Enhancing Mathematization in Physics Education by Digital Tools

Lana Ivanjek, David Perl-Nussbaum, Lorena Solvang, Edit Yerushalmi, and Gesche Pospiech

Abstract The GTG Mathematics in Physics Education follows the philosophy of supporting physics understanding by the conscious use of mathematical structures in physics teaching. We discuss the possible roles of digital tools in promoting physics understanding by fostering sense making of computational models, using geometrical visualizations or interpreting app-generated diagrams in a physics context. We look into three types of digital tools: (a) Smartphone apps that allow data collection from the phone's internal sensors to effortlessly produce graphical representations of the data. (b) GeoGebra, that combines different mathematical representations and allows their visualization and manipulation. (c) Computational modeling via Vpython where students can build or manipulate a computational model and compare it to experimental results. We will describe the potential of these tools to improve understanding of different mathematical features in physics, as well as obstacles that educators should take into account. In addition we present some empirical findings concerning graphs from smartphone apps and experiences from teacher professional development.

Keywords Digital tools · Mathematization

L. Ivanjek (✉) · G. Pospiech
Faculty of Physics, TU Dresden, Haeckelstr. 3, 01069 Dresden, Germany
e-mail: lana.ivanjek@jku.at

D. Perl-Nussbaum · E. Yerushalmi
Weizmann Institute of Science, 7610001 Rehovot, Israel

L. Solvang
Department of Engineering and Physics, Faculty of Health, Science and Technology, Karlstad University, 65188 Karlstad, Sweden

C. Fazio and P. Logman (eds.), *Physics Education Today*, Challenges in Physics Education, https://doi.org/10.1007/978-3-031-48667-8_3

1 Introduction

We live in a time when digitalization influences almost all areas of our lives and is also becoming increasingly relevant for students, school, and teaching. The diversity and multitude of possibilities to use and implement digital tools is overwhelming, allowing passive reception as well as active use and creative construction. In this contribution, we cover a range of possible tools, which we selected for their relevance to mathematization in physics lessons. The goal is to analyze which options digital media could offer in supporting mathematization in physics, focusing on modeling, sense making of the interplay between mathematical model and reality, and specifically the interpretation of app-generated diagrams. These aspects were chosen because digital media provide above all possibilities for visualization of functional relations and mathematical modeling, and thus might support the process of mathematization for learners. This means that students are enabled to relate abstract mathematical models or graphs of dependencies directly to the physics phenomena. We therefore describe and apply selected tools covering a broad range of possible uses in lessons connected with these specific potentials and analyse related difficulties. We start by describing smartphone apps, which provide easy access to experiments and graphs, but do not support the active creation of mathematical models. On the other end of the range of tools, we place computational modeling tools (we chose Vpython), where students can program and build their own physical–mathematical models and visualize them. In the middle between these two extremes we could place GeoGebra, a primarily geometric tool that offers a wide range of applications and can be used for active modeling as well as for applying and interpreting given or known models.

As smartphones are meanwhile a normal tool in everyday life and nearly every student has access to it, it offers itself for manifold usages in the physics classroom. Here we concentrate on the possibilities for experimenting with help of the in-built sensors. For this purpose, a big range of apps is available in the different app-stores that visualize the sensor data. To use these smartphone apps efficiently in the classroom, it is important to know how students can handle the diagrams resulting from experiments with the in-built sensors. In this case the focus lies on interpreting the acquired data with suitable models within a given physics context.

Another type of digital tools is the dynamic mathematics software GeoGebra [1]. Perhaps better known for its computer versions, GeoGebra can be used by physics teachers from primary school to university level to create simulations, augment real experiments, and/or directly involve students in the process of creating mathematical models of physical phenomena [2]. All these types of activity require the sense making of mathematical structures in physics contexts [3].

More advanced competences include students to be able to work with computational models. Environments such as Vpython [4] enable students to construct theoretical models and to compare them to physical phenomena. This also requires appropriate preparation of teachers.

In the following, we will first describe the mentioned digital tools in detail before we shed light on some implementation schemes.

2 Selected Digital Tools: Potential and Difficulties

Apps for phones have been developed that allow data collection from the phone's internal sensors and facilitate video analysis and stroboscopic recordings. These apps allow the user very quickly to get information about the measurement in form of a graphical representation. The apps used in this study were Phyphox [5] for use of the phone's internal sensors and Vernier Video Physics [6] and Vianna [7] for the video analysis.

The app Phyphox makes it possible to use the sensors in a phone or a tablet for experiments [5]. The app is available for free both on Android and on iOS. It was developed at the RWTH Aachen and has found a wide application at university physics courses and at school for simple phone experiments. Within the app it is possible to choose between the raw sensors (acceleration, gyroscope, location, light sensor, magnetometer and pressure sensor), acoustics experiments like audio amplitude and spectrum, Doppler effect or tone generator and prepared mechanics experiments (centripetal acceleration, pendulum experiment, spring oscillator and measurement of energy loss during inelastic collisions). It is possible to create a customized experiment in the app and to export the data for further analysis. The app does an automated data analysis, and the output is a numerical value or a graph.

Vernier Video Physics and Vianna are both apps for video analysis working on the iOS. They make possible the video analysis of pre-recorded motions and create distance-time and velocity–time graphs. In order to create the graphs, the user needs to do following steps: (1) record the motion of a ball, (2) choose the position for the origin of coordinate system, (3) use the scale in order to determine the real distances and (4) mark points or track the object in motion. The programs do all calculations and the user obtains the graphs of motion.

The previous examples demonstrate the use of digital tools to construct measurement models of data as well as to collect and interpret data [8]. Digital tools can also be used to bridge or connect these data and other real-world phenomena to mathematical and theoretical models, derived from the laws of physics. Recently, the dynamic mathematics software, GeoGebra [1] has received attention in physics education [9], due to the fact that it enables teachers (with or without programming knowledge) to create their own mathematical models of physical phenomena such as simulations and to design engaging learning environments that enhance students' cooperative learning about mathematical models of physical phenomena [10].

GeoGebra incorporates geometry, algebra, calculus, and spreadsheets into a single package, creating a dynamic connection between all these different mathematical representations. For example, the user can insert expressions or equations into the 'algebra window' which will be automatically, dynamically rendered in the 'graphical window' and vice versa. In this way, the process of modeling physical phenomena involves mainly the implementation and manipulation of mathematical representations. One advantage of GeoGebra when compared to the coding or the inserting of already existing coding sequences is that GeoGebra only requires prior understanding of the mathematics itself rather than specific programming knowledge [11]. In addition, GeoGebra can also be used as a video analyzing tool, that is capturing and/or opening a digital video file of experiments and then analyzing the motion of objects in the video. This way of using GeoGebra is related to the ways the phone apps described above are used. The major difference is that while data can be collected automatically by using the phone's sensors, in GeoGebra the data is collected separately and then inserted manually.

GeoGebra can be freely downloaded from its website, www.geogebra.org, or it can be used online. It works on many operating systems such as Windows, macOS, and Linux, on tablets and phones [11] and is multilingual in its menus and its commands [12]. GeoGebra is also a community-supported learning environment. At the time of writing, GeoGebra's library, www.geogebra.org/materials, contains many numerous educational materials uploaded by its users. Most materials designed for physics education are simulations of physical phenomena and are intended primarily for secondary school education. However, as the tool allows teachers to design custom-made simulations, to modify the existing ones, and augment real experiments, GeoGebra can be used at all levels of education—from preschool to advanced university courses [2]. One of its big advantages is that it allows the students to change the values of the variables, thus stimulating to reflect on the validity of the physics law in question. The simulations and other educational material can be used for classroom activities, homework, and for online learning.

One step further in modeling would be to introduce computation into introductory physics courses. This can provide students with opportunities to create and explore computational models and to visualize abstract mathematical concepts. Several programing environments were designed in the past few decades to suit the needs of undergraduate physics students, such as M.U.P.P.E.T [13], Netlogo [14], and EJS [15]. The computational modeling platform Visual Python (VPython) is a 3D graphics system developed by Scherer, Sherwood and Chabay [4]. It is an extension of the Python language that is relatively easy to learn and use providing the user with the ability to model three-dimensional scenes. The strength of VPython is that it minimizes the amount of programming constructs the students have to learn: They do not need to get into the detailed syntax of building the display environment. Other advantages of VPython include the matching of the basic program constructs to key physics constructs (i.e., vector notation) and the straight-forward animation method (motion is generated using loops, updating objects' position).

Figure 1 shows an example of a Vpython program that models a ball under gravitational and drag forces. The program represents the ball, its trace of motion and the updated position-time graph (https://www.glowscript.org/).

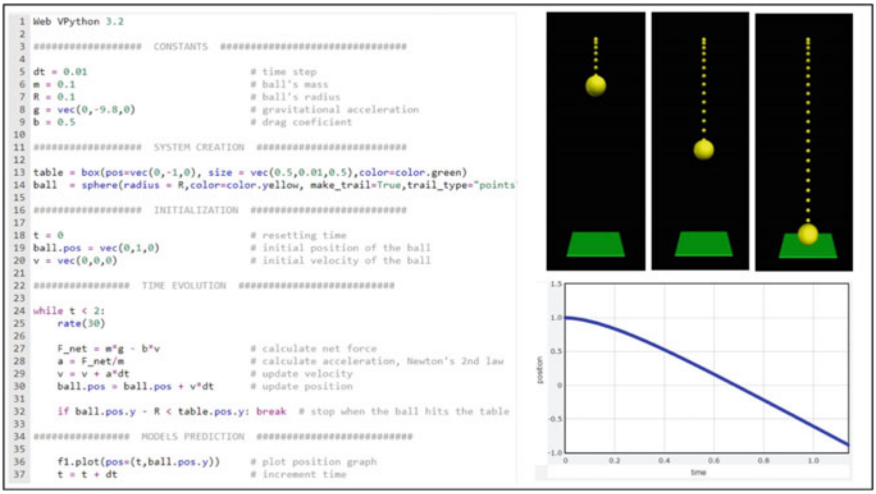

Fig. 1 A computational model of a ball falling in air, executed in the Vpython platform. The code itself is displayed on the left, and an animation of the modelled object and a position-time graph on the right. Copyright (C) 2011 by David Scherer and Bruce Sherwood

3 Status of Research and Theoretical Background

3.1 Visualization, Modeling and Mathematization

As shown in the section above, digital tools are often used to create mathematical models of data collected from the physical world or to create idealized models of physical phenomena [2, 4, 13, 15–18]. The resulting models are mathematical representations (i.e., formalism) of the phenomena displayed on a computer or phone screen. Thus, mathematization is an integral aspect of physical modeling and the digital tools used for creating the models are foremost visualization tools. In addition, digital tools can facilitate students' transition from experience to mathematical models and vice-versa by acting as a type of 'catalyst' between the physical world and the mathematical world. To explore how digital tools provide access to formal physics ideas, a chapter by Euler et al. [19] drafted for the forthcoming International Handbook of Physics Education Research, synthetizes the physics education literature related to the interplay of visualization and mathematization in physics education, specifically in the context of using digital tools.

Within this project, in order to describe the work related to the interplay of digital visualization and mathematization in physics education, the modeling framework of Hestenes [20] was combined with diSessa's (1988) perspective of semi-formalisms in physics learning and Uhden et al.'s [21] theory of modeling with degrees of mathematization (Euler et al. [19]). To highlight the interdependence between the physical domain and a mathematical domain (i.e., formalism), Hestenes [20, 22] describes the process of doing physics as a ‘modeling game’. During this process, physicists move between the physical and the formal domains as they use mathematical representations to describe the structure and dynamics of a physical phenomenon, and, not least, interpret how the mathematical model represents the phenomenon at hand [22]. To describe how digital tools can assist students during this mathematization process, diSessa suggests that digital tools can act as semi-formalisms for students by enabling the transition from personal experience of the physical world to formalisms and vice-versa. Such semi-formalisms allow students to control and manipulate certain variables connected to the physical phenomenon which is being modeled, acting as a ‘catalyst’ in students’ learning processes [23]. However, in the process of constructing a mathematical model of a physical phenomena, different degrees of mathematization might occur [21]. Hence, a digital tool that assists students during the mathematization process could function as a ‘catalyst’ in connecting the physical domain and the mathematical domain or in connecting different representations within the mathematical domain. Thus, Euler et al. [19] suggest two distinct functions that visualization tools can fulfill in facilitating mathematization in physics:

Function I: bridging between physical phenomena and formalisms, by

(a) linking physical phenomena to formalisms and/or,
(b) augmenting physical phenomena with formalisms, and

Function II: bridging between idealized models of physical phenomena and formalisms by

(a) linking models to formalisms and/or
(b) augmenting simulations with formal representations.

3.2 Smartphone Apps

The phone apps mentioned above have been used for a broad variety of experiments [16–18]. In their paper, Staacks et al. [16] describe the ways to use Phyphox in a rolling experiment and an elevator experiment. For the elevator experiment, the phone is put on the floor of an elevator and the movement of the elevator is tracked using the phone’s atmospheric pressure sensor and its accelerometer. The height differences are then calculated using the recorded atmospheric pressure. The (numerical) derivative of these height values gives a vertical speed and the accelerometer directly provides the vertical acceleration of the elevator. As a result, students get the graphs showing the altitude, vertical speed and acceleration as a function of time, as shown in Fig. 2.

Götze et al. [18] describe the use of Phyphox for two simple experiments on the simple harmonic oscillator, which can be done with high school students and Pierratos and Polatoglu describe the use of the optical stopwatch function for quantitative kinematics [17]. All those papers emphasize the potential benefit of the app for the motivation of the students, the accessibility of a variety of experiments for the students, as well as automated data analysis.

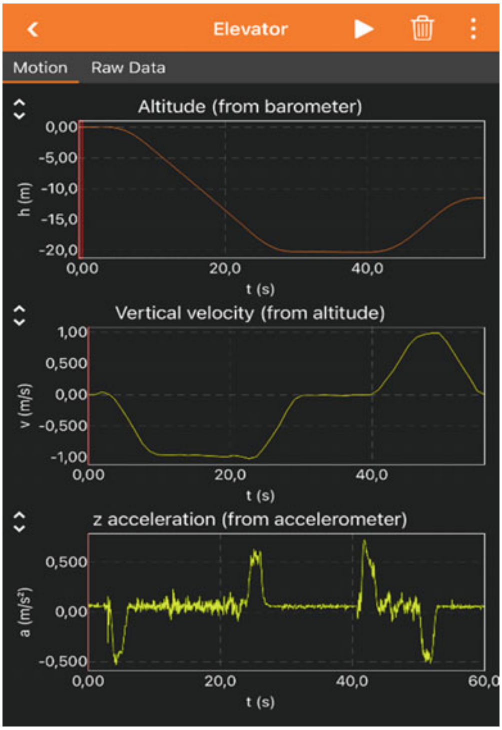

Fig. 2 Elevator experiment in Phyphox App: Three different kinematic quantitites are shown atop each other: position, velocity and acceleration. These graphs have to be related to each other and to be interpreted with respect to their physics meaning

However, up to now there is no accompanying research regarding the difficulties with reading and interpreting the graphs that students get as a result of the data analysis. Previous research on student understanding of graphs shows that a majority of students have difficulties interpreting and calculating the slope of the kinematic graphs [24–28], as well as interpreting the meaning of the area under the graph [27]. McDermott et al. detected in their research with kinematic graphs that students have the following difficulties: slope-height confusion, difficulty in making connections between different types of graphs, difficulty in interpreting the meaning of the area under a curve, difficulty in distinguishing the shape of the graph from the shape of the body's trajectory, difficulty in understanding the meaning of the sign of velocity and acceleration [24]. Leinhardt et al. have summarized the three main difficulties with the graphs as interval-point confusions (focusing on a single point of the graph instead of using an interval), slope-height confusions (when students mistake the height of the graph for its slope—just reading off the y-coordinate) and iconic confusions (incorrect interpretation of the graph as an actual picture of the motion) [28]. Although there is no research on the graphs from phone apps, it is to be expected that students could have similar difficulties, when it comes to the interpretation of the graphs generated from phone experiments. In order to exploit fully the potential of the direct visualization of experimental outcomes in graphs for bridging between the experiments or phenomena and models or formalism, these difficulties have to be precisely known and addressed in teaching.

3.3 GeoGebra

GeoGebra has been advocated as a user-friendly software that can be operated intuitively [11, 29, 30]. The studies on teaching sequences supported by GeoGebra simulations conducted by Malgieri et al. [30, 31] include a collection of GeoGebra simulations developed by the students or by the researchers to assist students in learning quantum physics at a basic level based on Feynman's sum over paths' approach. The studies show that by using these teaching sequences, students have improved their understanding of several conceptual issues and their ability to use the 'sum over path' method for problem-solving, as well as their ability to express themselves using an expert-like language. Regarding the collection of simulations and the software used for designing it, the researchers consider that GeoGebra is a valuable supporting software as it 'makes the mathematical models behind the simulations completely transparent and easily accessible to the user, and avoids producing the impression that complex and exotic algorithms are at work' [30]. A study conducted by Solvang and Haglund [32] analyses specific bodily practices (e.g. gestures, enactment) during students' interaction and constructions of representations in relation to a GeoGebra simulation of friction. The simulation represents a block sliding over a horizontal surface. The block is pulled by a hand holding a dynamometer, which shows the value of the pulling force. Simultaneously, a force–time diagram is displayed. Students could change the materials of the block and the surface, the value of the block's weight, and the base area. They could also start and pause the simulation or reset the graph. During their sense-making processes, students were triggered by specific features of the simulation—features connected with microscopic aspects of friction—to improvise their own representations as means of dealing with interpretational problems. For example, one of the students exaggerates the intermittent movement of the block caused by the protuberances of two rough surfaces by enacting the movement of a jumping frog and making choo-choo train sound effects. During their sense-making processes, students moved back-and-forth between the mathematical model of friction and the physical world of gestures and enactment, while the software was used as a catalyst in students' learning processes. With a simple push of a button, students could test and compare their ideas, the mathematical model, and their improvised bodily representations as the mathematical model could be reproduced dynamically in real-time.

3.4 Computational Modeling Environments

Computational modeling environments have been shown to enable secondary students to construct theoretical models for a variety of phenomena that are too complex to be modeled analytically, and compare them to experiments [33, 34]. As pointed out by Tang et al. [35] secondary school inquiry often emphasizes the experimental aspects of research and lacks the theoretical modeling aspect that is critical to the physics research process. Computational modeling can provide a solution to this challenge and overcome students' limited mathematical knowledge using step-by-step computational methods.

The computational modeling environments that were described above [13–15] have been designed and implemented in undergraduate physics courses. These courses are typically taught by researchers who are familiar with computational tools, at least to the extent required of their students. When integrating computational activities into school courses, we must consider the teachers who often lack programming skills and whose self-efficacy in this field is low, perhaps even lower than that of their students [36–38]. In computational activities for students in introductory physics courses, the learning goal is to create autonomy in constructing the computational model. Is it possible to reduce aspects of this autonomy in modeling activities for high school students—so that they can be adopted by high school teachers?

One possible way to increase teachers' sense of competence is through activities that aim to attribute to the understanding of an existing program through its activation and manipulation, without having to write its code. This differs from working with pre-built simulations, as learners can 'open the hood' of the model and understand the implementation of the physical laws and Euler's approximation method in the computer code. We adopted this approach, and report on the design and implementation of a sequence of computational modeling activities using Vpython in an inquiry-based workshop for 9th grade physics teachers.

4 Interpretation of Graphs Generated by Phone Apps

We investigate how students understand and interpret graphs generated by phone apps in order to identify if there are specific difficulties in addition to the known problems in interpreting kinematics graphs.

4.1 Research Questions and Method

The main research questions for the interpretation of graphs generated by phone apps were:

1. What are the main observed students' difficulties with graphical representations from phone apps?
2. What are similarities and differences to already reported students' difficulties with graph interpretation?

To investigate how future physics teachers can deal with graphs and images from phone apps and to answer these questions, a questionnaire with a total of 7 open-ended questions was developed and given to a total of 58 students from TU Dresden and 55 students from University of Vienna. The allocated time for taking the questionnaire was 45 minutes. The questionnaire contains graphs from the apps Video Physics, Vianna, PhyPhox and Sony Motion Shot. Two questions were related to the graphs from video analysis of the motion (free fall and a ball rolling on the incline), three graphs were generated with the app PhyPhox [5] using the internal smartphone sensors (elevator, rotational motion and motion of a car) and two representations included stroboscopic images of the motion. The students had to read different physical parameters from the graphs and analyze the graphs. The answers were analyzed and categorized using the framework of qualitative content analysis by Kuckartz [39] to find out the most common difficulties with the representations from phone apps. In the subsequent chapter, two examples will be discussed: the free fall example and the elevator example.

4.2 Results

Free fall question. The question shows the graph from the video analysis program Vernier Video Physics for a ball that has been released from the hand and falls to the ground and bounces back. Students were shown the video of the experiment, as well as the position-time and velocity–time diagrams that are the result from the video analysis. Based on the graph students were asked to determine the acceleration of the ball at the moment t = 2,5 s. This moment is the turning point of the ball. The main strategies that students used were:

- calculating the slope from the v-t diagram
- stating that the acceleration equals g, because it is the free fall situation
- stating that the acceleration is zero
- using the wrong sign for the acceleration

The correct strategy included calculating the slope from the v-t diagram. 10% of the students from TU Dresden and 18% of students from University of Vienna used that strategy. In addition, 28% of students from TU Dresden and 22% of students from University of Vienna concluded that the acceleration is 9,81 m/s^2, because of the free fall and due to the gravity. Although this is a right answer, those students were not using the graphs at all. Most of the students (30% of students from TU Dresden and 35% of students from University of Vienna) said that the acceleration of the ball is zero. Their explanations included the use of the wrong formula (a = v/

t) and slope-height confusion (because the velocity is zero, the acceleration is also zero).

Elevator question. The question shows the graph from the app Phyphox and the height-time, velocity–time and acceleration-time graphs for a motion of an elevator that first goes downwards and then upwards, as shown in Fig. 2. Students were first asked to detect when the elevator is moving in which direction and afterwards when the elevator is speeding up and when slowing down. The main correct strategies that students used were reasoning based on the v-t graph and the direction of motion or reasoning based on the a-t graph and direction of motion. In the sum 33% of students from TU Dresden used that strategy, as well as 31% of students from University of Vienna. The main wrong strategy was linked to the idea that the elevator is speeding up when the acceleration is positive and slowing down when the acceleration is negative (used by 21% of students from TU Dresden and 31% of students from University of Vienna), followed by the reasoning that the elevator is speeding up or slowing down only when the acceleration changes its value. Other difficulties included the thinking that the elevator is always speeding up or slowing down. Additionally, difficulties with non-idealized graphs, and interval-point confusion were observed.

5 Implementation of GeoGebra to Facilitate Mathematization

In this section, it is highlighted how digital technologies could perform the role of semi-formalisms. In the following, GeoGebra is used as an example of a digital tool which can facilitate mathematization through Function I and Function II, described above [19]. We illustrate the two mathematization functions with GeoGebra, which unlike many other visualization tools used in physics education can flexibly exemplify both mathematization functions depending on how it is implemented.

Function I: Bridging physical phenomena and formalism

(a) by linking physical phenomena to formalisms:

GeoGebra can be used as a video analyzing tool, that is capturing and/or opening a digital video file of experiments and then analyzing the motion of objects in the video. This way of using GeoGebra is related to interactive video. Users can also insert just a picture of a phenomenon, such as a basketball being thrown into a hoop (Fig. 3). Different positions of the ball at different times are already being marked in the picture. The equation of the fitting curve contains three sliders, a, h and k as coefficients. Because of the dynamic link between algebraic and graphical representations of an object, realized by dragging the sliders, the user can find the equation for the graph that best fits the trajectory of the ball.

(b) by augmenting physical phenomena with formalisms:

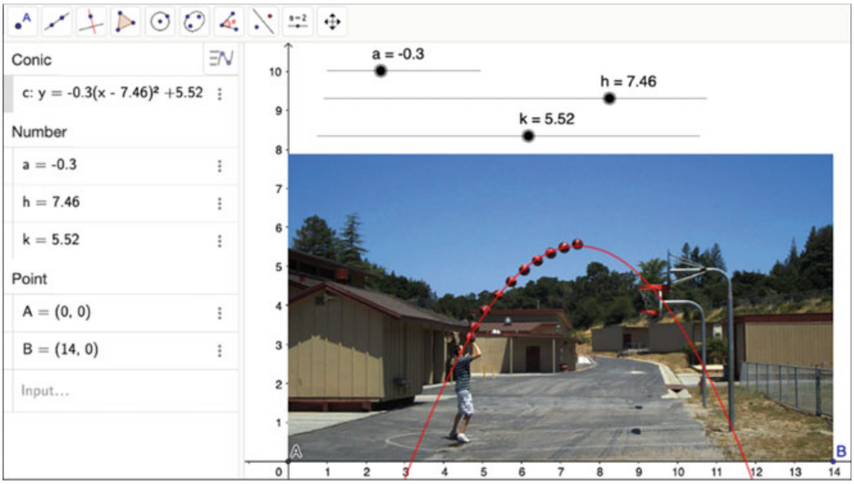

Fig. 3 Linking physical phenomena to algebraic and graphic representations using a GeoGebra simulation of a projectile motion, freely available at https://www.geogebra.org/m/pgqKNSak)

GeoGebra can also be used for augmenting physical phenomena. If a computer or phone has a camera, virtual objects, such as force arrows and light rays, can be constructed with GeoGebra and then accessed with GeoGebra 3D Calculator by pressing the tool's AR button. For example, the motion of an object on an inclined plane can be augmented by a dynamic GeoGebra model of the resulting force (Fig. 4).

The model displays the resulting force as the vector sum of the gravitational force and the normal force. The mass of the cart and the angle of inclination can be modified to correspond to the real setup (Fig. 4).

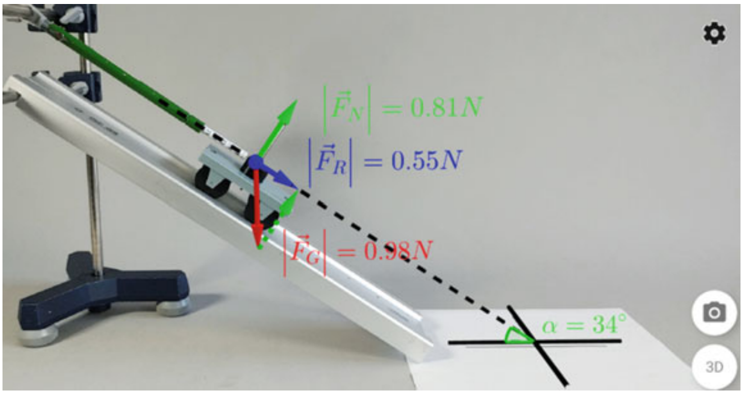

Fig. 4 Motion of an object on an inclined plane augmented by a dynamic model constructed in GeoGebra (reproduced with the permission of the authors from Teichrew and Erb [40], and available at www.geogebra.org/m/pafx6xfu#material/qhb4yeht)

Commands	Results
$\alpha = 25°; v\_0 = 1.6; g = 9.8; L = 7; t = 0; O = (0, 0)$	Defining system parameters
$a = g^*\sin(\alpha)$	Defining acceleration
$d = V\_0 * t + 1/2 * a * t\^2$	The displacement at time t
$t\_f = (-V\_0 + \mathrm{sqrt}(2a^*L + V\_0^2))/a$	The time required to cover the distance L
$M = ((L - d)^*\cos(\alpha), (L - d)^*\sin(\alpha))$	The position of the body at time t

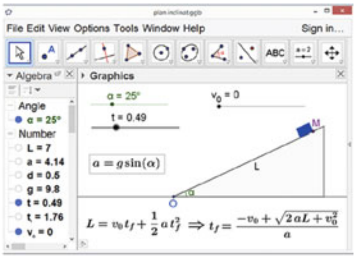

Fig. 5 GeoGebra screenshot showing the algebraic representations (left) and the idealized geometrical model (right) of an inclined plane (reproduced from Marciuc et al. [41] with the permission of the ADL ROMANIA)

Function II: Bridging idealized models of physical phenomena and formalisms

(a) by linking models to formalisms:

This simulation made in GeoGebra exemplifies the second function of visualization. It presents an idealized geometrical model of a block being pulled across a frictional surface (Fig. 5 right). To create this simulation, the user needs to insert the algebraic representations of all the geometrical representations (Fig. 5 left). In addition, relevant equations which describe the motion of the block can be inserted into the geometrical widow. The users can then manipulate the relevant parameters, such as the angle of the inclined plane and observe a dynamically generated motion of the block.

In all examples above, GeoGebra can be seen as a tool that ostensibly facilitates students' transition between relatable physical phenomena and the formalisms that the discipline of physics uses to mathematize those phenomena as part of problem solving and analysis.

6 Professional Development of Physics Teachers with Computational Modeling

We report on the design and implementation of a sequence of computational modeling activities using the Vpython platform in an inquiry-based workshop for 9th grade physics teachers. We focus on two research goals: (1) Characterizing design guidelines for computational modeling activities that enable teachers without programming expertise to successfully complete them in a limited time frame of a workshop. (2) Examining teachers' perceptions of the affordances and challenges of the computational modeling activities they experienced.

6.1 Research Approach

In order to characterize design guidelines for computational modeling activities that are manageable for 9th grade physics teachers (Research Goal 1), we examined two designs of the activity. The pilot design was tried out twice, in the summers of 2017 and 2018. The final design considered the feedback from the pilot version and was tried out on the summers of 2019 and 2020. The versions were tested based on two main measures: teachers' ability to complete the activities in the limited time frame that could be devoted to computational modeling in an inquiry-based PD workshop, and the extent of classroom implementation. To learn about teachers' perception on the activities (Research Goal 2) we used questionnaires and open-ended questions to reflect on the final version of the computational sequence.

6.2 Context

Gateway to physics is an inquiry-based program intended to motivate 9th grade students to choose physics as a major by increasing their interest and self-efficacy, as well as the self-efficacy of their teachers. Two learning modules were developed, both investigating straight-line motions under acting forces. The 1st module dealt with oscillations of a mass on a spring and the 2nd with objects falling in air. During the summers of 2017–2020, the modules were introduced in PD workshops (30 h over 4 days for each module) for teachers from a variety of disciplinary backgrounds. The first two days of the workshop focused on experimental investigations and the last two days on theory, discussing the related theory qualitatively, and using computational modeling to overcome mathematical complexity (both systems involve nonlinear equations) and produce quantitative models and predictions.

6.3 Design of Pilot Version

The activities were designed as a middle ground between using ready-made models and writing models from scratch. Our approach was to 'open the hood' and allow students to observe a working computational model, understand the function of each line of code, and then modify it according to their needs. The activities did not address the algorithmic considerations of the underlying program. Two activities served as an introduction to the computational activities. The first introduced the motivation for computational modeling: Teachers used structured worksheets to discuss the possibility of predicting motion in different situations, and the second introduced Euler's step-by-step computational method. The computational activities were carried out using Trinket.io—a free online tool for programming activities and courses. This platform runs Vpython—a 3D graphics package for Python, a widely used programming environment for scientific modeling. The sequence consisted of 4 activities:

1. Acquaintance with the programming environment—where students learn to create different objects and place them.
2. Constant velocity motion—students are introduced to the "while" loop and its use to move objects at constant velocity.
3. Motion under a constant force—students learn how to apply Euler's method to construct models of motion under constant forces based on Newton's second law of motion.
4. Comparison of model and experimental results—students produce a theoretical trace of the motion of objects, compare it with an experimental trace they created using the Tracker video analysis software [42], and revise their model to better fit experimental data.

Participants received minimal instruction, and learned the meaning of the different parts of the program through hands-on tasks. For example, in the constant velocity activity, they were shown a ball moving from the right side of the screen to the left and were asked to make it move in the opposite direction (requiring a change in the direction of velocity and in the initial position of the ball).

6.4 Findings—Pilot Version

53 teachers participated in the pilot activities. We witnessed a high dropout rate: ~20% of teachers did not complete the entire sequence of activities. Most of them had no prior background in programming, resulting in low self-efficacy. Among the teachers who did complete the activities, only a few implemented them in their physics classes, either due to external constraints (inadequacy to the curriculum, lack of computers or time) or lack of confidence to adapt such an innovative curriculum.

6.5 Design of Final Version

The results of the pilot study showed that reducing autonomy in writing the code was not sufficient, as the teachers expressed low self-efficacy and frustration. To enable more teachers to successfully accomplish the computational activities we revised the activities, using the scaffolding mechanisms of structuring and problematizing [43]. Each activity was divided into the following three steps:

1. Exploring an existing program by running it, making guided manipulations and describing their outcome.
2. Sense making of the program and the role of each command through guiding questions.

3. Application: modification of the code to meet different tasks.

For example, the mechanism of a simple loop is presented through: (1) Exploring—students run a loop that counts from 1 to 10 and describe the outcome of small changes they are guided to make. (2) Sense making of the components of the program through guiding questions, such as what is the role of the code line 'while m < 10'. (3) Application—students are required to change the program, so it counts to 20 or by jumps of 3.

Another revision made in the final version of the activities had to do with the comparison of the computational and experimental models. Models were compared through various mathematical representations: tables, velocity–time graphs, and traces of motion instead of only comparing the traces. In addition, the activities were incorporated into the same learning management system as all other PD activities—a Moodle-based platform the teachers were familiar with, instead of an external platform, to help teachers view the unit as an integral part of the workshop and avoid switching between platforms.

6.6 Findings—Final Version

67 teachers experienced the refined activities during the summer of 2019 and 2020. The final version of the activity was successful in keeping the teachers engaged: previous research [44] showed that the 2019 teachers reported higher programming self-efficacy after completing them. In the 2020 PD workshop, teachers successfully completed the activities—only 1/18 dropped out. Furthermore, they appreciated the activities as contributing to their understanding of the theoretical as well as the experimental aspects of the inquiry process:

> I really liked the perspective it gives, the digital "calculation" so you see results and graphs that come out… but while here we will get accurate and perfect graphs in the experiment we will get slightly different graphs, which gives us another way of understanding measurement errors in experiments
>
> Through the step-by-step solution of Newton's second law, without going into the concept of acceleration, made me re-examine my ways of teaching inquiry... step-by-step analysis develops students' good understanding of motion

However, most of the teachers (~75%) stated they still do not feel confident enough to implement the activities in their classrooms, mainly due to insufficient expertise in the programming environment.

7 Conclusion

We have described a possible theoretical framework concerning the relation of visualization and mathematization in physics with respect to selected digital tools. Normally, great expectations are placed on the supportive effect of these well-known tools for physics understanding. However, less is known about students' actual experiences with these tools. Our paper offers valuable design guidelines for implementation, as well as empirical results, indicating that a certain degree of precaution is advisable here. Students show clear problems in interpreting graphs that are generated by the corresponding apps, for example, when experimenting with the phone. This observation is coherent with previous research on interpreting graphs. It has to be considered also that the task is quite complex: students have to relate experiment, physical understanding and the characteristics of the graphs to each other in order to arrive at a correct interpretation. Difficulties known from the literature are observed, such as slope-height confusion or the inadequate differentiation of acceleration and velocity. In addition, specific difficulties e.g. connected to the fluctuations of experimental values were observed. Thus, when using such apps, the teachers must be aware that the interpretation of the graphically represented results from the experiments requires numerous steps done by the learners. Such problems can also arise in the case of GeoGebra. In this case, however, the possibility of switching between different representations with a simple click could support in interpreting the graphs and understanding the physical models. Comparable observations are made in the context of a computational modeling environments. While the mechanisms of structuring and problematizing [43] have helped teachers to make sense of the theoretical models and their implementation in a Vpython program, and to appreciate the potential of computational modeling to physics learning, it seems that there is still a long way to go in mastering this skill well enough for them to confidently implement this method in their classrooms. Overall, it is observed that in order to exploit the potential of digital tools in the context of mathematization, the teachers have to be aware of the pitfalls and have to be able to diagnose the arising difficulties, for which this contribution provides guidance. Taking those into account, the use of digital tools could enhance physics understanding and help to apply mathematical tools.

References

1. Hohenwarter, J., Hohenwarter, M., Lavicza, Z.: Introducing dynamic mathematics software to secondary school teachers: the case of GeoGebra. J. Comput. Math. Sci. Teach. 135–46 (2009)
2. Solvang, L., Haglund, J.: How can GeoGebra support physics education in upper-secondary school—a review. Phys. Educ. **56**, 55011 (2021)
3. Kapon, S.: Unpacking sensemaking. Sci. Edu. **101**, 165–198 (2017)
4. Chabay, R., Sherwood, B.: Computational physics in the introductory calculus-based course. AJP **76**, 307–313 (2008)
5. Staacks, S.: Phyphox
6. Vernier Video Physics

7. Freie Universitaet, Berlin, Vianna
8. Dounas-Frazer, D.R., Lewandowski, H.J.: The modelling framework for experimental physics: description, development, and applications. Eur. J. Phys. **39**, 64005 (2018)
9. Milner-Bolotin, M.: Rethinking technology-enhanced physics teacher education: from theory to practice. Can. J. Sci. Math. Technol. Educ. **16**, 284–295 (2016)
10. Wassie, Y.A., Zergaw, G.A.: Capabilities and contributions of the dynamic math software, GeoGebra—a review. N. Am. GeoGebra J. 68–78 (2018)
11. Walsh, T.: Creating interactive physics simulations using the power of GeoGebra. Phys. Teach. **55**, 316–317 (2017)
12. Hohenwarter, M., Fuchs, K.: Combination of dynamic geometry, algebra and calculus in the software system GeoGebra. In: Computer Algebra Systems and Dynamic Geometry Systems in Mathematics Teaching Conference, pp. 1–6 (2004)
13. Redish, E.F., Wilson, J.M.: Student programming in the introductory physics course: MUPPET. AJP **61**, 222–232 (1993)
14. Tisue, S., Wilensky, U.: Netlogo: a simple environment for modeling complexity. In: International Conference on Complex Systems, vol. 21, pp. 16–21 (2004)
15. Christian, W., Esquembre, F.: Modeling physics with easy java simulations. Phys. Teach. **45**, 475–480 (2007)
16. Staacks, S., Hütz, S., Heinke, H., Stampfer, C.: Advanced tools for smartphone-based experiments: phyphox. Phys. Educ. **53**, 45009 (2018)
17. Pierratos, T., Polatoglou, H.M.: Utilizing the phyphox app for measuring kinematics variables with a smartphone. Phys. Educ. **55**, 25019 (2020)
18. Götze, B., Heinke, H., Riese, J., Stampfer, C., Kuhlen, S.: Smartphone-Experimente zu harmonischen Pendelschwingungen mit der App phyphox. PhyDid B-Didaktik Der Physik-Beiträge zZur DPG-Frühjahrstagung, pp. 233–239 (2017)
19. Euler, E., Solvang, L., Gregorcic, B., Haglund, J.: Visualization and mathematization: how digital tools provide access to formal physics ideas. In: International Handbook of Physics Education Research. Dordrecht, Springer (2023)
20. Hestenes, D.: Modeling games in the Newtonian world. Am. J. Phys. **60**, 732–748 (1992)
21. Uhden, O., Karam, R., Pietrocola, M., Pospiech, G.: Modelling mathematical reasoning in physics education. Sci. Educ. **21**, 485–506 (2012)
22. Hestenes, D.: Toward a modeling theory of physics instruction. Am. J. Phys. **55**, 440–454 (1987)
23. Solvang, L.: Educational technology for visualisation in upper secondary physics education: the case of GeoGebra Doctoral dissertation (2021). Karlstads universitet
24. McDermott, L.C., Rosenquist, M.L., van Zee, E.H.: Student difficulties in connecting graphs and physics: examples from kinematics. Am. J. Phys. AJP **55**, 503–513 (1987)
25. Beichner, R.J.: Testing student interpretation of kinematics graphs. Am. J. Phys. AJP **62**, 750–762 (1994)
26. Planinic, M., Milin-Sipus, Z., Katic, H., Susac, A., Ivanjek, L.: Comparison of student understanding of line graph slope in physics and mathematics. Int. J. Sci. Math. Educ. **10**, 1393–1414 (2012)
27. Planinic, M., Ivanjek, L., Susac, A., Milin-Sipus, Z.: Comparison of university students' understanding of graphs in different contexts. Phys. Rev. ST Phys. Educ. Res. **9**, 020103 (2013)
28. Leinhardt, G., Zaslavsky, O., Stein, M.K.: Functions, graphs, and graphing: tasks, learning, and teaching. Rev. Educ. Res. **60**, 1–64 (1990)
29. Kolář, P.: GeoGebra for secondary school physics. J. Phys. Conf. Ser. **1223**, 12008 (2019)
30. Malgieri, M., Onorato, P., de Ambrosis, A.: Teaching quantum physics by the sum over paths approach and GeoGebra simulations. Eur. J. Phys. **35**, 55024 (2014)
31. Malgieri, M., Onorato, P., de Ambrosis, A.: GeoGebra simulations for Feynman's sum over paths approach. Il nuovo cimento C **41**(3), 1–101–10 (2018)
32. Solvang, L., Haglund, J.: Learning with friction—students' gestures and enactment in relation to a GeoGebra simulation. Res. Sci. Educ. **52**(6), 1659–1675 (2021)

33. Aksit, O., Wiebe, E.N.: Exploring force and motion concepts in middle grades using computational modeling: a classroom intervention study. J. Sci. Educ. Technol. **29**, 65–82 (2020)
34. Lee, I., Grover, S., Martin, F., Pillai, S., Malyn-Smith, J.: Computational thinking from a disciplinary perspective: integrating computational thinking in K-12 science, technology, engineering, and mathematics education. J. Sci. Educ. Technol. **29**, 1–8 (2020)
35. Tang, X., Elby, A., Hammer, D.: The tension between pattern-seeking and mechanistic reasoning in explanation construction: a case from Chinese elementary science classroom. Sci. Educ. **104**, 1071–1099 (2020)
36. Dodero, J.M., Sáiz, M.S.I., Rube, I.R. (eds.): Proceedings of the 5th International Conference Technological Ecosystems for Enhancing Multiculturality TEEM, p. 10182017. ACM, New York, USA. (Cádiz Spain, 18 10 2017 20 10 2017)
37. Ketelhut, D.J., Mills, K., Hestness, E., Cabrera, L., Plane, J., McGinnis, J.R.: Teacher change following a professional development experience in integrating computational thinking into elementary science. J. Sci. Educ. Technol. **29**, 174–188 (2020)
38. Dodero, J.M., Mota, J.M., Ruiz-Rube, I.: Bringing computational thinking to teachers' training: a workshop review. In: Dodero, J.M., et al. (eds.) Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 1–6. ACM, New York, USA (2017)
39. Metzler, K., Kuckartz, U. (eds.): Qualitative Text Analysis: A Guide to Methods, Practice & Using Software. SAGE, London, England (2002)
40. Teichrew, A., Erb, R.: How augmented reality enhances typical classroom experiments: examples from mechanics, electricity and optics. Phys. Educ. **55**, 65029 (2020)
41. Marciuc, D., Miron, C., Barna, E.S.: Using GeoGebra and Vpython software for teaching motion in a uniform gravitational field. Rom. Rep. Phys. **68**(4), 1603–1620 (2016). https://doi.org/10.12753/2066-026X-16-210
42. Brown, D., Cox, A.J.: Innovative uses of video analysis. Phys. Teach. **47**, 145–150 (2009)
43. Reiser, B.J.: Scaffolding complex learning: the mechanisms of structuring and problematizing student work. J. Learn. Sci. **13**, 273–304 (2004)
44. Langbeheim, E., Perl, D., Yerushalmi, E.: Science teachers' attitudes towards computational modeling in the context of an inquiry-based learning module. J. Sci. Educ. Technol. **29**(6), 785–7961–12 (2020)