



Mathematics in the Digital Age: The Case of Simulation-Based Proofs

Moritz Lucius Sümmerrmann¹ · Daniel Sommerhoff² · Benjamin Rott¹

Accepted: 27 November 2020 / Published online: 15 February 2021
© The Author(s) 2021

Abstract

Digital transformation has made possible the implementation of environments in which mathematics can be experienced in interplay with the computer. Examples are dynamic geometry environments or interactive computational environments, for example GEOGEBRA or JUPYTER NOTEBOOK, respectively. We argue that a new possibility to construct and experience proofs arises alongside this development, as it enables the construction of environments capable of not only showing predefined animations, but actually allowing user interaction with mathematical objects and in this way supporting the construction of proofs. We precisely define such environments and call them “mathematical simulations.” Following a theoretical dissection of possible user interaction with these mathematical simulations, we categorize them in relation to other environments supporting the construction of mathematical proofs along the dimensions of “interactivity” and “formality.” Furthermore, we give an analysis of the functions of proofs that can be satisfied by simulation-based proofs. Finally, we provide examples of simulation-based proofs in ARIADNE, a mathematical simulation for topology. The results of the analysis show that simulation-based proofs can in theory yield most functions of traditional symbolic proofs, showing promise for the consideration of simulation-based proofs as an alternative form of proof, as well as their use in this regard in education as well as in research. While a theoretical analysis can provide arguments for the possible functions of proof, they can fulfil their actual use and, in particular, their acceptance is of course subject to the sociomathematical norms of the respective communities and will be decided in the future.

Keywords Digital environments · Mathematics education · Proofs · Simulations

✉ Moritz Lucius Sümmerrmann
moritz.suemmermann@uni-koeln.de

Extended author information available on the last page of the article.

Introduction

Proofs can come in many forms, ranging from a system of logical deductions done in a formalistic symbolic way, as in the *Principia Mathematica* (Whitehead and Russell 1910), over plain text argumentations supplemented by formulas, which are the standard form in many research papers and textbooks, to so-called “Proofs Without Words” given by an image only (Nelsen 1993).

Computers have added to this variety by giving rise to computable proofs, i.e. proofs that can be checked by a computer (Voevodsky 2015), or even proofs executed by a program such as the much debated (Tymoczko 1979) proof of the Four-Color Theorem (Appel and Haken 1977), too long to ever be completely reviewed by a human.

The rise of computers and with it digital transformation of all aspects of human activities has also made environments possible that enable to do mathematics in an interactive way. These environments can come in different representation modes, as some are more formal-symbolic in nature, such as MATHEMATICA or JUPYTER NOTEBOOK, while others have rather informal visual representations, such as GEOGEBRA for geometry or ARIADNE for topology¹. In the same way as programming stimulated computational thinking (Papert 1980), we claim that mathematical simulations nurture mathematical thinking; the focus in both cases lies on human thought and development and not on technology.

Undoubtedly, these programs allow for mathematical activities such as exploration or checking special cases of conjectures (Borwein and Devlin 2008). These “experimental mathematics” (Borwein 2011) activities can be regarded as being equally important as “rigorous proving” in mathematics (Jaffe and Quinn 1993). However, we argue from a theoretical perspective that they can also provide new means to formulate proofs, that is mathematical arguments fulfilling the roles of proofs pointed out, for example, by de Villiers (1990) (see “[The Status of Proof in Mathematics and Mathematics Education](#)”).

Research on the impact of these new technologies on the concept of proof, which plays a central role in mathematics, is not very developed, despite their well-known impact on the understanding of mathematics in general (Hoyles and Lagrange 2010). Research is focusing more on the role technology plays in establishing conviction of a fact, which may be even counterproductive in justifying the necessity of proof (Marrades and Gutiérrez 2000; Hoyles and Noss 2003; Bolite Frant and Rabello de Castro 2000; Christou et al. 2004), and efforts to remediate this (Hadas et al. 2000; Jones 2000), see Sinclair and Robutti (2013) for an overview. This focus on technology establishing conviction is related to the use of many of these technologies primarily in educational contexts. In curricula, proofs are well-established almost only in geometry (Hanna and de Bruyn 1999; Stylianides 2007), where students do proofs by geometric constructions in the spirit of Euclid (Mogetta et al. 1999). Some go as far as to say that the central reason for the existence of geometry in the curricu-

¹ARIADNE is available for download at <https://imaginary.org/program/ariadne-drawing-topology>

lum is for serving as a paradigm for deductive proof (Hanna 1998). Overall, research investigating possible evolutions of the concept of proof when digital tools are used, is not well developed and focuses on their use for such constructions in Euclidean geometry (Marrades and Gutiérrez 2000).

Moreover, the effect of using digital geometry environments in educational contexts is controversial (Hanna 1998), as “proofs” in such environments are often seen as providing evidence for the truth of a statement by making available a large, by continuity seemingly infinite, number of examples (Hanna 1998), without actually being a *real* proof (Nam 2012). An example would be dragging the corners of a triangle while keeping track of the sum of the internal angles, which can convince a student of the fact that their sum is always 180° without giving any sort of explanation of this fact and its relation to the parallel axiom, and without allowing to be certain that no case exists, in which the sum is different. The scope of such software is thus often seen as limited to *exploration* (Christou et al. 2004) and *conjecturing* based on this exploration (Mogetta et al. 1999; Venema 2013).

This deeply rooted view on such technology as being there for exploration and conjecturing, and on proofs as being formal in their representation, led to attempts on doing proofs by integrating such technology and formal proofs in digital environments. This manifests, for example in, the implementation of proof assistants in dynamic geometry environments (Albano et al. 2019; Nam 2012; Kovács 2015; Miyazaki et al. 2017; Hanna et al. 2019).

We argue that these limitations and separation in informal/exploration – formal/proving is mostly due to the nature of geometrical constructions and the role of Dynamic Geometry Environment (DGE) software in proof and technology related activities. To emphasize that these limitations are not inherent to digital environments in general, but to the practice of use of those from geometry, we present several proofs² done in a different type of software, a Dynamic *Topology* Environment called ARIADNE (Sümmerrmann 2019a). In ARIADNE, the user cannot only explore basic concepts of topology including points, paths, and homotopies of paths, but can also formulate proofs of non-existence using invariants such as the winding number of a path. Examples of such proofs are given in Appendix A. They highlight the features of simulation-based proofs in such a digital environment, which are not limited to exploration, experiments, and constructions, but can satisfy the same functions as traditional proofs. In our argumentation, ARIADNE serves as an example; it can be exchanged for any simulation with similar characteristics.

Our first findings in this theoretical study are the necessary conditions that the broader class of environments and tools representing mathematics have to fulfill, in order to possibly allow the construction of proofs in such environments. These conditions define the new concept of a *mathematical simulation* and with it the term *simulation(-based) proofs*. An emphasis must be put on the word *necessary* in the previous formulation, as there are examples of mathematical simulations such as MATLAB or MAPLE, in which proofs that go beyond calculations are harder to achieve due to the focus on numerical manipulations.

²Examples can be found on Youtube (Sümmerrmann 2019b, c, d)

Following an analysis of this kind of simulation proofs, we present a classification of proofs along the lines of *interactivity* and *formality*, showing the place of simulation-based proof in the context of more traditional or alternative forms of proof. We also present an analysis of different functions of simulation proofs, following and extending the framework created for proofs in general by de Villiers (1990). This leads to the conclusion that simulation proofs are of particular interest in mathematics education, with some caveats.

Going beyond the familiar debate of the non-surveyability of computer-generated proofs (Tymoczko 1979) such as the proofs of the Four-Color Theorem and the Kepler conjecture, we introduce *technological reasons* as a category of acceptance criteria for proofs in the context of simulations.

Article Organization

Roughly speaking, we will give definitions, examples, and intricacies of the use of simulation-based proofs. Then follows a classification of simulation-based proof, first external (in relation to other types of proof) and then internal (highlighting functions of proof).

More precisely, in “[The Proof Process in a Mathematical Simulation](#)”, we will start by explaining more closely what we mean when talking about proofs. We then define the notion of a *mathematical simulation*, in particular distinguishing simulations from animations and microworlds. This is followed by a general analysis of the user interaction process with a simulation. As errors occurring in the interaction process are vital in establishing trust in the simulation, we give a categorization and examples of different types of error that may arise in the process.

This is followed by “[Dimensions of Proofs](#)”, in which various forms of proof and proving environments are classified along the dimensions of interactivity and formality. This aids in understanding the place of simulation proofs and mathematical simulations in relation to other forms of proof and mathematical environments.

Subsequently, “[Functions of Simulation Proofs](#)” employs de Villiers’ (1990) framework to discuss the functions of simulation proofs. In particular, this also encompasses new categories regarding the conviction of the proof recipient particular to simulation proofs.

Finally, “[Conclusion](#)” contains conclusions and implications for mathematics education and mathematics research, followed by some descriptions of simulation proofs in Appendix A.

The Proof Process in a Mathematical Simulation

The Status of Proof in Mathematics and Mathematics Education

While proofs are at the center of mathematics, their nature is highly contested and comprises a wide range of different objects. The spectrum of proofs begins with “formal proofs” in a mathematical logic theory sense, as chains of formalized deductions, which can at least in theory be checked by a computer. As Krabbe (2008) states, these

“are a logician’s gadget,” and do not exist in practice (Aberdein 2008). The view that formal proofs are practically non-existent is challenged by new generations of formal proof theories and resulting computational advances, leading to formalized proofs as feasible tools for mathematicians work (Voevodsky 2015), but it is still certainly true for “most” mathematicians and almost all of mathematics in educational contexts.

The next “step” are formal proofs in the sense of axiomatic, symbolic mathematics. Such proofs contain elements of an *argumentation* and are the type of proof that is most commonly employed in mathematics contexts. That mathematical proofs can be argumentations (Aberdein 2008), i.e. let room for debate, is again a controversial notion and rejected by some (Johnson 2012), as it conflicts with the view of proofs providing absolute conviction of truth *beyond doubt* (Krabbe 2008).

Proofs in educational settings, such as schools or undergraduate courses, are then again different, as they adhere to forms of reasoning and are communicated with forms of expression “that are valid and known to, or within the conceptual reach of, the classroom community” (Stylianides 2007). This can include other forms than the stricter representation in communications within the mathematical community.

What we propose, for the purpose of this article, is more of an implicit definition of proof, giving a list of attributes defining a proof in the sense of Lakoff (1987) and Weber (2014): By showing that simulation-based proofs can fulfill the roles and functions of proofs as specified by de Villiers (1990), they may be regarded as such. In this way, we adopt a view of proofs as a cluster concept in the sense of Weber (2014).

It should be noted that when we argue that “simulation-based argumentations” can be regarded as proofs, it is not our intent to somehow bypass the sociomathematical norms (in the sense of Yackel and Cobb 1996), and certainly not to set them; we cannot define externally what constitutes a proof, this has to be done by the mathematics community and time.

Mathematical Simulations

Generally, a *simulation* can be defined as any attempt to mimic a real or imaginary environment or system (Rieber 1996). Based on this, we define a *mathematical simulation* (MS) to be a simulation mimicking mathematics by following the *mechanisms of action* immanent to the mathematics being simulated, creating the representation as a consequence of general underlying rules (see Richter-Gebert 2013). A mathematical simulation comes equipped with a certain representation of the mathematical content, in general defined by the person who built the simulation, and the capability to allow user interaction with this representation, be it through manipulation of objects, images, symbols, or other such modes of representation (see “Formality”) yet to be conceived. We will call proofs based on a simulation either *simulation-based proofs* or simply *simulation proofs*.

Simulations are fundamentally different from animations, even if it may not be easy to distinguish them from another as a user. An animation is defined to be a software following a pre-determined stimulus response mechanism (Richter-Gebert 2013). While animations are defined by *what* is being presented, simulations are defined by *how* the presentation is made.

An example borrowed from physics may help to clarify the differences between animations and simulations. Two tablet apps are given, both allow the movement of a ball on the screen via dragging. If let go, the ball will fall to the ground, bouncing a few times. Behind the visual representation, this could be realized by either implementing the *mechanisms of action* of the ball, such as gravity and the spring force calculated from the kinetic energy. This would be a (physics) *simulation*. Another possibility would be to simply let the ball go down pixel by pixel until its coordinates reach a certain value, which then triggers a predefined movement along some curve, giving the “illusion” of the ball bouncing; this is an *animation*. Although they may not look different, at least at first sight, the simulation may allow the exploration of physical phenomena even beyond the intent of the programmer, while the animation does not. In the scenario of the bouncing ball, this may mean the deformation of the ball on impact, which may not have been intended by the programmer of the simulation but may nevertheless be observed, while it is only visible in the animation if the programmer has explicitly thought of it.

A mathematical example may be a software generating graphs of quadratic functions, in the case of a simulation really plotting the function, in the case of an animation presenting a predefined image from a collection of graphs best fitting the entered parameters. As the number of different graphs on a screen with a finite resolution is finite, the difference between the two cases would be virtually impossible to identify for a user.³

These examples showcase the predefined nature of animations, which certainly makes the construction of an animation easier than that of a simulation and may also have some advantages. By their very definition, they are, however, useless for the exploration or representation of mathematics beyond the build-in set of cases.

A term closely related to the one of a mathematical simulation is “microworld,” a concept introduced by Papert (1980) describing a subset of reality or a constructed reality whose structure matches that of a given cognitive mechanism so as to provide an environment where the latter can operate effectively (Papert 1980, p. 204), allowing learning. Hoyles et al. (1996) also stress the importance of *interactivity* for microworlds, stating “Software which fails to provide the learner with a means of *expressing mathematical ideas also fails to open any window on the processes of mathematical learning. A student working with even the very best simulation, is intent on grasping what the simulation is demonstrating rather than attempting to articulate the relationships involved.*” Hoyles et al. (1996, p. 54) [emphasis added].

Simulations can, however, be distinguished from these microworlds through design choices made in the design of the latter as to fit the environment to the learners’ cognitive state (Rieber 1996). The aims of a microworld go beyond those of a simulation as it not only represents a mathematical object faithfully, but also specifically strives to enable learning, making microworlds a subclass of simulations. In a microworld, the objects can be manipulated by the user “with the purpose

³Note that although most examples of animations or simulations that come to mind are visualizations, they do not have to be; any representation mode is possible.

of inducing or discovering their properties and the functioning of the system as a whole” (Edwards 1995, p. 144).

Mathematical simulations can serve as a framework for formulating proofs, as in addition to the capability of representing mathematics in a certain way, which can also be said for Proofs Without Words or even writing on a blank sheet of paper, the user can interact with the simulation in a meaningful way. The interaction is meaningful, as conjectures and arguments can be made in response to the behavior of the simulation, because the simulation reacts according to mathematical laws. Furthermore, this central feature of simulations makes it possible to not only replicate known results, but also to discover new results, as the grounding in underlying mathematical rules allows the user to go beyond the simulation designer’s imagination and intentions.

The simulation used to demonstrate some examples of simulation-based proofs in this paper is ARIADNE⁴, a dynamic topology environment developed by Sümmerrmann (2019a). In ARIADNE, the user can explore different mathematical spaces including points, paths, and homotopies of paths using touch gestures, constructing points by a touch, paths by dragging points, and homotopies by dragging paths. Mathematically speaking, the “dragging” feature of many DGEs represents a homotopy, so this exploration of spaces via homotopies goes to the core of this “most central tool of DGEs” (Sinclair and Robutti 2013).

The Interaction Process with a Mathematical Simulation

When arguing and possibly even proving based on a mathematical simulation, trust of the user in the simulation is crucial; for example, trust in the physics engine in the bouncing-ball simulation mentioned above. Errors in the interaction with a mathematical simulation could undermine the trust of the user. To determine the kind of errors that can arise in the interaction, we must analyze the interaction process with such a simulation. This analysis is itself independent of the purpose of the interaction, be it exploration or argumentation. The interaction process follows the same principles as the interaction with software in general. It can be described using a Subject-Object-Artifact triangle, derived from situated instrumented activity (Vérillon 2000); our model is specialized for the case of user interaction with mathematical simulations.

In an interaction process with a simulation, the users are in a cycle of adjusting their knowledge of mathematics to the observation of the simulation’s behavior, leading to the formulation of new actions to be taken, resulting in new output of the simulation, which is again observed by the users. Each step of the proof involving an interaction with the software leads to such an adjustment. The general nature of the relationship between the user, the software, and the underlying mathematics can be described using the triangle in Fig. 1.

⁴Although ARIADNE can also be seen as a microworld in the sense outlined above, we use it in its capability as a mathematical simulation.

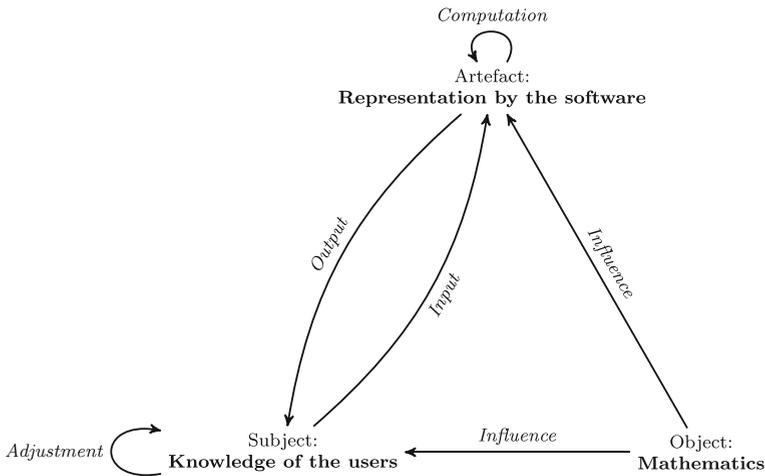


Fig. 1 An interaction process with the mathematical simulation

During the interaction process, the relations in the triangle are traversed by repeating several steps:

1. The users decide their next action as a consequence of **their knowledge** of mathematics (*influenced* by **mathematics**), taking the form of an *input* to the mathematical simulation.
2. The simulation internally *computes* a new **representation** as an *output*, *influenced* by the **mathematics** implemented in it.
3. The users interpret the new **representation** and *adjust* their expectations on the simulation's behavior and possibly also **their knowledge** of mathematics.
4. With new expectations in mind, based on the **representation** of the simulation, the users decide their next action, leading to a new *input* for the simulation.

Errors in the Interaction Process with the MS

For a productive use of mathematical simulations, the process described in “[The Interaction Process with a Mathematical Simulation](#)” must function properly. Different types of errors can disrupt this process, which are characterized in this section. We distinguish two categories: errors by the user and errors by the software. An error by the user means an incorrect interpretation of the representation of mathematics of the mathematical simulation, and an error by the simulation an incorrect representation of mathematics by the mathematical simulation.

It is, of course, impossible to determine what, in general, an incorrect representation of mathematics is. But the representation of mathematics in a mathematical simulation constitutes an agreement between user and software to the “language” used to carry information. So even if a representation generated by a simulation – like any representation – cannot be false by itself, it can be judged by its adherence to

the representation mode agreed upon and thus implicit to the simulation. In the following, the words “correct” or “error” will be used in this sense. In the triangle from Fig. 1, this error would be situated in “Representation by the software”.

Nevertheless, a mathematical error originating from interaction with the software can only be made by the user if the simulation’s dissenting representation leads to false assumptions about the represented mathematics. Notably, this implies that the error was observed by the user; if an error of the mathematical simulation has no consequences on the representation given as output or is overlooked, then it cannot have implications on the expectations and on the mathematical knowledge of the user.

In addition, the user cannot only hold correct or incorrect assumptions on mathematics, but also make errors in the interpretation of the simulation’s representation. The chance of misinterpretation may increase if the representation mode is not defined explicitly. In Fig. 1, this would correspond to errors in the arrows between artefact and subject.

This leads to the distinction shown in Table 1. The user can correctly or incorrectly interpret the representation by the mathematical simulation. In the same way, the mathematical simulation can render the representation agreed upon correctly or incorrectly. Even if all possible interactions may be situated in this grid, not all errors can be identified in this way; the user can also draw the right or wrong conclusions, corresponding to errors in the subject corner in Fig. 1, or the software can go beyond its representation capabilities.

To illustrate the types of errors given in Table 1, examples based on a fictional software are given. These simple examples do not represent proving situations, but suffice to showcase the different possible errors of user and software on the mathematical content.

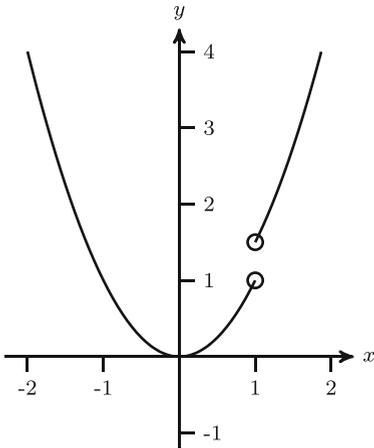
The software is a function plotter, which can represent a given polynomial function by drawing it in a coordinate system. It does so by randomly sampling points and connecting them by linear splines. It leaves a gap where the function is not defined. For all presented functions, the user wants to investigate singularities of the function using the software, disregarding the fact that a thoughtful user might not trust the software (compare “[Criterion 1: Trust in the Technology](#)”), or double-check the result by other means. The error types in Fig. 2 can then be described as follows.

- (a) The representation is correctly understood by the users: They see the singularity pointed out by the software.
- (b) The users have the correct assumptions about the representation used, but the software depicts an incorrect representation: It might be that the software did

Table 1 The different types of errors along two dimensions, accounting for representation-related errors

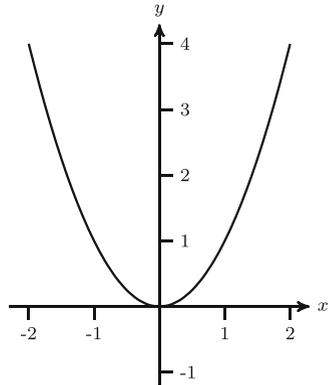
	MS representation	
User interpretation	Correct	Incorrect
Correct	Case (a)	Case (b)
Incorrect	Case (c)	Case (d)

$$f : \mathbb{R} \setminus \{1\} \rightarrow \mathbb{R}, f(x) = \begin{cases} x^2 & x < 1 \\ x^2 + 0.5 & \text{else} \end{cases}$$



(a) User interpretation correct, MS representation correct.

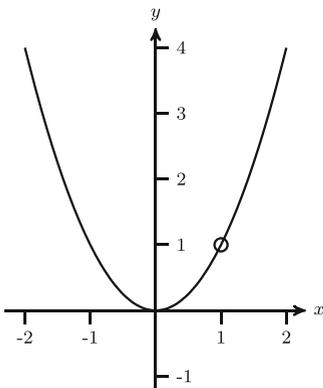
$$f : \mathbb{R} \setminus \{1\} \rightarrow \mathbb{R}, f(x) = x^2$$



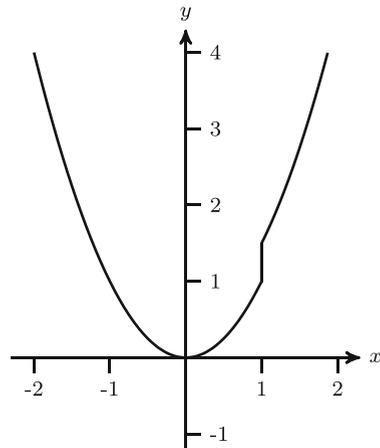
(b) User interpretation correct, MS representation incorrect.

$$f : \mathbb{R} \setminus \{1\} \rightarrow \mathbb{R}, f(x) = \begin{cases} x^2 & x < 1 \\ x^2 + 0.5 & \text{else} \end{cases}$$

$$f : \mathbb{R} \setminus \{1\} \rightarrow \mathbb{R}, f(x) = x^2$$



(c) User interpretation incorrect, MS representation correct.



(d) User interpretation incorrect, MS representation incorrect.

Fig. 2 The error types, as seen in a software for function plotting, according to Table 1

not include 1 in the approximation of the curve, and thus did not observe the singularity. The users “correctly” assume that the function does not have a singularity.

- (c) The users have incorrect assumptions about the representation used, but the software has no fault in the display: The user does not understand that the circled point is a singularity, even if removable, and believes the function to be free of singularities.
- (d) The users have incorrect assumptions about mathematics and the simulation has a fault in the display: The software displays a continuous line along the jump discontinuity, which is not the agreed representation for this type of singularity (correct representation in Fig. 2a). The users think such an abrupt and non-differentiable change of slope is a sign of a discontinuity, and interpret the function to have singularities at the points (0.99, 1) and (1.01, 1.5).

Dimensions of Proofs

Mathematical proofs can be classified along several dimensions, for example along the dimension of formality (Lakatos 1978). Besides this dimension, we concentrate on “interactivity” as a second dimension, to account for the focus on more dynamic proof forms, which are the focus of this article. The classification helps to distinguish the various currently existing environments for basing proofs upon and the corresponding proof, and to subsequently point out the place simulation-based proofs take in relation to these other forms of proof. We will also discuss the concept of “transferability” of proofs, which certainly plays an important role in this context. All these dimensions are not necessarily orthogonal, but represent a way of distinguishing some proofs or mathematical environments.

Formality

Formality describes the adherence to standardized mathematical notation in the representation of mathematics, which is distinct from “formal,” meaning written in a formalized language with formalized derivation rules (Krabbe 2008). Here, formal means adhering to a strict, structured symbolic representation.

Proofs, as all mathematical content, can be distinguished with regard to their representation. Different systems have been proposed to categorize representation modes, for example enactive, iconic, and symbolic by Bruner (1966), or somatic, mythic, romantic, philosophical, and ironical by Egan (1997).

We use the much coarser distinction into formal–informal, as this suffices to classify existing environments allowing proofs for the purposes of this article.

Interactivity

Interactivity is defined by the dictionary Merriam-Webster as “mutually or reciprocally active.” That means, not only has the medium to be active, such as in a video, but the user’s actions have consequences on the activity of the medium.

Proofs can present themselves at different levels of interactivity. One end of the spectrum are *static* proofs. These can range, in different levels of structuredness, from Proofs Without Words (Nelsen 1993), over “traditional” formal proofs from mathematical practice, such as proofs in textbooks, to proofs that can be checked by a computer, the “platonic ideal” (Lampert 2012).

More interactive proofs are given by videos or even animations, which may even be altered by tools such as sliders. These give the users some kind of control over the way the proof is presented to them, but does not constitute a simulation proof in the sense described in “[Mathematical Simulations](#)”.

At the other end of the spectrum are fully interactive proofs in mathematical simulations, such as the ones presented in ARIADNE. These are proofs in which the user has total control to alter the proof, albeit limited by the allowances of the simulation used.

Overview on representation modes of proofs

Several different types of environments in which proofs can be shown or done, such as digital environments or simply textbooks, are depicted in Fig. 3. The aim of this overview is to give a sense of the place mathematical simulations and with them simulation-based proofs take in comparison to other forms of proof-supporting environments. We will now give some more information about the objects referenced in the figure.

The standard type of proof is a text with symbols, sometimes accompanied by images for clarification or illustration purposes, such as in a standard textbook. Standard non-interactive proofs can, however, range from informal to formal. A very informal category are Proofs Without Words, where the image itself is the proof, only sometimes accompanied by text or symbols for clarification. Their status as to being a proof or representing an idea of a proof is not clear⁵.

A relatively new type of argumentations are so-called “Proofs Without Words 2.0” (Doyle et al. 2014), which are animated versions of Proofs Without Words, sometimes giving the user some control over the animation. This places them at the same level of formality as traditional Proofs Without Words, but more interactive, together with digital textbooks incorporating them such as Mathigon (<https://mathigon.org/>). Further up in the figure are environments that allow the user to manipulate the objects more freely and to come up with own proofs, such as DGEs or other mathematical simulations. These programs are mostly informal in their representation, which is, however, no requirement for a mathematical simulation.

On the more formal side of mode of representation are software such as SURFACE EVOLVER (Brakke 1992) or JUPYTER NOTEBOOK (Kluyver et al. 2016). SURFACE EVOLVER is a program to do operations, such as the simulation of geometric flows, on surfaces, working with a text-based interface. JUPYTER NOTEBOOK is

⁵In his famous collections of Proofs Without Words Nelsen (1993, 2000, 2015), Roger Nelsen writes in the introduction to Volume I: “Of course, ‘proofs without words’ are not really proofs”, while the introduction to Volume II, quite ironically, contains: “Of course, some argue that ‘proofs without words’ are not really proofs” concluding with a quote from Brown (1999) ending with: “In short, pictures can prove theorems”.

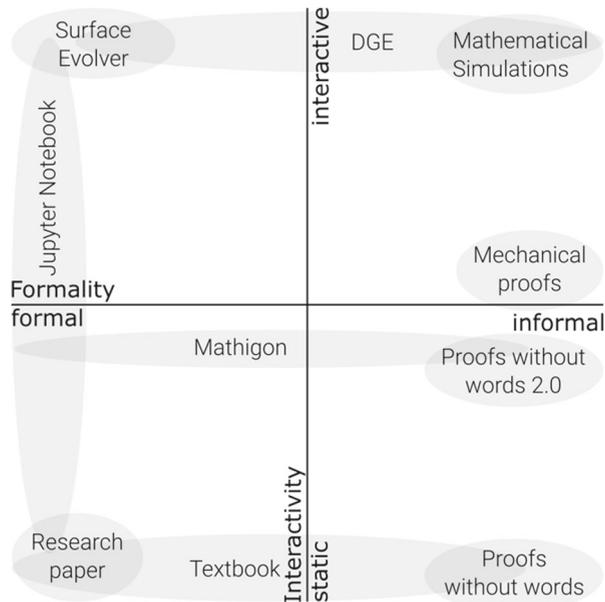


Fig. 3 Some proofs and environments for doing proofs sorted along the dimensions interactivity and formality, ranging from static to interactive and formal to informal, respectively. For examples of mechanical proofs, see Richard et al. (2019)

a web-based interactive computational environment, making it possible to mix text with programming language outputs of, for example, Python, making “seamless the communication between human and machine” (Barba 2015).

An environment, which is both highly interactive and formal, is also possible, for example in the form of proof assistants such as ISABELLE (Paulson n.d.) or LURCH (Carter and Monks 2013), or even general mathematical text editors, possibly giving another mode of distinction, which is, however, not the focus of this paper.

Transferability

While the idea behind a proof is in theory independent of its representation, it is nevertheless hard to do comparisons between proofs in different representation modes (Giaquinto 2020). It is even hard to compare proofs in the same representation: given two proofs of the same theorem, there is no formal notion of which proof is “simpler” (Cain 2019) (also compare Inglis and Aberdein’s (2014) discussion on the meaning of “simple” in mathematics). Also, not every proof can be represented in all modes of representation with the same ease, as many Proofs Without Words demonstrate; many would be challenging to formalize, and probably even lose their elegance.

There are, however, cases where the representation of a proof can be changed without changing the proof’s idea. We describe the degree to which this is possible and the

amount of “mental work” necessary to do so by *transferability*. This concept is certainly related to the concept of cognitive unity (Boero et al. 1996; Pedemonte 2007), but does not consider the transition from a person’s argumentation to a formal proof, but the comparison of the products of proving processes in general. It stands out of question that, based on this vague definition, transferability cannot easily be quantified. It is nevertheless an important concept for understanding the role of proofs in different modes of representation. The dimension of transferability is relevant in mathematics as well as mathematics education, as the current *de facto* representation of choice for proofs is formal-symbolic and all other representation modes are measured as for their alignment to this representation (Brunner and Reusser 2019).

Functions of Simulation Proofs

We follow de Villiers’ (1990) extension of a categorization by Bell (1976) to analyze the functions of simulation proofs. De Villiers highlights the functions of explanation, systematization, discovery, verification, and communication. We split up the function of *verification* in relative and absolute conviction, to emphasize the difference between believing the statement to have a high probability to be true, which is relative conviction, and believing the statement to be surely true, which is referred to as absolute conviction (Weber and Mejia Ramos 2015).

We identify several criteria particular to simulation-based proofs that influence the relative conviction function: *trust in the technology*, *level of detail*, and *limits of the representation*. These criteria add to the established ones for proofs in general as well as perhaps also to the criteria surrounding computer-based proofs.

Explanation

Explaining *why* a result holds is one of the main motivations for a formal proof and goes beyond the relative conviction *that* it holds. That relative conviction can be achieved by, for example, a lack of counterexamples after some searching, is also one of the criticisms of the use of dynamic geometry in educational settings; by being able to provide a large, seemingly infinite, number of examples, students are so convinced of the result they should prove that they no longer feel the need to prove it (Bolite Frant and Rabello de Castro 2000). In the framework of Harel (2013), the students draw certainty from an “inductive proof scheme” instead of the educationally desired “transformational proof scheme”.

On the other hand, numerous Proofs Without Words give examples of visual arguments explaining why a result holds. This shows that the capacity of a proof for explanation does not necessarily depend on the representation. “Proofs Without Words 2.0” denote animated versions of such proofs (Doyle et al. 2014), which can certainly be seen as being something between static visual proofs and simulation-based proofs, and retain the explanatory capacity of Proofs Without Words while being more interactive.

A mathematical simulation goes even further, giving the user more freedom to explore the phenomenon being reviewed and thus more opportunities to find an

explanation. There are many examples of explanations using dynamic geometry software, which show that the problem addressed by the criticism above should be directed more to the overemphasis of the relative conviction aspect of proofs in education (Hanna 1998), which leads to the question of “why” being overshadowed and thus neglected by students.

Relative Conviction

While the notion of “relative conviction” was only put forward in Weber and Mejia Ramos (2015), the connection between exploration of mathematical situations using technology and conviction through quasi-empirical testing of the truth of a result is a well-explored concept (see the examples and references in the introduction). It is the main reason for the use of simulations such as MATHEMATICA to check statements by computation in research, and the employment of dynamic geometry software in education. In such a setting, verification is often, especially in lower grades, understood as *quasi-empirical* verification providing relative conviction (Hanna 1998), as opposed to a deductive proof, which would provide absolute conviction. We believe that simulation proofs can go beyond this kind of relative conviction and also provide absolute conviction by being more than just a collection of examples, however compelling, but providing insights and explanations as to why a result holds.

Absolute Conviction

This aspect not only denotes the absolute conviction of the truth of the result, but also the conviction of the validity of the proof (which certainly implies the former). As there is no consensus on a definition of proof in mathematical practice, there are no generally accepted criteria for its validity (Hanna and Jahnke 1996). Proofs have evolved through history and range over a wide variety of type, and while there are some techniques of proof accepted by most, such as mathematical induction or *reductio ad absurdum*, there is no one proof type that fulfills all needs and demands of every mathematician (McAllister 2005). Thus, the question of the validity of a proof is a deeply subjective one, influenced by the ever-changing norms of the mathematical community (Sommerhoff and Ufer 2019). This sort of change of norms is not unique to mathematics; quantum theory offering no visualizations led to some physicists not accepting it, until overwhelming evidence forced physicists to reshape their criteria for theory acceptance, abandoning the need for visual representations of phenomena (McAllister 2005).

However, different criteria can be identified that may influence this absolute conviction to a varying extent. We propose that these be grouped into the following categories:

1. Mathematical-logical reasons, the personal understanding of the theorem, and an *a priori* judgment of its validity; the logical consistency of the proof.
2. Socio-cultural reasons, such as the trust in the author or in the mathematical community that examines the proof.
3. Technological reasons.

The first two categories contain established criteria for proof acceptance (Sommerhoff and Ufer 2019; Yackel and Cobb 1996; Hilbert 1931) and are not the focus of this paper.

The third category of technological reasons is of particular importance for proofs in the context of computers, including mathematical simulations. Several criteria of this category can be identified.

Criterion 1: Trust in the Technology

If a proof involves the work of a computer, then trust in the computer may be a factor in accepting the proof. This is a multi-faceted aspect, which encompasses several subcategories.

It can mean the trust in the operations done by a computer, such as in computer-assisted proofs. This can further be elaborated, as, for example, the original Appel-Haken proof of the Four-Color Theorem not only needed the trust in the operations executed, but also in the validity of the computer program generating the proof itself. This was eliminated not until decades later with a formal proof in Coq (The Coq Development Team 2019), reducing the trust to the Coq system and the computer operations (Gonthier 2008).

The main problem with computer-based proofs, such as computer-assisted proofs, is often their length, which makes them *non-surveyable*, that is inaccessible for verification to the mathematical community (Tymoczko 1979). But even if a proof seems surveyable, its acceptance by the recipient can still require trust in the technology, as the following example demonstrates.

If users drag a triangle in a DGE, the users most probably have the *expectation* that only the position of the triangle should change, not its area or other intrinsic features. Each time they perform the dragging operation, they can perform a visual inspection and check if the triangle has changed, strengthening their trust that dragging is a faithful representation of this translation they have in mind. They could further strengthen this trust by looking at the source code of the application, if accessible.

In some points, the users will check against their expectations and mathematical knowledge if the simulation is indeed one, that is if the mathematics are accurately represented. At other times, they will trust the software to be honest with them, and will adjust their expectations and mathematical knowledge according to their interpretation of the visualization given by the software.

Criterion 2: Level of detail

In a traditional paper-and-pencil proof, every argument can be broken down further, until the steps performed are small enough to be understood by the recipient or the axioms are reached. This strengthens the trust of the users in a proof, as they have the certainty that they could, in theory, probe every part of it.

In a digital environment, such as a mathematical simulation, the detail of a proof is limited by the resources the software provides. In addition to the interference of the trust, and with it the conviction of the users in the proof, this massively inhibits the users in their own creativity, as they, opposed to the software designer, are not able to

construct their own objects independently of the software's functionality, a limitation that can hopefully be overcome by future simulations.

A prime example is the “winding number tool” in ARIADNE, which provides a way of computing the winding number of a path around a point. By doing this in a certain way, it not only inhibits the users to think of their own way to compute the winding number of a path, but also which invariant of a path to choose in the first place. Therefore, proving in such an environment is always only proving given these constraints.

Criterion 3: Limits of the representation

Every representation mode has its limits, which are independent of the technology used. This is a problem in some types of proof such as visual Proofs Without Words that can hamper proof acceptance (Bardelle 2010). This factor is magnified in dynamic proofs in digital environments, as the representations of mathematical objects here are not directly generated by the prover, i.e. the mathematician. They are thus even more susceptible to error, as the possibility of a simulation going beyond the capabilities of its representation always exists (compare “[Errors in the Interaction Process with the MS](#)”). The users know this, which can impede their relative conviction.

Instances of these limits are found readily, for example given a mathematical simulation plotting a function with a removable singularity such as $f : \mathbb{R} \setminus \{2\} \rightarrow \mathbb{R}$, $x \mapsto x^2$. A programmer would have had to think of a way to highlight this singularity, for example by drawing a small circle at $(2, 4)$, as no resolution would be sufficiently fine to show its existence. In such a way, the programmer must have thought about the representation of *every possible* function that might be plotted by the software.

The more general and powerful the simulation, the more cases have to be covered in beforehand. In the light of the research-level proving process, one is certainly in the realm of new mathematical objects and connections between them, which makes it all but impossible to construct the environment to account for all possible cases. Herein lies the special challenge for the designers of such mathematical simulations, but also for the user interacting with the software, who has to take this type of limitation into account.

Systematization

Systematization describes the organization of several results into a (deductive) system. This is certainly made easier by using the standardized formal Bourbaki-style notation ubiquitous in mathematics today, allowing the description of results from different fields in the same language.

For a specialized piece of software such as a simulation designed to simulate a certain part of mathematics, the systematization of results from different fields is certainly hard to achieve. If the simulation uses visualization as its mode of representation, this may add to this problem, as at least today, generality and power of a simulation stand opposed to its intuitiveness and informality. This means that a visual simulation often tries to incorporate an informal and intuitive interface, which

then limits its generality. Hopefully, future simulations will be able to overcome this limitation.

One could say that systematization is thus a weak point of a mathematical simulation.

Local systematization on the other hand is, however, very much possible; the organization of several definitions, lemmas, and theorems in one (sub-)field being more homogeneous in their representation. An example is a function plotter, which can very well classify different functions such as polynomials using their coefficients, or ARIADNE systematizing paths on the plane by their homotopy classes, showing the relation between theorems on these objects.

Discovery

Discovery of new results is certainly not limited to proofs, but is an aspect relevant to proofs. The historical example of sphere eversions illustrates a discovery process in mathematics.

The eversion of the sphere is a regular homotopy turning \mathbb{S}^2 in \mathbb{R}^3 inside out. Constructions of such eversions were contrived by, for example, Shapiro (Levy and Thurston 1995), Morin (turned into a video by Max 1977), and Thurston (featured in a movie (Levy et al. 1994)⁶).

Visualization in itself can bring new ideas into mathematical research (Bartzos et al. 2018). Now imagine a software had been available to these mathematicians, making possible the deformation of manifolds shown in the videos by directly controlling their manipulation. It is quite possible that the construction of such an eversion would have been more accessible. This is even more plausible as the discovery of the later eversions correlates with the expanded use of visualization software.

In education, one of the main uses of dynamic geometry software is to facilitate the forming of conjectures in students by analyzing specific mathematical settings given by the teacher (Mogetta et al. 1999). This is certainly a part of the discovery aspect of proofs; by investigating arguments for the validity of statements, new statements are conjectured. Furthermore, there is no reason to believe this feature is only of use in education, given a software capable of representing objects of interest to research-level mathematicians, it would certainly be used to discover new results.

Communication

It is a central function of a proof to communicate mathematical ideas. In a simulation, this function encompasses both the communication with the mathematical community and the communication with the software itself. In both situations, the software provides a medium to express thought, which is strongly dependent of the representation form of the proof.

⁶There is even a more interactive animation available at <http://profs.etsmtl.ca/mmcguffin/eversion/>.

The most used form to communicate proofs is by large a static and formal representation. To quote Mazur (2014) “As with almost all advances, something was lost,” talking about losing “the public” (i.e. non-mathematicians) by this “code, unintelligible for the uninitiated.” However, Victor (2014) argues that even more was lost: The use of formal notation restricts many human capabilities, forcing humans to use only a part of their cognitive abilities as they are confined to sitting in front of a small screen or piece of paper, manipulating it indirectly with a pen or a mouse creating static objects.

Simulation-based proofs may create the possibility to externalize thoughts in a way closer to the way we think. Touch- or gesture-based interfaces can express argumentation in a more *embodied* fashion, reacting dynamically to human input (Abrahamson and Bakker 2016). While this may also be seen as an economical argument as such an interface may allow “more” work to be done, the idea is to achieve more *humane* means of communication.

Mathematical simulations also allow, using the internet, a collaborative form of proving, by being able to manipulate objects simultaneously (Borba et al. 2017). While this has been a standard way of working in mathematics for a long time, this collaboration can be now made independent of physical restraints.

A mode of representation often chosen by simulations is a visual one, which has several influencing factors. Visualizations have a long history in mathematics, and in the acceptance of theories in general, having been seen a prerequisite or at least necessary component of a “proof” (von Fritz 1955).⁷ Also, visual representation modes of mathematics may be “closer” to the way we think, as mathematical thinking more often deals with images than with formulas, at least in some areas of mathematics (Hadamard 1954). Hopefully, future simulations can unite the advantages of visual embodied communication with the power, precision, and universality of formal-symbolic representation modes.

Conclusion

This article points out the role mathematical simulations can play in the context of proving. As this connection is not yet well explored and as proving is a core activity in mathematics, this has far-reaching implications for mathematics educational practice and research, as well as for mathematics research itself.

These implications require, to some extent that the presented simulation-based proofs can indeed be regarded as proofs. This issue can surely not be resolved in this general formulation, but will depend on the same criteria other proofs need to fulfill as well, such as their exact form and implementation, and the context of their use.

⁷See Giaquinto (2020), citing Gauss: “anybody who is acquainted with the essence of geometry knows that [the logical principles of identity and contradiction] are able to accomplish nothing by themselves, and that they put forth sterile blossoms unless the fertile living intuition of the object itself prevails everywhere.”

Implications for Mathematics Education Practice and Research

The development of new technologies and the programming of new software is changing the educational landscape. Decades ago, calculators replaced slide rules and logarithm tables in classrooms. Years ago, DGE and CAS have revolutionized teaching. Their impact on proof has been to give a preliminary exploratory and quasi-empirical step to increase relative conviction regarding a conjecture, without influencing the *actual* proof, the content and form of which has not changed. Now, with more mathematical simulations such as ARIADNE appearing, the form of the actual proof is challenged. This makes it all the more important to do research in this area.

As simulation-based proofs can fulfill all functions of proof, and can do so without the at times inhibiting factor of formulaic representation, they are of particular interest for educators in undergraduate courses. Here, the restrictions of creativity through design choices of the software (compare “[Criterion 2: Level of detail](#)”) can even be thought of as a feature, as such restriction can constitute an aid in doing proofs by narrowing the number of choices available. Note that it is not the purpose of simulation-based proofs, nor is it the question addressed in this paper, if simulation-based proofs are to be considered as “real” proofs, in the sense of being a perfect substitute for traditional symbolic proofs. Furthermore, such a discussion cannot be settled by theoretical debate, as the status of proofs depends on the norms of the community (compare “[The Status of Proof in Mathematics and Mathematics Education](#)”). Such proofs can rather be a gateway into proving, giving an alternative *access* to proofs in a non-formal highly interactive setting. They may guide learners on their transition from mental argumentations in the sense of Mamona-Downs and Downs (2010) to formal proofs by giving them an appropriate environment to project their thoughts upon, maybe with the guiding rails of the affordances of the environment.

Examples may be the introduction of the concept of continuity through a suitable mathematical simulation software, for example giving a visual representation of the ϵ - δ definition of continuity, maybe along the lines of Fig. 4. This software might then be used to prove the continuity or non-continuity of some functions. However, as described in “[Mathematical Simulations](#)” and “[Discovery](#)”, the software must be powerful enough to allow real discovery by being more than just an animation, but a mathematical simulation.

There may be an obstruction to the implementation of simulation-based proofs in educational settings; the absence, at times, of transferability or “parallelism” (Miller 2012) to formal representation also implies a deficiency of connectivity with further study or even other fields of mathematics. In an educational setting, visual proofs or arguments are mostly made to support a formal proof, which currently is the gold standard in mathematics, and aid in its understanding. A visual proof thus has to be transferable into a formal proof to a certain degree, which may not always be possible (compare “[Transferability](#)”). It is however possible, with future development in mathematics visualization that this can be made more balanced or even reversed; a proof will first be provided visually and translated into formal proof as an addition, for example for automated verification purposes.

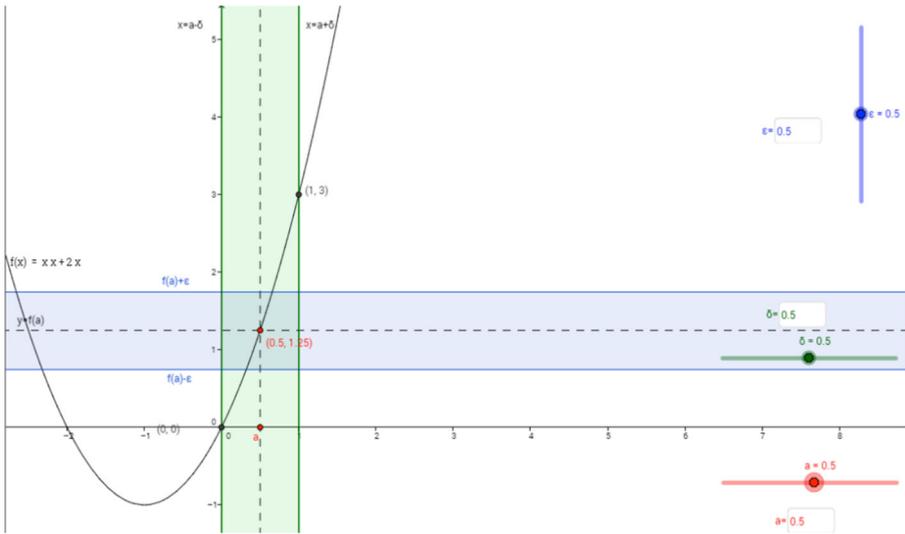


Fig. 4 A GEOGEBRA-Applet letting the user explore the uniform continuity of some functions (Dikovic 2017)

Following the distinction into “learning to argue” and “arguing to learn” (Baker et al. 2019), simulation-based proofs may not only be used to learn mathematics by proving in a mathematical simulation, but also for inciting a discussion with students on the nature of proof. The debate on whether, or to which extent, the arguments made in a simulation constitute a proof, or how visual arguments relate to proofs in general in the sense of the above discussion, can be used to foster the understanding of the concept of proof.

The consideration of simulation proofs also has implications for mathematics education research. Approaching the subject from the students point of view, researchers investigating ways of teaching proofs or the learning of proofs may consider the use of simulations for either purpose. The implications of their use might then be assessed for their influence on the beliefs on the nature of proof. From the technological side, researchers working on the assessment or development of learning software might be driven to consider existing software for use in learning and teaching of proofs. For educational designers, requisites for the design of such environments must also be established.

Implications for Mathematics Research

Visual proofs are playing an ever larger role in research (Bartzos et al. 2018). As the possibilities of representing mathematics in computer environments as well as the possibilities of interacting with computer-generated content continue to expand,

this progress will surely sustain. We believe that simulation-based proofs may be a bridge between traditional proofs and experimental mathematics, combining the deductiveness of the former and the explorational capacities of the latter.

As this is highly dependent on available software, which is arguably harder to develop for research-level mathematics than for undergraduate or school mathematics in general, the long-term developments in this area are hard to predict.

Outlook

While simulation-based proofs can in principle already fulfill all functions of proofs, we are still at the beginning of the digital age, so many more changes are to be expected. Further advances both in technology as well as representations will hopefully lead to simulations vastly more powerful while still being intuitive to use, realizing many of the features outlined in this article. This will undoubtedly lead to a shift in the use of technology away from purely exploratory capacities to other areas of mathematics practice. Research is and will be needed to understand which areas, such as proofs or problem posing and solving, these are, and how they are affected.

As remarked in “[Absolute Conviction](#)”, proof is not a static concept, but shaped by the community. It would be interesting to examine the opinions of research mathematicians as well as educators on the status of simulation proofs as acceptable proofs, in mathematics as a scientific discipline and as an item in the curriculum, and likely also identify further factors influencing their opinion.

Funding Open Access funding enabled and organized by Projekt DEAL.

Appendix A: Examples of proofs in ARIADNE

Theorem 1 *Given the plane with one puncture $\mathbb{R}^2 \setminus \{0\}$, there exists a path γ with $W_\gamma(0) = 1$.*

Proof See Sümmerrmann (2019b): □

Theorem 2 *Given the plane with one puncture $\mathbb{R}^2 \setminus \{0\}$, the path γ with $W_\gamma(0) = 1$ is not null-homotopic.*

Table 2 The proof of Theorem 1

	Step of proof	In ARIADNE
1	Designate a point.	Touch the canvas
2	Construct a path. γ	Drag the point around somehow, such that it does not stay in one quadrant relative to the puncture, and then back to itself.
3	Compute the winding number of γ .	Click on winding number, then on γ , then on the puncture.

Table 3 The proof of Theorem 2

	Step of proof	In ARIADNE
1	Construct a path δ .	Drag the point around somehow, such that it stays in one quadrant relative to the puncture, and then back to itself.
2	Compute the winding number of δ .	Click on winding number, then on δ , then on the puncture.
3	Compute the winding number of homotopic paths constructed by dragging δ .	Drag δ around and compute the winding number of the dragged path. If necessary, do this first for small dragging steps, and then for larger ones; see that larger drags are “made up” of smaller ones.
4	Conclude that the winding number is invariant under homotopy.	By understanding the construction of the winding number, and seeing how the dragging of paths influences this computation.
5	Show that δ is null-homotopic.	Drag the path to the puncture.
6	Construct another path γ around the puncture.	Drag the point around the obstacle.
7	Compute the winding number of γ .	Click on winding number, then on γ , then on the puncture.
8	As the two winding numbers differ, conclude that γ is not homotopic to δ .	By invariance of the winding number under homotopy.
9	Conclude that γ is not null-homotopic.	As a null-homotopic path has winding number zero.

This is a theorem asking for a proof of non-existence, as it postulates there cannot exist a homotopy between the path with winding number 1 and the constant path.

Proof See Sümmerrmann (2019c): □

The arguably most tricky step is Step 3, where the user has to convince herself/himself of the invariance of the winding number under homotopies. This seems to be a situation as described in Hanna (1998), but is decidedly different.

Hanna describes a student who convinces himself of the truth of a theorem, that the perpendicular bisectors of a triangle intersect at a point, by dragging the corners of the triangle, thereby deforming it, while the bisectors always intersect (technically speaking a homotopy Lehrer and Chazan 2012). Here, Hanna argues that the student does not learn *why* this theorem holds.

We argue that the student has no chance to see why the theorem holds as the *construction* of the perpendicular bisectors does not give the same insight as their *defining property*: that all points on the bisector have equal distance to the endpoints of the bisected line segment.

In ARIADNE, the winding number $W_\gamma(p)$ of a path γ around a point p is not given as the result of a calculation invisible to the user, but as the result of a construction following the definition. So, when users drag a path and recompute its winding

Table 4 The proof of Theorem 3

	Step of proof	In ARIADNE
1	Designate a point.	Touch the canvas
2	Construct a path with winding number 1, possible by Theorem 1. γ	As detailed in the proof for Theorem 1.
3	Redraw the same path to obtain a new path δ .	Drag the starting point of γ along γ .
4	Concatenate the two paths, obtaining a new path ϵ .	First touch γ , then δ , touching both simultaneously.
5	Compute the winding number of ϵ . If necessary, repeat from Step 3, concatenating a new path to ϵ .	Click on winding number, then on ϵ , then on the puncture.
6	Conclude that concatenating γ with a path increases its winding number by one.	Observe the construction of the winding number.

number, they can see *how* the dragging of the path changed the computation of the winding number. This enables the generalization to the fact that the winding number is null-homotopic. This is an example of a use of a mathematical simulation in contrast to the construction above, as the users gain insight into the mechanism of action behind the winding number, and thus understanding and not just conviction.

Theorem 3 *Given the plane with one puncture $\mathbb{R}^2 \setminus \{0\}$. If γ is the path with $W_\gamma(0) = 1$, then $W_{\gamma^n}(0) = n$ for all $n \geq 1$.*

Proof See Sümmerrmann (2019d): □

This proof relies on the user seeing how the winding number is computed. As the winding number is computed iteratively by analyzing how the path changes locally, the concatenation of paths results in the addition of winding numbers. This naturally and inductively generalizes to any number of paths n .

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aberdein, A. (2008). Mathematics and Argumentation. *Foundations of Science*, 14(1-2), 1–8. <https://doi.org/10.1007/s10699-008-9158-3>.

- Abrahamson, D., & Bakker, A. (2016). Making sense of movement in embodied design for mathematics learning. *Cognitive Research: Principles and Implications*, 1(1), 33. <https://doi.org/10.1186/s41235-016-0034-3>.
- Albano, G., Dello Iacono, U., & Mariotti, M.A. (2019). A computer-based environment for argumenting and proving in geometry. In Jankvist, U.T., van den Heuvel-Panhuizen, M., Veldhuis, M. (Eds.) *Eleventh Congress of the European Society for Research in Mathematics Education*, Vol. TWG04. Utrecht: Freudenthal Group. Retrieved from <https://hal.archives-ouvertes.fr/hal-02402096>.
- Appel, K., & Haken, W. (1977). Every planar map is four colorable. Part I: Discharging. *Illinois J. Math.*, 21(3), 429–490. <https://doi.org/10.1215/ijm/1256049011>.
- Baker, M.J., Andriessen, J., & Schwarz, B.B. (2019). Collaborative Argumentation-Based Learning. In Mercer, N., Wegerif, R., Major, L. (Eds.) *The Routledge International Handbook on Dialogic Education* (pp. 76–88): Routledge.
- Barba, L.A. (2015). Computational Thinking and the Pedagogy of Computable Content. Retrieved from <https://youtu.be/JaLIWbID530> (BIDS Data Science Lecture Series).
- Bardelle, C. (2010). Visual Proofs: An Experiment. In Durand-Guerrier, V., Soury-Lavergne, S., Arzarello, F. (Eds.) *Proceedings of CERME 6* (pp. 251–260). Lyon.
- Bartos, E., Borrelli, V., Denis, R., Lazarus, F., Rohmer, D., & Thibert, B. (2018). An explicit isometric reduction of the unit sphere into an arbitrarily small ball. *Foundations of Computational Mathematics*, 18(4), 1015–1042. <https://doi.org/10.1007/s10208-017-9360-1>.
- Bell, A.W. (1976). A study of pupils' proof-explanations in mathematical situations. *Educational Studies in Mathematics*, 7(1), 23–40. <https://doi.org/10.1007/BF00144356>.
- Boero, P., Garuti, R., Lemut, E., Mariotti, A.M. (1996). Challenging the traditional school approach to theorems: A hypothesis about the cognitive unity of theorems. In Puig, L., & Gutierrez, A. (Eds.) *Proceedings of the 20th Conference of the International Group for the Psychology of Mathematics Education*, (Vol. 2 pp. 113–120).
- Bolite Frant, J., & Rabello de Castro, M. (2000). Proofs in geometry: Different concepts build upon very different cognitive mechanisms. In *Paper presented at the ICME 9, TSG12: Proof and Proving in Mathematics Education*. Tokyo.
- Borba, M.C., Askar, P., Engelbrecht, J., Gadanidis, G., Llinares, S., Aguilar, M.S. (2017). Digital technology in mathematics education: Research over the last decade. In Kaiser, G. (Ed.) *Proceedings of the 13th International Congress on Mathematical Education* (pp. 221–233). Cham: Springer International Publishing.
- Borwein, J., & Devlin, K. (2008). *The computer as crucible: An introduction to experimental mathematics*: AK Peters/CRC Press.
- Borwein, J.M. (2011). Exploratory experimentation: Digitally-assisted discovery and proof. In *New ICMI Study Series* (pp. 69–96): Springer Netherlands.
- Brakke, K.A. (1992). The surface evolver. *Experimental Mathematics*, 1(2), 141–165.
- Brown, J.R. (1999). *Philosophy of mathematics: An introduction to the world of proofs and pictures*: Routledge.
- Bruner, J.S. (1966). *Toward a Theory of Instruction*: Belknap Press of Harvard University.
- Brunner, E., & Reusser, K. (2019). Type of mathematical proof: personal preference or adaptive teaching behavior?. *ZDM*, 51(5), 747–758. <https://doi.org/10.1007/s11858-019-01026-y>.
- Cain, A.J. (2019). Visual thinking and simplicity of proof. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 377(2140), 20180032. <https://doi.org/10.1098/rsta.2018.0032>.
- Carter, N.C., & Monks, K.G. (2013). Lurch: a word processor that can grade students' proofs. In *MathUI, OpenMath, PLMS and ThEdu workshops and work in progress at the conference on intelligent computer mathematics*.
- Christou, C., Mousoulides, N., Pittalis, M., Pitta-Pantazi, D. (2004). Proofs through exploration in dynamic geometry environments. *International Journal of Science and Mathematics Education*, 2(3), 339–352. <https://doi.org/10.1007/s10763-004-6785-1>.
- de Villiers, M.D. (1990). The role and function of proof in mathematics. *Pythagoras*, 24, 17–24.
- Dikovic, L. (2017). Understanding and Visualization of the Uniform Continuity of Functions. In *International Conference on Advanced Technologies Enhancing Education*, (Vol. 68 pp. 6–9): Atlantis Press.
- Doyle, T., Kutler, L., Miller, R., Schueller, A. (2014). Proofs Without Words and Beyond. *Convergence*. <https://doi.org/10.4169/convergence20140801>.

- Edwards, L.D. (1995). Microworlds as Representations. In diSessa, A.A., Hoyles, C., Noss, R., Edwards, L.D. (Eds.) *Computers and Exploratory Learning* (pp. 127–154). Berlin: Springer.
- Egan, K. (1997). *The Educated Mind: How Cognitive Tools Shape Our Understanding*. Chicago: University of Chicago Press.
- Giaquinto, M. (2020). The Epistemology of Visual Thinking in Mathematics. In Zalta, E.N. (Ed.) *The Stanford Encyclopedia of Philosophy*. Spring 2020: Metaphysics Research Lab, Stanford University.
- Gonthier, G. (2008). Formal proof—the four-color theorem. *Notices of the AMS*, 55(11), 1382–1393.
- Hadamard, J. (1954). *An essay on the psychology of invention in the mathematical field*: Dover Publications.
- Hadas, N., Hershkowitz, R., Schwarz, B.B. (2000). The role of contradiction and uncertainty in promoting the need to prove in dynamic geometry environments. *Educational Studies in Mathematics*, 44(1), 127–150. <https://doi.org/10.1023/A:1012781005718>.
- Hanna, G., & Jahnke, H.N. (1996). Proof and Proving. In *International Handbook of Mathematics Education* (pp. 987–1033): Springer Netherlands.
- Hanna, G. (1998). Proof as Explanation in Geometry. *Focus on Learning Problems in Mathematics*.
- Hanna, G., & de Bruyn, Y. (1999). Opportunity to learn proof in Ontario grade twelve mathematics texts. *Ontario Mathematics Gazette*, 37, 23–29.
- Hanna, G., Reid, D.A., de Villiers, M. (2019). Proof technology: Implications for teaching. In *Proof Technology in Mathematics Research and Teaching* (pp. 3–9): Springer International Publishing.
- Harel, G. (2013). Intellectual Need. In *Vital Directions for Mathematics Education Research* (pp. 119–151): Springer New York.
- Hilbert, D. (1931). Die Grundlegung der elementaren Zahlenlehre. *Mathematische Annalen*, 104(1), 485–494. <https://doi.org/10.1007/BF01457953>.
- Hoyles, C., & Lagrange, J.-B. (Eds.) (2010). *Mathematics Education and Technology – Rethinking the Terrain*. Berlin: Springer.
- Hoyles, C., Noss, R., Noss, R. (1996). *Windows on Mathematical Meanings*. Netherlands: Springer. Retrieved from https://www.ebook.de/de/product/2458424/celia_hoyles_richard_noss_r_noss_windows_on_mathematical_meanings.html.
- Hoyles, C., & Noss, R. (2003). What can digital technologies take from and bring to research in mathematics education?. In Bishop, A.J., Clements, M.A., Keitel, C., Kilpatrick, J., Leung, F.K.S. (Eds.) *Second International Handbook of Mathematics Education* (pp. 323–349). Netherlands: Springer.
- Inglis, M., & Aberdein, A. (2014). Beauty Is Not Simplicity: An Analysis of Mathematicians' Proof Appraisals. *Philosophia Mathematica*, 23(1), 87–109. <https://doi.org/10.1093/philmat/nku014>.
- Jaffe, A., & Quinn, F. (1993). "Theoretical mathematics": Toward a cultural synthesis of mathematics and theoretical physics. *Bulletin of the American Mathematical Society*, 29(1), 1–14. <https://doi.org/10.1090/s0273-0979-1993-00413-0>.
- Johnson, R.H. (2012). *Manifest rationality: A pragmatic theory of argument*: Routledge.
- Jones, K. (2000). Providing a Foundation for Deductive Reasoning: Students' Interpretations when Using Dynamic Geometry Software and Their Evolving Mathematical Explanations. *Educational Studies in Mathematics*, 44(1), 55–85. <https://doi.org/10.1023/A:1012789201736>.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In Loizides, F., & Schmidt, B. (Eds.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90).
- Kovács, Z. (2015). Computer based conjectures and proofs in teaching euclidean geometry. Ph. D. thesis, Johannes Kepler University, Linz.
- Krabbe, E.C.W. (2008). Strategic Maneuvering in Mathematical Proofs. *Argumentation*, 22(3), 453–468. <https://doi.org/10.1007/s10503-008-9098-7>.
- Lakatos, I. (1978). What does a mathematical proof prove?. In Warrell, J., & Currie, G. (Eds.) *Mathematics, Science, and Epistemology* (pp. 61–69): Cambridge University Press.
- Lakoff, G. (1987). *Women, Fire and Dangerous Things*. Chicago: University of Chicago Press. Retrieved from https://www.ebook.de/de/product/3237968/george_lakoff_women_fire_and_dangerous_things.html.
- Lampert, L. (2012). How to write a 21st century proof. *Journal of Fixed Point Theory and Applications*, 11(1), 43–63.
- Legner, P. Mathigon. <https://mathigon.org/>.

- Lehrer, R., & Chazan, D. (2012). *Designing learning environments for developing understanding of geometry and space*: Routledge.
- Levy, S., Maxwell, D., Munzner, T. (1994). *Outside In*. Wellesly: AK Peters.
- Levy, S., & Thurston, W.P. (1995). *Making waves: A guide to the ideas behind outside in*. MA: AK Peters Wellesley.
- Mamona-Downs, J., & Downs, M. (2010). Necessary realignments from mental argumentation to proof presentation. In Durand-Guerrier, V., Soury-Lavergne, S., Arzarello, F. (Eds.) *Proceedings of the Sixth Congress of the European Society for Research in Mathematics Education* (pp. 2336–2345): Institut National De Recherche Pédagogique.
- Marrades, R., & Gutiérrez, A. (2000). Proofs produced by secondary school students learning geometry in a dynamic computer environment. *Educational Studies in Mathematics*, 44(1), 87–125. <https://doi.org/10.1023/A:1012785106627>.
- Max, N. (1977). *Turning a Sphere Inside Out*: International Film Bureau.
- Mazur, J. (2014). *Enlightening Symbols: A Short History of Mathematical Notation and Its Hidden Powers*: Princeton University Press.
- McAllister, J.W. (2005). Mathematical Beauty and the Evolution of the Standards of Mathematical Proof. In Emmer, M. (Ed.) *The Visual Mind II*: MIT Press.
- Miller, R.L. (2012). *On Proofs Without Words*: Master's Thesis, Whitman College. Retrieved from <https://www.whitman.edu/Documents/Academics/Mathematics/Miller.pdf>.
- Miyazaki, M., Fujita, T., Jones, K., Iwanaga, Y. (2017). Designing a web-based learning support system for flow-chart proving in school geometry. *Digital Experiences in Mathematics Education*, 3(3), 233–256. <https://doi.org/10.1007/s40751-017-0034-z>.
- Mogetta, C., Olivero, F., Jones, K. (1999). Designing dynamic geometry tasks that support the proving process. *Proceedings of the British Society for Research into Learning Mathematics*, 19, 97–102.
- Nam, N. (2012). The development of the proving process within a dynamic geometry environment. *European Researcher*, 32, 1731–1744.
- Nelsen, R.B. (1993). *Proofs Without Words: Exercises in Visual Thinking*: MAA.
- Nelsen, R.B. (2000). *Proofs Without Words II: More Exercises in Visual Thinking*: The Mathematical Association of America.
- Nelsen, R.B. (2015). *Proofs Without Words III: Further Exercises in Visual Thinking*: Mathematical Association of America.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*, 1st edn., Basic Books.
- Paulson, L.C. (n.d.) Introduction to Isabelle [Computer software manual]. <http://isabelle.in.tum.de/doc/intro.pdf>.
- Pedemonte, B. (2007). How can the relationship between argumentation and proof be analysed?. *Educational Studies in Mathematics*, 66(1), 23–41. <https://doi.org/10.1007/s10649-006-9057-x>.
- Richard, P.R., Venant, F., Gagnon, M. (2019). Issues and Challenges in Instrumental Proof. In Hanna, G., Reid, D., de Villiers, M. (Eds.) *Proof Technology in Mathematics Research and Teaching*, (Vol. 14 p. 323): Springer International Publishing.
- Richter-Gebert, J. (2013). Mikrolaboratorien und virtuelle Modelle in universitären Mathematiklehrveranstaltungen. In Ableitinger, C., Kramer, J., Prediger, S. (Eds.) (pp. 169–186). Heidelberg: Springer.
- Rieber, L.P. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research & Development*, 44(2), 43–58.
- Sinclair, N., & Robutti, O. (2013). Technology and the Role of Proof: The Case of Dynamic Geometry. In Clements, M.A.K., Bishop, A.J., Keitel, C., Kilpatrick, J., Leung, F.K.S. (Eds.) *Third International Handbook of Mathematics Education* (pp. 571–596). New York: Springer.
- Sümmerrmann, M.L. (2019a). Ariadne – A Digital Topology Learning Environment. *The International Journal for Technology in Mathematics Education*, 26(1), 21–26.
- Sümmerrmann, M.L. (2019b). Construction of a path with winding number 1. <https://youtu.be/JCiy--dZpZg>.
- Sümmerrmann, M.L. (2019c). Path with winding number 1 is not null-homotopic. <https://youtu.be/Y0TcB8UjEWE>.
- Sümmerrmann, M.L. (2019d). The winding number is additive. <https://youtu.be/8rZD733bx8w>.
- Sommerhoff, D., & Ufer, S. (2019). Acceptance criteria for validating mathematical proofs used by school students, university students, and mathematicians in the context of teaching. *ZDM*, 51(5), 717–730. <https://doi.org/10.1007/s11858-019-01039-7>.

- Stylianides, A.J. (2007). Proof and Proving in School Mathematics. *Journal for Research in Mathematics Education*, 38(3), 289–321.
- The Coq Development Team (2019). The Coq Proof Assistant, version 8.9.0. <https://doi.org/10.5281/zenodo.2554024>.
- Tymoczko, T. (1979). The four-color problem and its philosophical significance. *The Journal of Philosophy*, 76(2), 57. <https://doi.org/10.2307/2025976>.
- Venema, G. (2013). *Exploring Advanced Euclidean Geometry with GeoGebra: Mathematical Association of America*.
- Vérillon, P. (2000). Revisiting Piaget and Vygotsky: In Search of a Learning Model for Technology Education. *Journal of Technology Studies*, 26(1), 3–10.
- Victor, B. (2014). Humane Representation of Thought: A Trail Map for the 21st Century. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (pp. 699–699). New York: ACM.
- Voevodsky, V. (2015). An experimental library of formalized mathematics based on the univalent foundations. *Math. Structures Comput. Sci.*, 25(5), 1278–1294. <https://doi.org/10.1017/S0960129514000577>.
- von Fritz, K. (1955). Die APXAI in der griechischen Mathematik. *Archiv für Begriffsgeschichte*, 1, 13–103.
- Weber, K. (2014). Proof as a Cluster Concept. In Fukawa-Connelly, T., Karokak, G., Keene, K., Zandieh, M. (Eds.) *Proceedings of the 17th Annual Conference on Research in Undergraduate Mathematics Education*.
- Weber, K., & Mejia Ramos, J. (2015). On relative and absolute conviction in mathematics. *For the Learning of Mathematics*, 35, 15–21.
- Whitehead, A.N., & Russell, B. (1910). *Principia Mathematica* Vol. 1: Cambridge University Press.
- Yackel, E., & Cobb, P. (1996). Sociomathematical norms, argumentation, and autonomy in mathematics. *Journal for Research in Mathematics Education*, 458–477.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Moritz Lucius Sümmerrmann¹  · Daniel Sommerhoff²  · Benjamin Rott¹ 

Daniel Sommerhoff
sommerhoff@math.lmu.de

Benjamin Rott
benjamin.rott@uni-koeln.de

¹ Institute of Mathematics Education, University of Cologne, Gronewaldstrasse 2, D-50931, Cologne, Germany

² Chair for Mathematics Education, Ludwig-Maximilians-Universität München, Theresienstrasse 39 D-80333 Munich, Germany