Computer-Verified Linear Algebra Proofs With Programming in MATLAB

Sepideh Stewart
University of Oklahoma

Caleb Judkins
University of Oklahoma

Research on teaching and learning proofs in linear algebra is scarce. This paper used the instrumentation genesis approach to examine students' proofs and MATLAB codes. The students were studying a proof-based second course in linear algebra. The instructor included application-focused labs and a final project as part of this study. One of the labs required a proof validation using MATLAB. This lab, a proof question from the midterm exam, and an exit survey were analyzed in this paper.

Keywords: linear algebra, proof, programming, MATLAB

Literature Review

Proof is a critical component of mathematics. In light of the current technological advances, has the way mathematicians do proof changed? According to Hanna, Reid, and de Villiers (2019, p. 3):

> Proving is sometimes thought to be the aspect of mathematical activity most resistant to the influence of technological change. While computational methods are well known to have a huge importance in applied mathematics, there is a perception that mathematicians seeking to derive new results are unaffected by the digital era. The reality is quite different. Digital technologies are influencing the way mathematicians work together and the way they go about proving.

In an article by Bayer et al. (2024), ten authors from various stages of academic careers individually expressed their views on proof within eight segments of the article. The article began by stating that:

> There is a striking difference between how a proof is defined in theory and how it is used in practice. This puts the unique status of mathematics as exact science into peril. Now may be the time to reconcile theory and practice, i.e., precision and intuition, through the advent of computer proof assistants. This used to be a topic for experts in specialized communities. However mathematical proofs have become increasingly sophisticated, stretching the boundaries of what is humanly comprehensible, so that leading mathematicians have asked for formal verification of their proofs. At the same time, major theorems in mathematics have recently been computer-verified by people outside of these communities, even by beginning students (p. 79).

The article "investigates the different definitions of a proof, the gap between them, and possibilities to build bridges. It is written as a polemic or a collage by different members of the communities in mathematics and computer science at different stages of their careers, challenging well-known preconceptions and exploring new perspectives" (p. 79). As Tall (2023) mentioned, different communities have their own needs and view things (in this case, proof) differently.

Shifting our attention to pedagogy and classroom practices, Melhuish et al. (2022) synthesized 104 papers from various countries and methods to describe what is known about proof-based learning and what is still missing. Their exploration uncovered two schools of thought for teaching proofs: lecture-based proofs and student-centered learning. While student-based learning is promising, a significant gap remains in "understanding of the theoretical mechanisms which specific instructor moves may encourage or scaffold student activity is only beginning to emerge" (Melhuish et al., 2022, p. 17). Acknowledging the wealth of studies in proof in mathematics education literature, we focus our attention on linear algebra proofs in this paper.

Linear algebra is a key topic for many mathematics majors and other fields. In the early 1990s, Steven Leon and his colleagues initiated a national project (ATLAST) to teach linear algebra with MATLAB (Leon, Herman, & Faulkenberry, 2002). Despite these national initiatives, research studies on linear algebra and using MATLAB in classrooms are lacking. The Linear Algebra Curriculum Study Group (LACSG) 2.0 recommends that mathematics departments offer a variety of second courses, including numerical linear algebra, and include wider topics (Stewart et al., 2022). The recommendations also paid significant attention to the use of technology. In a survey paper by Stewart, Andrews-Larson, and Zandieh (2019), the authors found that research on topics in second courses of linear algebra, which contain more abstract content and proof, is rare. In response to this gap, in a study by Meyer, Stewart, and Madden (2024), they employed Schoenfeld's (2010) theory of Resources, Orientations, and Goals (ROGs) to examine an instructor's (and co-author's) textbook in a second course in linear algebra. The instructor created tailored resources with many hints and pedagogical attributes to encourage students to construct their own proofs to understand and learn the materials and invited students to explore, conjecture, and prove.

Our current paper examines students' instrumentation genesis using MATLAB to verify proofs, a proof question from the midterm exam, and an exit survey.

Theoretical Framework

Our theoretical framework for this study is based on the instrumentation genesis (Vérillon & Rabardel, 1995). An illustration of the instrumental genesis is shown in Figure 1. In the instrumentation approach, an instrument starts as an artifact before the subject interacts with it and only becomes an instrument "when the subject has been able to appropriate it for himself and has integrated it." (Vérillon & Rabardel 1995, p. 84). This development into an instrument is called "instrumental genesis" (Sinclair et al., 2023, p. 398), and it incorporates more than just the subject and the artifact itself. An important aspect of an instrument includes the actions taken and the goals of the subject that led to those actions (Trouche, 2018). It is significant to note that the artifact does have an impact on the subject's experience, as the subject's actions are formed largely in response to the design and limitations of the artifact and their own comprehension. The subject must impart their own knowledge of both the goal and the artifact into the development of the instrument, an aspect of the process that shapes the resulting instrument (Balacheff, 1994). In a sense, "the artifact does not intervene in a neutral way in the teaching-learning situation. It carries its own techniques, instrumental knowledge, signs, meanings, and a technological transposition of mathematical knowledge also takes place (Sinclair et al., 2023, p. 399). According to Trouche (2018, p. 407):

> This frame leads us to clarify our initial definition: instrumentation is the action by which a subject acquires an artifact, and the effect of this action on the subject, who develops, from this artifact, an instrument for performing a task. An instrument is thus made up of

an artifact component and a cognitive component (knowledge necessary for/ from using the artifact for performing this type of task).

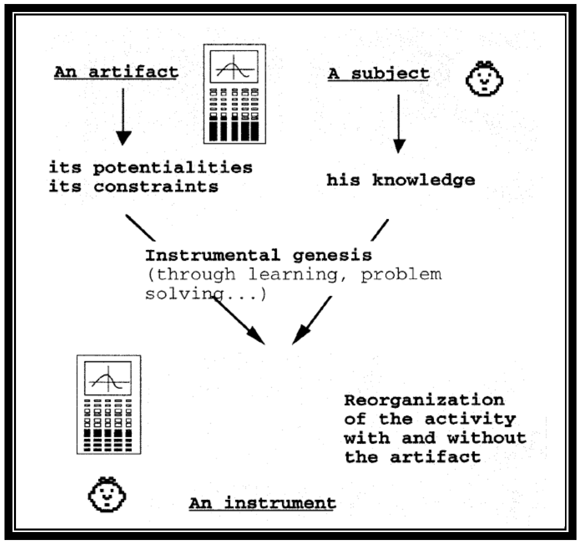

Figure 1: Instrumental genesis, from an artifact to an instrument (Guin and Trouche, 1999, p. 202).

As described by Sinclair et al. (2023), the perspective of instrumentation has been employed widely in the literature, especially with graphics calculators. Notably, Gueudet et al. (2022) studied one student's ability to use programming to explore various mathematical conjectures by examining the evolving schemes surrounding his work. Their framework sought to orient the relationship between tools and students as having an impact in both directions, with the student adapting the programming process to his needs and the student adapting to the operations of the programming language.

Our research question to guide the study was the following: What was the nature of the instrumentation of programming with MATLAB in the production of linear algebra proof?

Method

This is a case study of a proof-based second course in linear algebra. There were 22 mathematics and engineering students enrolled in this course, and 15 students agreed to participate in this study. The first-named author was teaching this second course, which was highly theoretical and proof-based, and selected the textbook *Linear Algebra Done Right* by Axler (2024) for this course. The course was modified to incorporate six labs (one on proof and five on applications) and a final project using MATLAB to help students explore numerical linear algebra and its applications via programming. The instructor (first author) met weekly with a specialist and a support person from MathWorks to create the labs and provided help with MATLAB for students. The course covered the following topics: Vector spaces and their properties (including special Vector Spaces such as Isomorphic Vector Spaces and Invertibility), subspaces, span and linear independence, bases, dimension, linear maps, polynomials, eigenvalues, eigenvectors, and invariant subspaces, inner product spaces/ operators on inner product spaces (The Spectral Theorem, Self-Adjoint, and Normal Operators, etc.), and trace and determinant. The course was taught as a mixture of lectures and various group activities, such as evaluating proofs for clarity and elegance using MATLAB in class to validate results left by the textbook as an exercise. The class participation also included unpacking proofs in their own words and sometimes coming up with different proofs and presenting them to the class.

The data for this study was collected from students' class assessments (final exam, midterm exam, and homework), the labs, the final project, and an exit survey. This paper will only show data analysis from two students' (Student 1 and Student 11) lab 2 and a midterm test question related to the proof of Theorem 2.30. Lab 2 asked students to validate the proof of Theorems 2.19 and 2.30 from the textbook (see Figure 2 for the statement of the theorem and proof).

The exit survey questions had two parts (Proof and MATLAB). A sample of proof survey questions were: What is the purpose of the proofs in linear algebra? Describe the nature of the proofs in linear algebra. How can we best teach linear algebra proofs to enhance your learning experiences? What were the best teaching techniques presented in class that helped you learn proofs? A sample of MATLAB survey questions were: Were you familiar with MATLAB before this course? Have you done any programming before this course? Did you learn coding in this course? Did your knowledge of linear algebra increase as a result of coding?

We have included the results of whole class responses to survey questions related to MATLAB, as well as the results for Students 1 and 11 on proof questions.

2.30 *every spanning list contains a basis*

Every spanning list in a vector space can be reduced to a basis of the vector space.

Proof Suppose $v_1, \ldots, v_n$ spans V. We want to remove some of the vectors from $v_1, \ldots, v_n$ so that the remaining vectors form a basis of V. We do this through the multistep process described below.

Start with B equal to the list $v_1, \ldots, v_n$.

Step 1

If $v_1 = 0$, then delete v_1 from B. If $v_1 \neq 0$, then leave B unchanged.

Step *k*

If v_k is in $\operatorname{span}(v_1, \ldots, v_{k-1})$, then delete v_k from the list B. If v_k is not in $\operatorname{span}(v_1, \ldots, v_{k-1})$, then leave B unchanged.

Stop the process after step n, getting a list B. This list B spans V because our original list spanned V and we have discarded only vectors that were already in the span of the previous vectors. The process ensures that no vector in B is in the span of the previous ones. Thus B is linearly independent, by the linear dependence lemma (2.19). Hence B is a basis of V. ■

2.19 *linear dependence lemma*

Suppose $v_1, \ldots, v_m$ is a linearly dependent list in V. Then there exists $k \in \{1, 2, \ldots, m\}$ such that

$$v_k \in \operatorname{span}(v_1, \ldots, v_{k-1}).$$

Furthermore, if k satisfies the condition above and the k^{th} term is removed from $v_1, \ldots, v_m$, then the span of the remaining list equals $\operatorname{span}(v_1, \ldots, v_m)$.

Figure 2: Axler (2024, pp. 33 & 40).

Results

Case of Student 1

The portion of code not included involves the student creating random vectors using linear combinations of basic unit vectors already in a list by multiplying them with random scalars. In this code (see Figure 3), the student starts by clearing all zero vectors from the list. Next, it is worthwhile to note the last part of this code, as the student creates a function that tests if an additional vector is in the span of a given list by comparing the rank of the list with and without the vector. Student 1 used this function in the next loop to see if the ith vector is in the span of the previous vectors. If it is, then delete that vector from the list. Thus, the resulting list is a linearly independent list of vectors and forms a basis. The only shortcoming of the code is that it does not necessarily check what the spanning space is. However, this could be considered a demonstration of Student 1's understanding that for any vector space, the number of vectors in a linearly independent list gives you the dimension.

The design of Student 1's code is also significant, as the concept of checking the *i*th vector against all previous being sufficient does directly come from the dependence lemma. In this way, Student 1 seems to have made a significant connection between their use of MATLAB as an instrument and the proofs being computer-verified.

Student 1's proof of Theorem 2.30 (see Figure 3) is similar to Axler's (2024) proof (see Figure 2). They both used the linear dependence lemma to iteratively delete linearly dependent vectors from the list until they reached a list satisfying the definition of a basis. However, Student 1 also incorporates the theorem that a linearly independent list of vectors has a length less than or equal to the length of any spanning set of vectors for that space, though this statement is not necessarily being used in the proof itself. This directly connects to their code's test of comparing the rank of a list with and without a vector appended, suggesting their instrumentation of MATLAB has impacted their understanding of the proof.

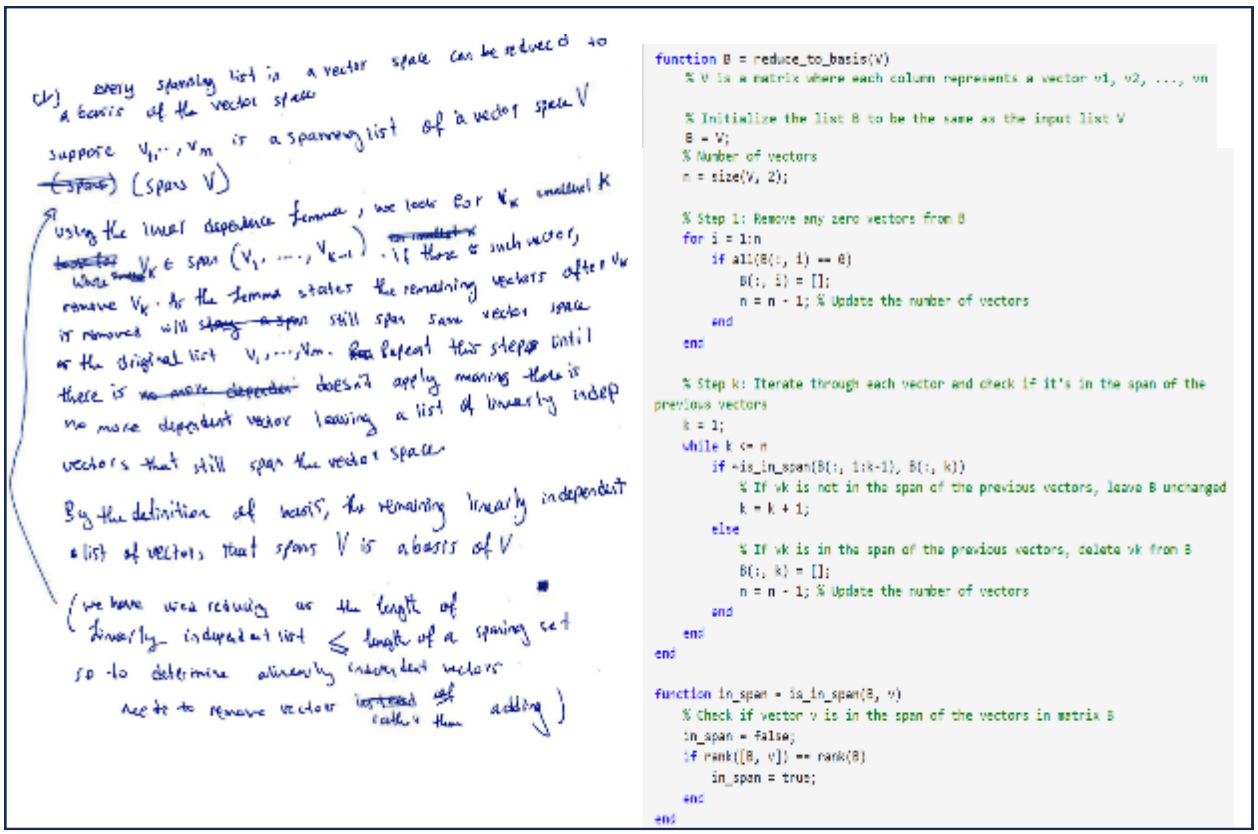

Figure 3: Student 1's proof and MATLAB code for Theorem 2.30.

Case of Student 11

While not included in this block of code (see Figure 4), Student 11 finished this exercise by verifying if the list was a basis by checking the rank against the length of the list. However, there is concern regarding the student's ability to construct vectors that meaningfully demonstrate the theorem. While generating a list of vectors, the student forced it to be linearly independent. Thus, there is not a general list of vectors being tested; there is only a list that is already going to be a basis. However, it does seem that the student has a well-developed understanding of what qualities of a list are necessary to be a basis for the space, as they verified that the length matches the rank of the list. That being said, it is not a proper test of the theorem, as they only check whether their list has a property already built into it. It is worth pointing out that the construction of a list of vectors worth testing is an entirely distinct exercise from testing the qualities of a given list, so the student did seem to adapt the work that they had practiced and learned to MATLAB even though they were not able to properly develop a new skill in creating a sufficient list of vectors.

Student 11's proof produced in the midterm exam (see Figure 4) started with a proof of the linear dependence lemma (Theorem 2.19), which was not required as long as the result of the theorem was mentioned. In the next steps, the conditions were not set up correctly, and it is not clear what step is getting repeated and for what reason. The proof ends with the definition of linear independence without mentioning the concept of a basis.

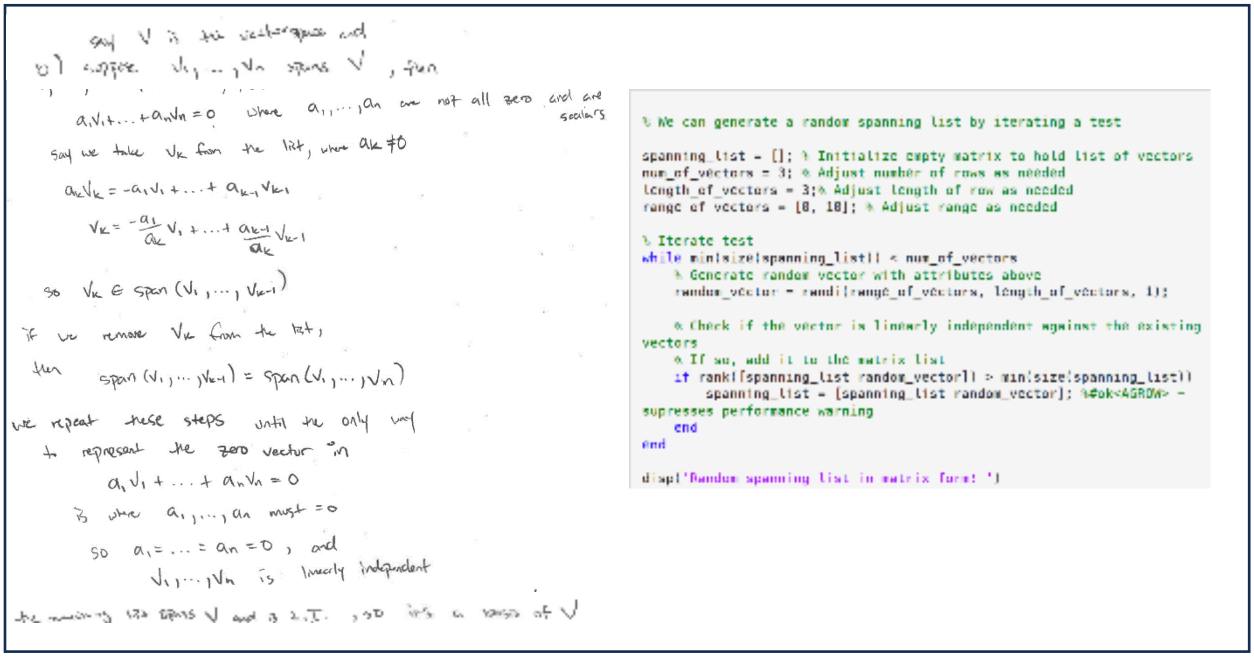

Figure 4: Student 11's proof and MATLAB code for Theorem 2.30.

Analysis of the Exit Survey

The results of the first portion of the exit survey are related to the purpose and nature of linear algebra proofs. Students 1 and 11 both referenced that proofs offer some form of verification. However, Student 11 also claimed that proofs help to understand by providing alternate ways of perceiving the concepts and equipping learners with "a skillset for applied work". Both students mentioned that proofs connect and incorporate different concepts.

Regarding teaching and how students' learning could be enhanced, Student 1 claimed that they best learned proofs by comparing different proofs. Student 11, on the other hand, suggested that they learned best by studying proofs and looking at examples of the relevant mathematics being worked out beside it.

For the best teaching techniques presented in class that helped to learn proofs, Student 1 directly referenced the labs, saying, "…the labs … showed how many of the concepts were applied. Though I enjoy theory more, I cannot deny that application further strengthens my understanding." Meanwhile, Student 11 claimed they learned best from the verbal explanations of the proofs in class.

The second section of the survey was designed to assess the students' perceptions of MATLAB and its use in the course. The results showed that four out of thirteen students were familiar with MATLAB before this course, including Students 1 and 11. Additionally, twelve students reported that they had learned coding in the course. In regards to whether or not coding helped their understanding of linear algebra, some responses were similar to one student's claim that it enabled them to "see what the theory was", as multiple students referenced some form of visualization even with labs that did not offer anything apart from symbolic results. In contrast, some students suggested it helped them learn how to apply the theorems in linear algebra. Additionally, there were a number of students claiming it helped them understand the theorems, such as Student 2's response, "Labs 2 and 6, and the final project were very important to my understanding of 2.19 [the linear dependence lemma], Gram-Schmidt, SVD, and other concepts."

Discussion and Conclusions

The results of Student 1's performance (MATLAB code and the proof of theorem) revealed that although the proof was available at the time of writing the MATLAB code, it required specific skills and expertise. For example, how to make functions, use the commands available, generate random vectors, modify lists of vectors, validate their own codes, and follow the proofs to reach a desired goal with the artifact. While the algorithmic structure of the proof made it somewhat ideal for programming, the translation of the proof into MATLAB was not trivial. As Sinclair et al. (2023) pointed out, the artifact carries its own techniques. Based on the differences between Axler's (2024) proof and Student 1's validation of it, the intervention of the artifact seems to have impacted the transposition of their proof understanding into the MATLAB environment. In the limited data analyzed here, there is an argument for some potential relationship between the quality of a student's code and their proof-writing abilities. Notably, the entire process in this particular lab was based on an abstract theorem, did not include any numerical computations, and naturally added to the complexity of the process for students. While we can conclude the results from the instrumentation process, a closer examination of students' instrumental genesis would be necessary to better understand the possible role of MATLAB in developing their understanding of the proof concept.

The results of the exit questionnaire revealed that students had a positive perception of the role MATLAB played in their learning and exposure to new experiences in an abstract course.

The authors are in the process of analyzing the remaining labs, which are primarily related to applications. Future research projects will include evaluating students' abilities to validate other linear algebra proofs with programming.

References

Axler, S. (2024*). Linear Algebra Done Right*, (4th ed.) Springer.

Balacheff, N. (1994). Didactique et intelligence artificielle. Recherches en Didactique des Mathématiques, 14(1–2), 9–42.

Bayer, J., Benzmuller, C., Buzzard, K., David, M., Lamport, L., Matiyasevich, Y., Paulson, L., Schleicher, D., Stock, B., & Zelmanov, E. (2024). Mathematical proof between generations. *The Notices of AMS*, *71*(1), 79-92.

Gueudet, G., Buteau, C., Muller, E. (2022). Development and evolution of instrumented schemes: a case study of learning programming for mathematical investigations. *Educ Stud Math* 110, 353–377. https://doi.org/10.1007/s10649-021-10133-1

Guin, D, & Trouche, L. (1999). The complex process of converting tools into mathematical instruments. The case of calculators. Int J Compute Math Learn3(3):195–227

Hanna, G., Reid, D., de Villiers, M. (2019). Proof Technology: Implications for Teaching. In: Hanna, G., Reid, D., de Villiers, M. (eds.) Proof Technology in Mathematics Research and Teaching. *Mathematics Education in the Digital Era*, vol 14. Springer, Cham.
Leon, S., Herman, E., & Faulkenberry, R. (2002). ATLAST Computer Exercises for Linear Algebra (2nd ed.) Pearson.
Melhuish, K., Fukawa-Connelly, T., Dawkins, P. C., Woods, C., & Weber, K. (2022). Collegiate mathematics teaching in proof-based courses: What we now know and what we have yet to learn. *The Journal of Mathematical Behavior*, *67*, 100986.
Meyer, J., Stewart, S., & Madden, A. (2024). Teaching Formal Proofs in a Second Linear Algebra Course: A Mathematician's Resources, Orientations, Goals and Continual Decision Makings. In Cook, S., Katz, B. & Moore-Russo D. (Eds.). *Proceedings of the 26th Annual Conference on Research in Undergraduate Mathematics Education*. Omaha, NE. (pp. 887-895).
Schoenfeld, A. H. (2010). *How we think. A theory of goal-oriented decision making and its educational applications*. Routledge: New York.
Sinclare, N., Haspekian, M., & Robutti, O. (2023). Revisiting Theories That Frame Research on Teaching Mathematics with Digital Technology. In *The Mathematics Teacher in the Digital Era* (Vol. 16, pp. 391–418).
Stewart, S. Andrews-Larson, C., & Zandieh, M. (2019). Linear algebra teaching and learning: Themes from recent research and evolving research priorities, *ZDM Mathematics Education*, 51(7), 1017-1030.
Stewart, S., Axler, S., Beezer, R., Boman, E., Catral, M., Harel, G., McDonald, J., Strong, D., & Wawro, M. (2022). The linear algebra curriculum study group (LACSG 2.0) recommendations. *The Notices of American Mathematical Society*, 69(5), 813-819.
Tall, D. O. (2023). Long-term principles for meaningful teaching and learning of mathematics. In S. Stewart (ed.), *Mathematicians' Reflections on Teaching: A Symbiosis with Mathematics Education Theories*, *Advances in Mathematics Education* (pp. 217–252). Springer.
Trouche, L. (2018). Instrumentation in mathematics education. In S. Lerman (Ed.), *Encyclopedia of mathematics education* (pp. 404–412). Springer.
Vérillon, P., & Rabardel, P. (1995). Cognition and artifacts: a contribution to the study of thought in relation to instrumented activity. *European Journal of Psychology of Education*, 10(1), 77–101.