose students do not have a strong background in a programming language* (*much like the one for lower secondary mathematics teacher education programs in different countries*)*. We propose several further steps to ameliorate the emerging framework.*

Keywords: Teaching and learning of linear and abstract algebra, Digital and other resources in university mathematics education, Computational thinking, Three teaching principles, Instrumental genesis.

INTRODUCTION

The term *computation* is considered one of the basic skills in school curricula; therefore, it could sound familiar to every mathematics teacher and mathematics educator (Li et al., 2020). When we combine computation with *thinking*, it immediately leads to a certain meaning: the practices of computer science, such as coding and programming. However, the notion of computational thinking is a way of thinking that encompasses a number of interrelated thinking skills (e.g., algorithmic thinking and problem-solving) that are beyond computing/programming practices (Lockwood, DeJarnette & Thomas, 2019; Wing, 2006). As a result, a growing body of research has recently attempted to define and characterize computational thinking in mathematics and science education (i.e., Kallia et al., 2021; Weintrop et al., 2016).

In addition to the growing interest in different levels of education (that involve various unplugged and plugged activities), a recent call for higher education has been raised by Lockwood and Mørken (2021). Lockwood and Mørken (2021) invite researchers to focus on machine-based computing activities in undergraduate mathematics education, especially those associated with practices of creating algorithms and running them through digital tools. However, this might require a certain level of maturity in programming languages (Buteau et al., 2020). Lockwood and Mørken's (2021) call has motivated us to focus on an undergraduate linear algebra course. Our question is: *How can computational thinking be integrated into teaching and learning linear algebra?*

We focus on linear algebra because it includes various mathematical notions (e.g., row reduction, echelon forms, linear independence, and rank), and different representations (e.g., equations, vectors, matrices and so on) that are interrelated with algorithmic thinking and problem-solving. Students are often challenged when they encounter such unusual (and new) ideas/steps/representations after high school, and "for a majority of the students, linear algebra is no more than a catalogue of very abstract notions" (Dorier et al., 2000, p. 85). We believe that the integration of the computational thinking perspective would be beneficial for linking different representations specific to linear algebra. Consequently, the current paper introduces an emerging framework for integrating computational thinking practices into linear algebra by grounding the framework in three perspectives as described in the next section.

CONCEPTUAL FRAMEWORK

Three teaching principles of linear algebra.

Learning linear algebra requires a certain level of coordination among different contexts, so designing the teaching setting for this has a core role in arranging the shifts and balance between the (new) notions and representations. Harel (2000) proposes three teaching principles that can be used for designing a teaching setting: concreteness, necessity and generalizability. The concreteness principle considers students' cognitive backgrounds and readiness for learning the proposed concept(s); this is strongly connected to student difficulties. The students need to be equipped with the proposed notions/concepts, and they need to have "… mental procedures that they can take these objects as inputs" (Harel, 2000, p. 180).

The necessity principle is about finding problematic situations that invite students into doing mathematics, and this should correspond to students' intellectual needs. Harel (2000) suggests that considering a *need for computation*, which means providing contexts that ask students to compute objects and explore mathematical properties, is the most effective way to invite students to start a mathematical discussion. This could enable students to find their way by elaborating a number of core ideas from their own work. The generalizability principle is strongly connected to the previous two principles because it enables students to arrive at a generalization in the end. The classroom activities, argumentation and teachers' orchestration of student learning should provide an environment where students move from their (own) work to generalization and the formation of ideas.

Harel (2000) highlights the use of *digital tools* for student exploration and geometry as a pedagogical context to enter a problematic set of situations. Following the three principles above, this context with digital tools should include a particular emphasis on the notion of the "need for computation" and development of ideas and generalization of the mathematical concepts. However, this is based on the manner of "tool use". The tool use shapes student thinking, and this process shapes tool use synchronously (Drijvers, 2019). Here, therefore, we point out the importance of estimation of student thinking (with tool use) to design the teaching-learning context. This brings us to the idea of "hypothetical [utilization] schemes" (Drijvers et al., 2010, p. 113) regarding tool use, which mainly comes from theory of instrumental genesis.

Theory of Instrumental genesis.

Theory of Instrumental Genesis (TIG) is based on the distinction between artifacts and instruments (Artigue, 2002). Here, an artifact can be any material, both physical and/or digital, but it is called a tool when used by the user for a particular aim. When the user develops one or more utilization schemes while using the artifact, we speak of instruments. Here, instruments involve the utilization schemes the user develops over time, in addition to the artifact(s). As a result, this can be simplified into the following formula: "Instrument = Artifact + Scheme" (Drijvers, 2019, p. 15). This process of scheme development is called instrumental genesis (Artigue, 2002). The process of instrumental genesis involves the development of both conceptual and technical elements. However, it is a subtle, continuous, and complex process. Techniques, which are the manner of tool use that lead to accomplishing a task (Artigue, 2002), are observable and explicit. The techniques give us clues regarding those utilization schemes that are invisible. Conceptual elements, on the one hand, convey the techniques that the user develops (over time); on the other hand, they are shaped by the artifact's affordances and constraints (Drijvers, 2019).

In the current paper, we focus on the *hypothetical* [utilization] *schemes* (Drijvers et al., 2010), (under the umbrella of TIG) that capture the synergy between techniques regarding the artifact and conceptual elements and their development. We hypothesize that the estimation of student thinking with tool use could help us design classroom activities with a particular lens that links the three teaching principles.

Computational thinking.

Computational thinking is an "umbrella term" (Kallia et al., 2021, p. 180) that involves a number of overarching and sophisticated skillsets, such as algorithmic thinking, decomposition, modelling, and abstraction. Wing's (2006) seminal description of computational thinking which "… involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science" (p. 33) opened the door to a growing body of research on computational thinking. Weintrop et al. (2016) define a four-category taxonomy regarding computational thinking in mathematics and science education (p. 135): "data practices, modelling and simulation practices, computational problem-solving practices, and systems thinking practices." The commonalities between Wing's (2006) and Weintrop et al.'s (2016) approaches concerning mathematics show a particular link to problem-solving, which means breaking a problem down into subproblems.

Recently, a particular characterization of computational thinking in mathematics education has been proposed by Kallia et al. (2021). This characterization has three main aspects (Kallia et al., pp. 179–180):

- *Problem-solving* (like understanding the problem, developing a solution strategy, performing the strategy),

- *Cognitive processes* (like abstraction, decomposition, pattern recognition, algorithmic thinking, modelling, logical and analytical thinking, generalization and evaluation of solution and strategies),
- *Transposition* (like phrasing the solution of a mathematical problem in such a way that it can be transferred/outsourced to another person or a machine).

The characterization above does not necessarily imply considering all aspects in a setting. For example, following a particular didactical aim, the topic and appropriation of the tools (both physical and digital) would not be practical if one tries to combine all the aspects described above. We concur with Kallia et al. (2021), who note:

> "… maybe some aspects of computational thinking are more critical than others and learning opportunities that consider computational thinking should provide opportunities for students to practice as many aspects as possible." (Kallia et al., 2021, pp. 179–180)

Therefore, based on the context, the teacher or educational designer can focus on specific aspects that invite students to perform (mathematical) explorations. Another fact is that the characterization above does not propose a particular set of tools, even though some of the aspects are directly related to computer science. If we go back to the context of linear algebra, there seem to be many topics related to computational thinking, for example, linear systems (particularly row reduction and echelon forms), matrix transformations and applications to computer graphics, the Gram-Schmidt process and so on.

The emerging framework for task design.

In this subsection, we relate the three teaching principles to those hypothetical utilization schemes with aspects of computational thinking. The first item that we need to address is that the backgrounds of the three teaching principles and TIG seem to be similar regarding students' mental development. The three teaching principles come from a Piagetian perspective (Harel, 2000), while TIG has foundations in both Piagetian (i.e., schemes) and Vygotskian (i.e., tool use) perspectives (Drijvers, 2019). Our particular aim is *not* networking these lenses and checking their grand theories, but rather considering and combining them into design tasks with a computational thinking lens. Regarding the shared theoretical background, there exists a link between tool use and the notion of the need for computation. Before beginning to explain this link, let us address the function of the concreteness principle in the emerging framework. The concreteness principle is carefully specific to the choice of the context. In other words, this principle is something that we can think of as a point of departure to think/decide about the setting. We claim that the following (interrelated) questions would be beneficial in setting the scene:

1) Which topic is going to be considered?
2) What is the (tentative) didactical aim/goal?
3) What do students know, what do they not know (*perhaps this is the most important one*), what would be *concrete* to the students' eyes, and why?
4) How to build on their existing knowledge/phenomenological experiences?

These questions bring us to the necessity principle, indeed to the notion of the "need for computation." To invite students into a rich context that is open to exploration, argumentation, conjecturing, and testing conjectures, we underline the role of tools (Harel, 2000) as mediators:

5) Which tool(s) would be beneficial and why?
6) How would these tools function to achieve the didactical goal?
7) What kind of experience do the students have with the thought tools?
8) Which conceptual elements would emerge when students use the tools?

These four questions imply an estimation of the "manner of tool use" (Artigue, 2002) to design the teaching-learning environment, which brings us to the notion of hypothetical utilization schemes (Drijvers et al., 2010) and TIG. The main aim of these questions is to elaborate on the (hypothetical) techniques and associated conceptual elements that could help us picture/discuss the potentiality of the tools for the didactical aim. The responses to the questions (5 to 8) could be research-informed. A literature search for potential tools and manner of student use would be helpful here as well. The third point concerns embedding the aspects of computational thinking into the eight questions above. The designer could focus on certain aspect(s) in the sense of Kallia et al. (2021) by considering the following (general) question:

9) How would the context and tool enable students to engage with a problem-solving activity, cognitive processes, and transposition?

The generalization principle plays a central role here, and it is linked to the cognitive processes (aspect) of computational thinking. The designer could focus on the function of the tools at stake, along with how this would create a mathematical sense, meaning-making or would help students arrive at a conclusion. Hence, the designer could then finalize their didactical aim/goal. To conclude, we note that the combination and synergy among the nine questions above constitute the emerging framework for task design. Figure 1 summarizes the nine questions, which shows the components as the axes of a Cartesian view.

The designer can refer to Figure 1 by considering ordered triples (from three teaching principles to aspects of computational thinking) while brainstorming. For example, point *A* represents the role of concreteness, the tool and problem-solving and how these three (the selection of the context/aim, student pre-knowledge and deciding on the tool and problem-solving activity) would be aligned. As another example, *B* represents the triple of necessity, conceptual elements, and transposition. Focusing on *B* would help the designer think about and discuss how the need for computation would interlace with the targeted conceptual elements after instrumented activity. Through this way, it can be discussed how the solution of a mathematical problem can be transferred to another person/machine. It may not be necessary to discuss all the possible (27) triples here, however, we believe that Figure 1 would guide the designer as they consider and decide on the function of each component considered here.

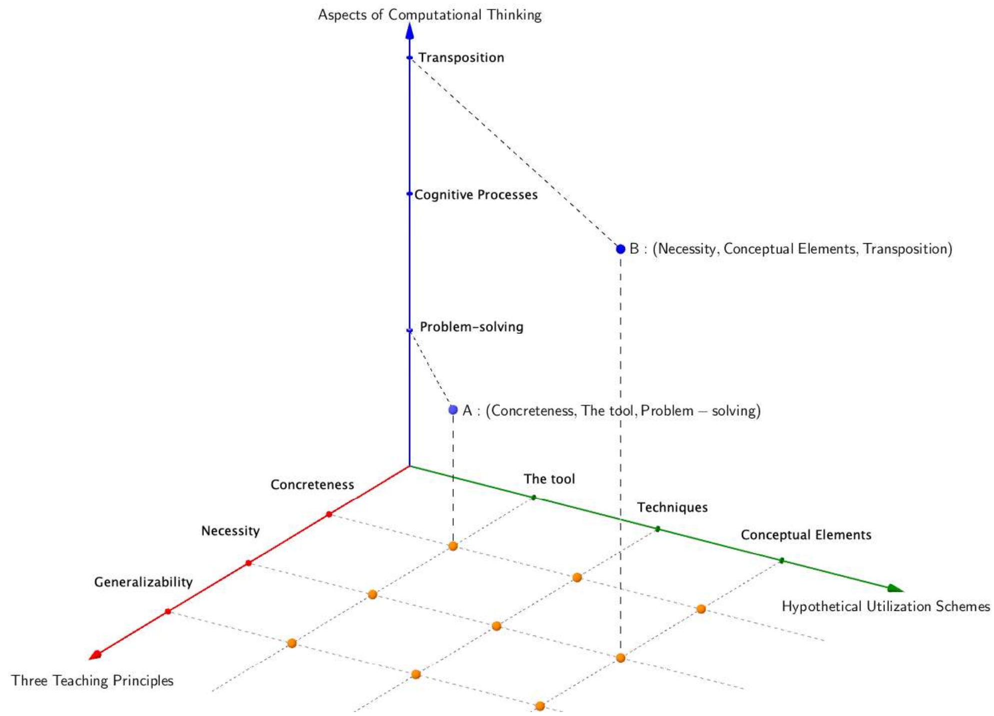

Figure 1: A Cartesian view of the emerging framework

A VIGNETTE

We present an exemplary vignette by following the concreteness principle (particularly point *A* in Figure 1). We have decided on the topic of systems of linear equations (SLE) for two main reasons. The first is the topic (in itself), specifically because SLE is one of the core topics in elementary linear algebra, which has many applications in different fields (Anton & Rorres, 2014). The second reason is SLE's dynamic geometry software availability. We have recently shown how a dynamic geometry environment creates coordination between algebraic and 3D geometric views regarding SLE (Turgut & Drijvers, 2021). The second question in the above subsection and notion for the need for computation help us consider the parameters in the task. As we have experienced (Turgut & Drijvers, 2021), the use of parameters in SLE creates a rich context to explore the geometry of lines and planes in $\mathbb{R}^3$. Therefore, we formulate a tentative aim: making sense of the role of parameters in SLE. We believe that the topic is more relevant after the matrix algebra topic and after the students have learned to solve SLEs on paper-and-pencil activities. Therefore, we plan to build on matrix algebra and consider that the target group does not know the role of the parameters in SLE yet.

The paragraph above can be summarized as addressing those questions from 1 to 4. Now, we focus on questions 5 to 8. In our exemplary case, we refer to GeoGebra based on three criteria. The first is GeoGebra.org's classroom function, where the teacher can design a set of activities/tasks and share the interface of the activity by providing a code. The second is the commands and some tools of GeoGebra that have been recently considered part of the computational thinking activity (van Borkulo et al., 2021). The software is now popular in many countries, so we consider that the target group is familiar with the basic tools/functions. The third is that we focus on the case where the target group does not have a programming language background.

Now, we briefly summarize hypothetical techniques and utilization schemes (based on Turgut & Drijvers, 2021) as follows. While exploring SLE solutions, students could refer to synchronic algebra and geometry windows that provides dynamic variations. The "Solve" command could be used to solve the given equations, and the software would provide different solutions where students could explore different values of the parameters. For example, in certain cases, there is a solution or no solution. Students could type and form matrices of given SLE through a spreadsheet window and attach sliders to matrices. The students later could also refer to the "Reduced RowEchelon Form" command to compute the echelon form of the matrices. This could enable students to see completely zero rows etc. in the matrix and its meaning in the SLE solution, helping them create a link between the role parameters of in row echelon forms. The latter command could also provide a meaning for infinite solution, single solution, or no solution. To plot lines and planes, the students could use the "Input" line and the "Intersect" command, which could provide a geometric feature of the role of parameters (like the intersection of planes and its meaning in the SLE solution).

Regarding computational thinking, we focus on algorithmic thinking and generalization with a problem-solving activity for solving a set of SLEs. In light of this, we re-design a set of activities borrowed from literature (Anton & Rorres, 2014; Turgut & Drijvers, 2021), which are divided into three episodes. The first starts with a figure to overview and link some key notions about SLE and the associated commands of GeoGebra. The first step of the activity (Episode 1) is presented in Figure 2.

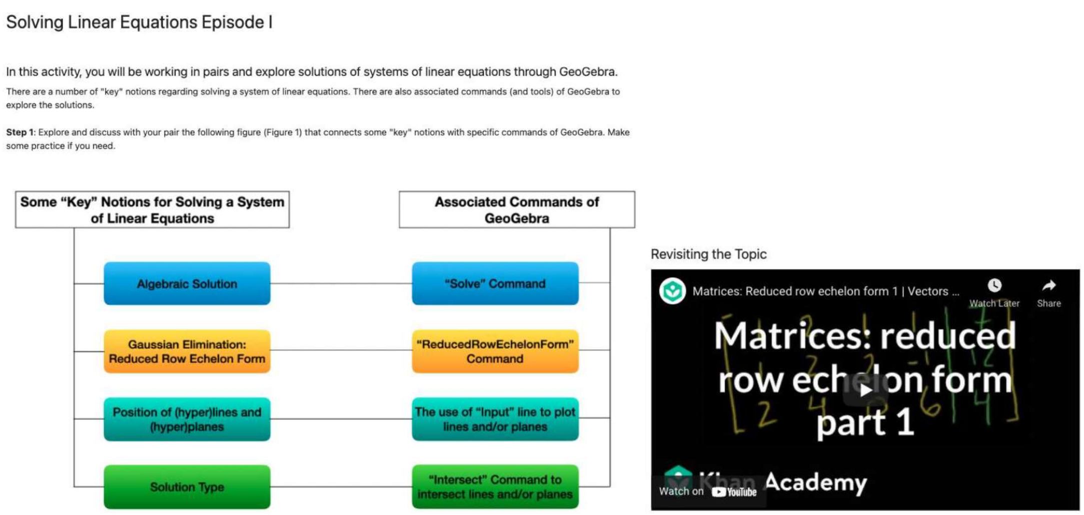

Figure 2: The First Step of Episode I

As a first step (Step 1 in Figure 2), the students discuss some key notions and associated commands of GeoGebra. They can also recall their knowledge by watching some topic-related Khan academy videos (e.g., reduced row echelon form, as seen on the right-hand part of Figure 2). Step 2 asks the students to sketch a tentative algorithm on an embedded (blank) GeoGebra applet to approach solving SLE (by keeping in mind the key notions and associated commands of GeoGebra, as in Figure 2).

Figure 3 summarizes Steps 3 and 4. The third step asks the students to solve a system of linear equations, including a single parameter k: $x + y + z = 4$, $z = 2$ and $(k^2 - 4) \cdot z = k - 2$. We provide a blank GeoGebra applet that includes an algebra window, and 3D graphics window with a spreadsheet window by considering the hypothetical utilization schemes explained on the previous page. For example, the use of the slider k could be referred to solve the proposed SLE.

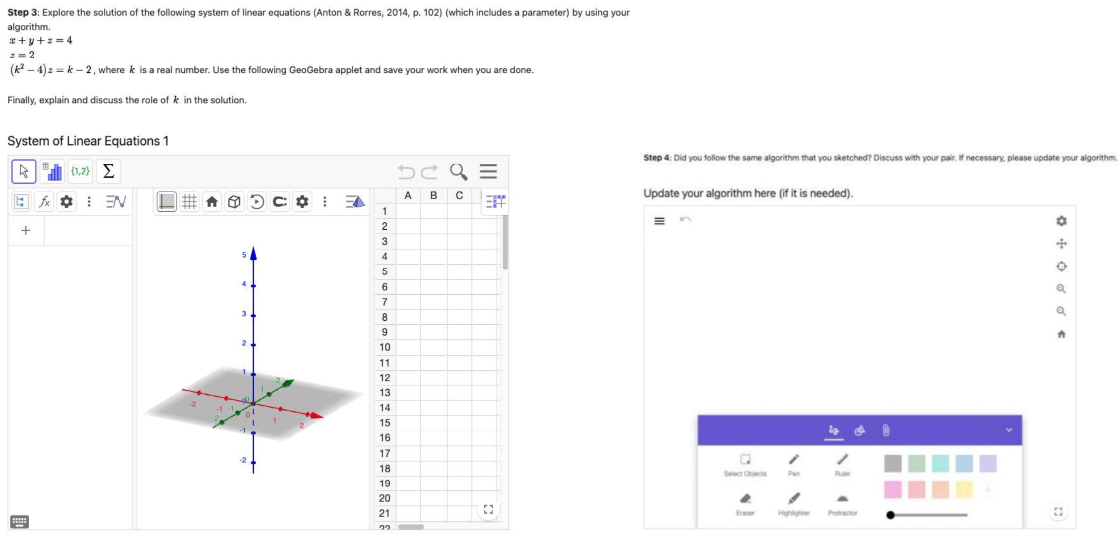

Figure 3: Steps 3 and 4 in Episode 1

In this part, the use of a slider as k is an estimated technique that could help to explore the dynamic effects of the parameter in the given SLE. As a fourth step, the students must discuss the initial algorithm after they have solved the given SLE with a single parameter. The final step of Episode 1 asks to discuss the role of k in the given system (the left-hand-side of Figure 4).

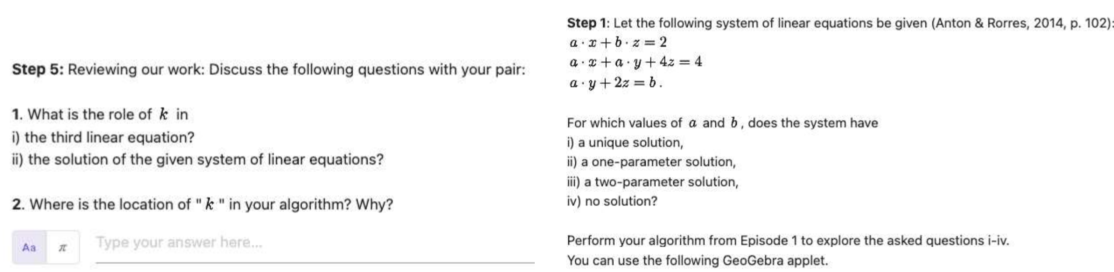

Figure 4: Step 5 of Episode 1 and Step 1 of Episode 2

The second episode starts with another system (the right-hand-side of Figure 4) that has two specific parameters: *a* and *b*. The user is asked to use the (updated) algorithm while solving the given system. As in Episode 1, as a next step, the students are asked to review the algorithm after solving SLE, and then discuss questions to overview the role of parameters in SLE. However, the parameter(s) in the given two SLE appear both in coefficients and known parts. Therefore, in the final episode, a specific system is proposed: $x + y + z = a$, $2x + 2z = b$, and $3y + 3z = c$, which has three parameters. In the final SLE, all three parameters are defined in the known part of the system. This episode also follows reviewing algorithms and making a generalization regarding the role of parameters in SLE by overviewing all episodes and (revisiting) all versions of algorithms.

CONCLUSIONS

In the current paper, we have introduced an emerging framework for integrating computational thinking into teaching and learning linear algebra. We present an exemplary case where the aim is to promote the knowledge and role of parameters in SLE and associated solutions. We note that the emerging framework needs further elaborations (e.g., through design-based research) to discuss its functioning in teaching learning settings. For example, the presented set of activities could be merged and field-tested with some come from matrix transformations and applications to computer graphics, and Gram-Schmidt process. As a limitation, the exemplary case is GeoGebra centric. Another context, such as R, Python or Trinket, could be focused on when designing machine-based computing activities (Lockwood & Mørken, 2021). These could be further steps to ameliorate the presented emerging framework.

Acknowledgement

I would like to thank Paul Drijvers for his input on earlier version of this paper.

REFERENCES

Anton, H., & Rorres, C. (2014). *Elementary Linear Algebra* (11th Edition). John & Wiley Sons.

Artigue, M. (2002). Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *International Journal of Computers for Mathematical Learning*, *7*(3), 245–274. https://doi.org/10.1023/A:1022103903080

Buteau, C., Muller, E., Mgombelo, J., Sacristan, A.I., & Dreise, K. (2020). Instrumental genesis stages of programming for mathematical work. *Digital Experiences in Mathematics Education, 6*(3)*,* 367–390. https://doi.org/10.1007/s40751-020-00060-w

Dorier, J.-L., Robert, A., Robinet, J., & Rogalski, M. (2000). The obstacle of formalism in linear algebra. In J.-L. Dorier (Ed.), *On the Teaching of Linear Algebra* (pp. 85–124). Kluwer Academic Publishers.

Drijvers, P. (2019). Embodied instrumentation: combining different views on using digital technology in mathematics education. In U.T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (Eds.), *Proceedings of the 11th Congress of the European*

Society for Research in Mathematics Education (pp. 8–28). Freudenthal Group & Freudenthal Institute, Utrecht University and ERME.

Drijvers, P., Kieran, C., & Mariotti, M.A. (2010). Integrating technology into mathematics education: Theoretical perspectives. In C. Hoyles & J.-B. Lagrange (Eds.), *Mathematics Education and Technology – Rethinking the Terrain* (pp. 89–132). Springer International Publishing.

Harel, G. (2000). Three principles of learning and teaching mathematics: Particular references to linear algebra. In J.-L. Dorier (Ed.), *On the Teaching of Linear Algebra* (pp. 177–189). Kluwer Academic Publishers.

Kallia, M., van Borkulo, S.P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education*, *23*(2), 159–187. https://doi.org/10.1080/14794802.2020.1852104

Li, Y., Schoenfeld, A.H., diSessa, A.A., Graesser, A.C., Benson, L.C., English, L.D., & Duschl, R.A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research,* *3*(1), 1–18. https://doi.org/10.1007/s41979-020-00030-2

Lockwood, E., & Mørken, K. (2021). A call for research that explores relationships between computing and mathematical thinking and activity in RUME. *International Journal of Research in Undergraduate Mathematics Education*, *7*(3), 404–416. https://doi.org/10.1007/s40753-020-00129-2

Lockwood, E., DeJarnette, A.F., & Thomas, M. (2019). Computing as a mathematical disciplinary practice. *Journal of Mathematical Behavior*, *54*, 100688. https://doi.org/10.1016/j.jmathb.2019.01.004

Turgut, M., & Drijvers, P. (2021). Instrumentation schemes for solving systems of linear equations with dynamic geometry software. *International Journal of Technology in Mathematics Education*, *28*(2), 65–80. https://doi.org/10.1564/tme_v28.2.01

van Borkulo, S., Chytas, C., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Computational thinking in the mathematics classroom: Fostering algorithmic thinking and generalization skills using dynamic mathematics software. In M. Berges, A. Muhling & M. Armoni (Eds.), *Proceedings of WiPSCE'21: The 16th Workshop in Primary and Secondary Computing Education* (pp. 1–9). ACM.

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. https://doi.org/10.1007/s10956-015-9581-5

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