

The benefit of computational modelling in physics teaching: a historical overview

Jannis Weber¹  and Thomas Wilhelm¹ 

Department for Physics Education Research, Goethe-University, Frankfurt am Main, Germany

E-mail: weber@physik.uni-frankfurt.de and wilhelm@physik.uni-frankfurt.de

Received 6 November 2018, revised 17 February 2020

Accepted for publication 26 February 2020

Published 20 April 2020



CrossMark

Abstract

Computational modelling is not only an important element in scientific research, it also has a rich history in physics education and its application in the classroom has changed greatly over the last few decades. To explain why computational modelling is used in physics education, we discuss its use in teaching and the benefits to the teaching and learning process. Secondly, historical developments are highlighted, and different methods of computational modelling are discussed in more detail, developing the desired features of modern software for classroom computational modelling. A review of the research in this field was conducted, showing what is known about the effects of computational modelling on students' conceptual understanding, systemic thinking, views about nature of science and interest in physics, among others variables. Derived from the research results, recommendations are given about the use of computational modelling in physics education and research recommendations are presented with the goal to better understand the interaction between the student and computational modelling process.

Keywords: modelling, software, computational modelling, mechanics

(Some figures may appear in colour only in the online journal)

¹ Author to whom any correspondence should be addressed.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

1. Educational reasons for the use of modelling systems²

Modelling is a central method in physics. In the context discussed here, modelling is defined as the construction of a network of physical concepts and relationships with which the behaviour of a physical system can be described and predicted ([2], p. 23). A modelling system is a computer program with which physical models can be constructed and calculated, and the results of the calculations to be displayed. These are computational models that are quantitative realisations of a mental model. The aim of such models is to understand observed phenomena and processes and to gain new insights into complex interrelationships.

One of the aims of physics teaching is to give students an understanding into how different physical quantities are related. In many cases there is even a chain of (inter-) dependencies that needs to be considered to understand the whole process. For example, the most important connections of Newtonian dynamics could be drawn on the blackboard as shown in figure 1. The ‘dynamic chain’ $F_{net} \rightarrow a \rightarrow v \rightarrow x$ together with $m \rightarrow a$ is also called the ‘standard model’ [3]. This physical structure is used in modelling to illustrate the cause-effect relationship [4] between forces and acceleration and the connection between the other important physical quantities in mechanics, such as acceleration and velocity and velocity and position.

Developing the model shown in figure 1 with learners, facilitates learning processes that help to clarify the learner’s own ideas about the underlying structural relationships of different physical quantities. The calculation and presentation of the model sequence also provides the learners with crucial feedback: does an observed phenomenon or a given prediction correspond to the model sequence? Possible discrepancies can be analysed and discussed: did the learner’s idea match the model, is the model consistent with what is observed, were important aspects such as the direction of forces taken into account? [5].

An often discussed prerequisite for good curriculum design is that physics teaching should focus on students’ everyday experiences. Given the small number of experiments conducted in many physics lessons, students often find it difficult to understand key ideas and retain them over a longer period of time. On the other hand, students already come to class with many prior experiences, especially in the field of mechanics. Here especially, it is important to relate everyday experiences to the physical concepts in the classroom. This is particularly important as students often believe that the concepts taught in physics lessons have nothing to do with their everyday life and only refer to an ideal world or a laboratory context ([6], p. 166). By discussing authentic problems in real-life contexts, however, students can see that physics is relevant to their life, applicable to it, and that physical concepts are useful outside the context of the classroom.

However, in physics, authentic problems are usually also complex problems. In regard to real-life examples in the field of dynamics, this usually means two things: firstly, that several forces act simultaneously. Secondly, that friction must also be taken into account since it plays a crucial role in almost all motion in nature and technology. In physics lessons, however, a variety of idealisations, e.g. the elimination of friction, are used to form theories and to obtain ‘bare phenomena’ in order to develop simple concepts, principles and laws. As in applied physics and engineering, the application of these theories to real-life situations must also be a part of physics lessons. However, if a force and thus the acceleration is dependent on the velocity or the location, this can lead to differential equations which are difficult or impossible to solve explicitly. A promising way to still discuss these real-life situations in the classroom is to use computational modelling, which allows these problems to

² Some ideas discussed in this article are taken from [1].

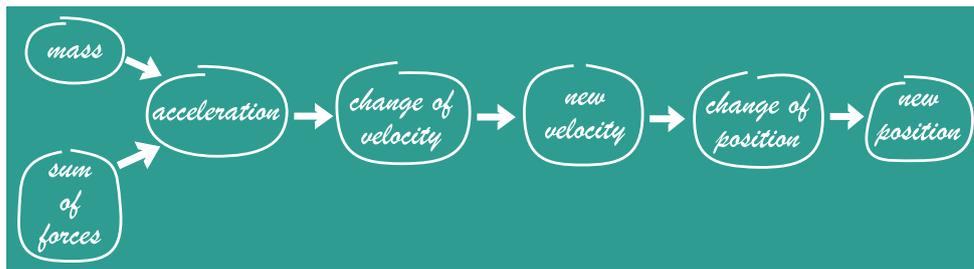


Figure 1. Possible sketch to visualise the relationships between various physical quantities.

be solved numerically. For the reasons previously outlined computational modelling is often used in, but not limited to, Newtonian mechanics. Therefore, this article is focused on this context.

The big advantage of computational modelling is that even very complex phenomena in mechanics can be analysed using only a small number of basic concepts and rules. However, in physics lessons the focus is usually on certain equations, in which a closed form solution is accessible [7], such as $x(t) = x_0 + v_0 \cdot t + \frac{1}{2} \cdot a \cdot t^2$ and $v(t) = v_0 + a \cdot t$ when dealing with uniform acceleration. This limitation often has purely mathematical reasons. In contrast, model-building systems work with basic definitions such as $v = \Delta x / \Delta t$ and $a = \Delta v / \Delta t$ as well as fundamental laws such as $a = \Sigma F / m$. As mentioned before, the idea is to explain a large number of phenomena using only a small number of general laws and rules ([8], p. 153).

If a speed-dependent air friction force is included in the model, a number of interesting examples such as falling cones [9], parachutists, meteors, raindrops or a car accelerating with a constant force can be analysed. Coulomb friction, on the other hand, is not dependent on speed, but only on the direction of motion. A location-dependent force, for example, must be taken into account when a trolley collides with a spring, an athlete bounces on a trampoline or in any oscillations. In particular, the degree of idealisation can be reduced step by step by taking into account initially neglected effects such as friction later in the modelling process. In that way, the user is able to decide autonomously which effects should be included in the model and therefore practices an important scientific skill.

Thus, computational modelling also presents the possibility to include aspects of nature of science in physics classes [10]. Using computer models, learners are able to learn important skills, such as estimation and approximation that are needed to model nature [11]. Learners can create models in a dynamic and iterative process [12] and reflect on scientific reasoning. More generally, computers are increasingly important in scientific research [13] and the role of models in gaining scientific insight can be discussed in class. The interplay between theory, experiment and computation in scientific research [14] can be highlighted using computational modelling.

The term ‘simulation’ on the other hand is used to describe the application or ‘running’ of a model under certain boundary conditions. In order to run a simulation, it is therefore necessary to develop a model first. It is this development of a model, not its simulation, where most of its educational value lies [15–17]. Therefore, the use of computers for simply executing simulations will not be addressed in this article.

2. The history of computational modelling

The development of computational modelling in physics education can be divided into different phases. It is important to note, however, that these phases did not necessarily take place one after another, but rather overlapped.

2.1. Open programming languages

The first phase of computational modelling in physics teaching was defined by the use of personal programming languages. While computers were already successfully used in university introductory physics for simulations and ‘controllable worlds’ in the 1970s [18], computational modelling using programming languages started to be used in physics teaching shortly after [19, 20], mostly using ‘Fortran’ and ‘Basic’ programming languages. Though focussing on simulations and calculations, *Kane* and *Sherwood* also used the mini-language ‘grafit’ during the 1970s to model physical systems [21]. In the 1980s, the first course was created that revolved around the use of programming languages for computer-based model building. The Maryland University Project in Physics and Educational Technology (M.U.P.P.E.T.) used the IBM XT and AT personal computers to analyse and model real physical systems with the ‘Pascal’ programming language with the goal of teaching students the power tools of physics [22]. However, the Commodore 64 was the first computer to be used in physics lessons in secondary schools in the same decade. Teachers were primarily using it in a similar way as the M.U.P.P.E.T. program—namely to calculate motions based on certain mechanical forces. For this purpose, an imperative programming language was used, in which the source code determines what is calculated in which order and how. The programming languages commonly used were also ‘Basic’, ‘Comal’ and ‘Pascal’ as well. These line-oriented programs essentially consisted of a loop, where a single pass corresponds to a single time step Δt . In addition to the calculation of the single forces and the resulting total force F_{net} , each loop contains the following steps:

$$a = F_{net}/m;$$

$$\Delta v = a \cdot \Delta t; \quad v = v + \Delta v;$$

$$\Delta x = v \cdot \Delta t; \quad x = x + \Delta x;$$

$$t = t + \Delta t;$$

Location and velocity are determined by adding up the individual changes based on a start value (see figure 2).

This simple numerical method is also called the ‘Euler method’ [23]. In many cases though, such as oscillations, it can lead to major errors after a relatively short time.

Soon other programs like ‘Dynamos’ were developed for the operating system MS-DOS. A different contemporary approach was to program pocket calculators to numerically solve the (differential) equations of motion of physical systems [24].

2.2. Spreadsheet processing

The use of spreadsheet programs such as ‘Excel’ can also be regarded as a variant of the use of personal programming languages. VisiCalc appeared on the Apple II in 1979 [25] and spreadsheets were used in education as early as 1984 [26]. Their main advantage is that users do not need to learn a programming language to use them [27]. Therefore, the focus can remain on teaching physics with computing—not the other way around [28]. Analogous to

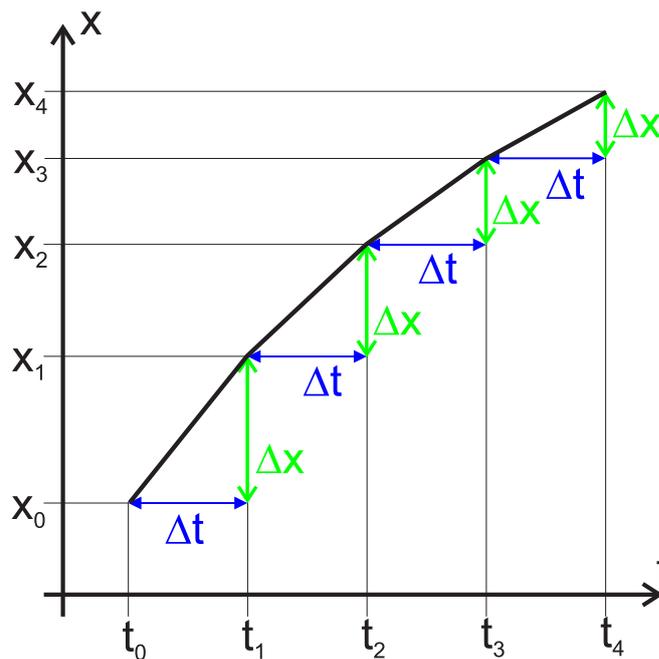


Figure 2. Basic principle of the ‘Euler method’.

the programmatic method outlined in the previous section, a row in the spreadsheet corresponds to one time step. In the columns, different physical quantities are calculated from other quantities of the row or those of the previous row. By adding further rows to the spreadsheet, the motion path can be calculated over time. This process highlights the iterative nature of numerical calculations. With spreadsheet software it is also possible to show the results graphically. However, the use of spreadsheets for this purpose is not intuitive and requires a good grasp of the software. Also, the result of the simulation is a table full of numbers with no overview of the calculations, which makes both finding errors and modifying the calculation difficult [29]. Nonetheless, spreadsheet processing has been a popular way of implementing computational modelling in physics teaching in various contexts [30].

2.3. Graphical modelling

In the 1990s, graphic-structure-oriented modelling systems became increasingly popular. Based on ‘STELLA’, further programs such as ‘Powersim’ were brought to market. The programs ‘Dynasys’, ‘Modus’, ‘Coach’ as well as ‘VisEdit’ were specifically developed for educational purposes. With these programs, a model is first created with a graphical model editor by introducing the relevant physical quantities and their relationships. Specifically, symbols for the individual quantities are set and linked according to their corresponding physical relationships (see figure 3, analogous to figure 1). Only in a second step are these qualitatively defined relationships between the individual physical quantities then quantified using fundamental physical relationships ([31], p. 151). The graphical representation has many advantages over equations. First of all, the structure of the model can be recognised quickly and easily, which makes it easier to discuss it in group work, for example. In addition, it is easy to immediately recognise how the different quantities affect each other. Furthermore,

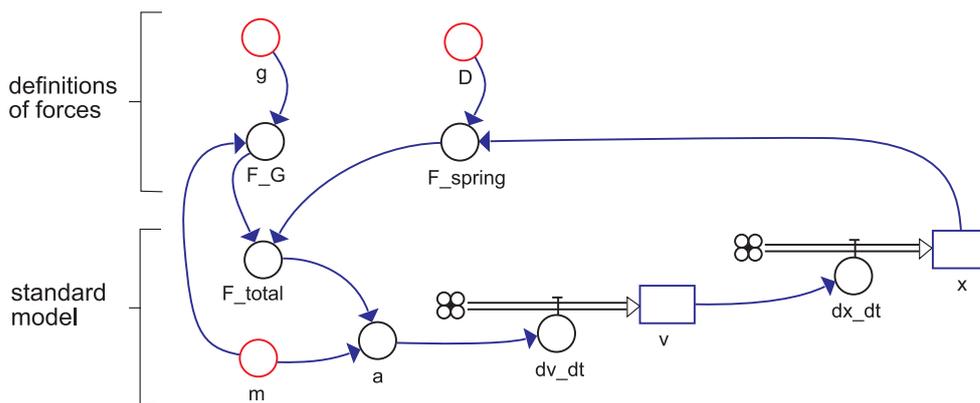


Figure 3. Graphical model of a vertical spring oscillation in ‘Dynasys’.

learners only have to be able to manipulate a few symbols and not learn an additional programming language.

These modelling systems go back to Forrester [32], who developed System Dynamics in the 1950s. In System Dynamics, three fundamentally different types of system parameters can be distinguished:

1. Default parameters, i.e. parameters and exogenous influences, that is, values that act on the system from the outside but are not influenced or changed by it (red circles in figure 3).
2. State parameters (‘stocks’), i.e. memory variables in which the current state of a system is expressed, based on its previous development (blue rectangles in figure 3). These parameters represent the memory of the system.
3. Intermediate parameters, i.e. parameters that can be calculated at any time directly from default and state parameters (black circles in figure 3).

The programs ‘STELLA’, ‘Powersim’, ‘Dynasys’ as well as ‘Coach 6’ and ‘Coach 7’ and others use a ‘flow analogy’ to illustrate these relationships: changes ‘flow’ through a pipeline into a ‘stock’, while the flow (the rate of change) can be regulated by valves (a comparison sometimes referred to as ‘stock-and-flow model’ or ‘flow diagram’). This ‘flowing into the stock’ corresponds to integrating an analytical solution. If a variable is both a state variable and a rate of change for another state variable, it must be entered twice, which can be confusing for learners [[2], p. 33–34]. This is the case, for example, with velocity and acceleration in figure 3.

Tinker ([33], p. 98f) criticises that this ‘flow analogy’ requires an intuitive understanding of the flows of incompressible fluids and points out that it is particularly problematic in the case of Newton’s second law (What is it that ‘flows’ controlled by the ‘acceleration’ into the stock ‘velocity’?). Sander ([34], p. 197), on the other hand, claims that students have little difficulty with the analogy itself, but rather with the software STELLA ([34], p. 179). The programs ‘VisEdit’, ‘Modus’, ‘Moebius’ and ‘Coach 5’ use another analogy: A syringe is used to make changes to a stock, which can be positive or negative, while the visualisations used by ‘STELLA’ misleadingly suggest that the flow can only enter the stock from the source. Other programs like ‘IQON’ (Interacting Quantities Omitting Numbers) go even further and only focus on the qualitative aspects of modelling. The software helps the user to formulate causal relationships between different physical quantities by expressing these

```

1 from vpython import *
2 scene.background=vector(0.8,0.8,0.8)
3 f1 = gcurve(color=color.orange)
4 cone = cone(pos=vector(0,1,0),axis=vector(0,-0.1,0),radius=0.1)
5 vcone = arrow(pos=cone.pos, axis=vector(0,0,0), shaftwidth=0.015,
6             color=color.orange)
7
8 cone.velocity = vector(0,0,0)
9 deltat = 0.005
10 t = 0
11 m = 0.00048
12 k = 0.0019
13
14 while cone.pos.y>0:
15     rate(100)
16     f1.plot(pos=(t,cone.velocity.y))
17     Fg=-m*9.81
18     Ffric=k*cone.velocity.y**2
19     Fnet=Fg+Ffric
20     cone.velocity.y = cone.velocity.y + Fnet/m*deltat
21     cone.pos = cone.pos + cone.velocity*deltat
22     vcone.pos = cone.pos
23     vcone.axis = cone.velocity/5
24     t = t + deltat

```

Figure 4. Code in VPython for a falling cone.

relationships as strong and weak links. It can therefore be helpful for younger learners and serve as a preparation for the quantification of models [3].

2.4. Output of animations

Students find it difficult to interpret graphs [35–37]. It can therefore make sense to not only show the results of a calculation in the form of a diagram, but also in an animation. When simulating the model in a modelling software, the animation immediately reveals an incorrect input without having to interpret the corresponding graph.

For this reason, software products such as ‘VisEdit’ (with ‘PAKMA’) and ‘JPAKMA’ were developed in the 1990s and 2000s. Similar to the newer software ‘Modellus 4’, these programs were specifically designed for teaching purposes in schools. Other programs like ‘VPython’ [38] and ‘Easy Java Simulations’ [39] can also be used for this purpose. ‘VPython’ was built to let the user create 3D objects and animations easily, whereas Easy Java is a modelling tool that allows the user to create scientific simulations, while reducing the amount of programming necessary to implement an idea. Today ‘VPython’ seems to be the most used tool for computational modelling in universities because it teaches basic programming skills in Python and combines that with an easy 3D animation output [7]. It is, however, also possible to utilise ‘VPython’ in schools [40].

‘VPython’ requires the user to write the code by themselves. In the example shown below (see figure 4), a falling cone is modelled with air friction. In the first section, the background colour is specified, the plot and the cone and its velocity vector are defined. The cone is a 3D object that is part of the VPython module. The second section includes

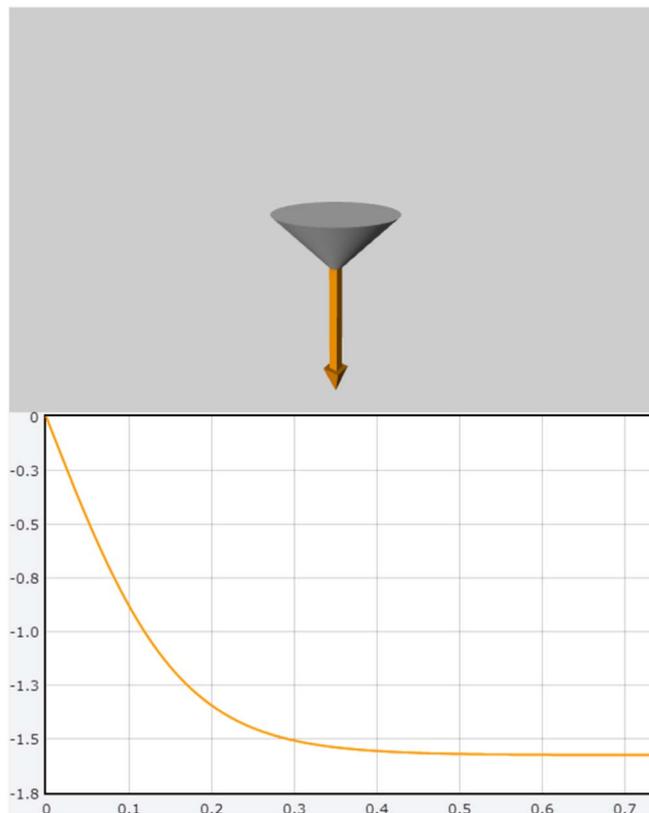


Figure 5. Animation and t - v_y -diagram in VPython for a falling cone.

the starting conditions and parameters that influence the motion. In the while loop, the velocity is plotted, the forces are defined and the calculation of motion takes place.

The use of VPython is similar to the use of the personal programming languages described above. The biggest difference is the possibility to create 3D animations easily. The code produces an animation that is shown in figure 5. While the cone is moving towards the ground, its velocity vector is embedded in the animation and an additional graph of the y -component of the velocity appears simultaneously with the motion of the cone.

3. Computational modelling schools today

3.1. Current expectations for good software

Today, software is expected not only to run stably, but to also be highly intuitive to use. If modelling software is difficult to use, the problem is shifted from a computational level to a technical level and the original physical question might easily get lost. Programming languages, spreadsheet programs, graphic-oriented programs as well as the creation of animations require users to invest a significant amount of time to familiarise themselves with the product and, while programming can teach students useful skills needed for a physics major [7], it is often impractical to utilise open programming languages in secondary schools. Unfortunately, this is probably one reason why modelling software is rarely used in schools

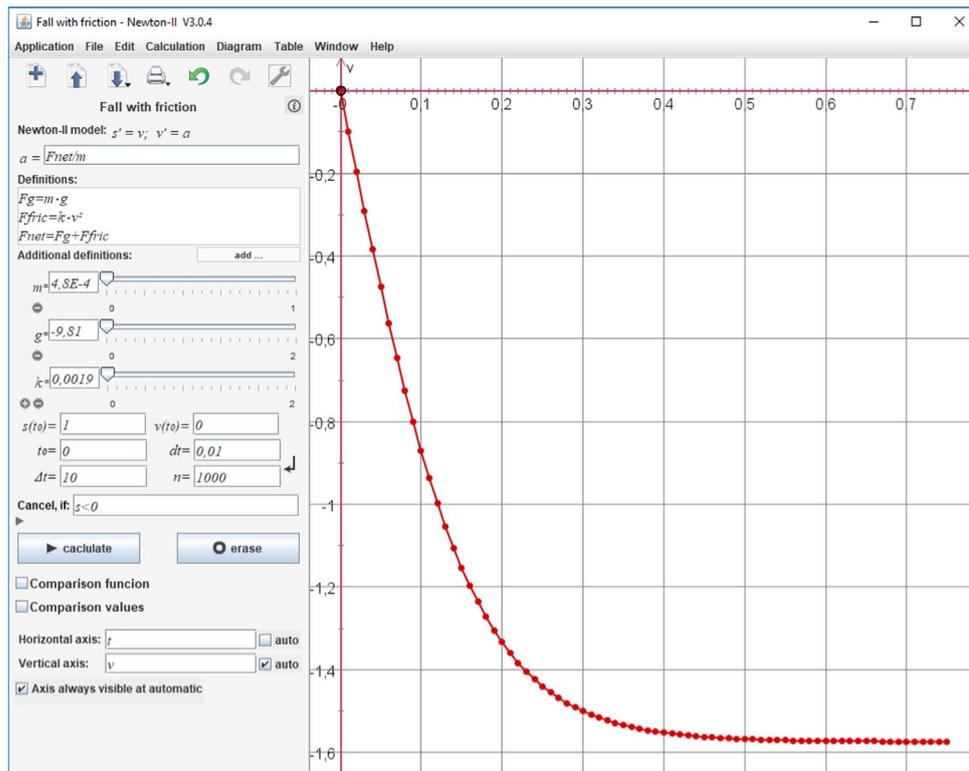


Figure 6. Fall with friction, modelled in Newton-II.

today and why ready-to-use animations tend to be more popular with teachers [41]. As a result, unfortunately, important phenomena such as friction are often not covered at all in traditional physics lessons. This can easily be changed by using modern modelling software.

Both ‘Newton-II’ and ‘Fluxion’ (available free of charge [42]) can be used on all standard operating systems and are designed as one-window applications. This means that all important elements of the programs except a few dialogue boxes for special settings are displayed in one window. In the input and action area on the left, all equations, constants, calculation conditions as well as axis settings can be entered in order to start the calculation (see figure 6). In the display area on the right, a graphical visualisation of the solution can be found. The program is designed to display the equation defining the force, the parameters and the output as a graph in one window (spatial contiguity) [43]. The trajectory is calculated immediately and displayed simultaneously with all inputs and parameter values as a graph (temporal contiguity) [43]. This allows a fast and uncomplicated variation of the different parameters in real time. In addition, the software offers an intuitive way to make certain changes to the graph: the axes can be stretched or compressed by clicking and moving the mouse with the mouse button pressed or by simply turning the mouse wheel.

‘Newton-II’ is specifically designed for problems in mechanics. It is therefore not necessary for the user to specify how the velocity and the location are obtained from the acceleration. Instead, the user only needs to specify how the acceleration is calculated. Since the calculation of velocity and location are already a fixed component of the software, users can fully concentrate on the modelling.

Table 1. Software already used at least once by region ($N_{\text{Frankfurt}} = 51$; $N_{\text{Franconia}} = 60$; $N_{\text{Swabia}} = 50$; tested for significant differences using the χ^2 test, *: $p < 0.05$, **: $p < 0.01$, +++: $p < 0.001$).

	City of Frankfurt	Lower Franconia	Bavarian Swabia
Spreadsheet (e.g. Excel)	47%*	68%*	60%
Dynasys	4%	2%	8%
Newton-II	2%**	22%** +++	0%+++
VisEdit with PAKMA	2%	13%*	2%*

The model-building module of the video analysis software ‘Tracker’ [44] goes one step further. Here, only the mass and the resulting total force for each component have to be specified, while the software automatically calculates the resulting acceleration. In other words, the entire ‘standard model’ F_{net} and $m \rightarrow a \rightarrow v \rightarrow x$ is already part of the software.

3.2. Software currently used in school

Between November 2014 and February 2015, a survey was conducted in three regions of Germany using an online questionnaire. The objective was to find out which modelling software is used today in the respective regions [41]. It should be noted here that numerical calculations are a mandatory part of the curriculum in Lower Franconia and Bavarian Swabia. In total, $N = 163$ participants took part in the survey, well distributed between the three regions.

As shown in table 1, the number of different pieces of software used in schools is rather limited. The frequent choice of spreadsheet programs is in contradiction to the recommendations in the educational literature. However, it is unclear whether the teachers who stated that they used ‘Excel’ in their teaching were actually using it for computational modelling or for different purposes. The significantly higher popularity of ‘Newton-II’ in Lower Franconia compared to Frankfurt or Swabia is probably due to the fact that the software was developed in that region.

4. Research results on computational modelling in literature

As already mentioned, the educational reasons to include computational modelling in the physics curriculum are varied. Therefore, different research has been conducted to investigate the effects of computational modelling in schools and universities, ranging from aspects of nature of science to systemic thinking and content knowledge. A brief overview of the findings is given here.

There were two major course structure changes to introductory physics in the US in the 1980s that included computational modelling. The M.U.P.P.E.T. project was a curriculum reform study that included computers to analyse real physical systems. By including student programming in Pascal in introductory physics in the late 1980s, Redish and Wilson [11] were able to change the curriculum in a way that professional skills, such as theoretical and modelling skills, could be introduced at an earlier stage, which lead to a significant increase in valuable research projects carried out by their students.

In contrast, the creators of the Workshop Physics decided not to include a programming languages in their program in order concentrate on physics rather than computation. They used the computer in various ways and restructured introductory physics completely, with the

addition of mathematical modelling using spreadsheet software. There have been different evaluations, showing that the conceptual understanding of mechanics was higher than in the traditional approach [45] and the interest of students in studying physics improved with no deterioration relative to the classical approach in the ability to solve classical textbook problems [46]. Due to the extensive changes made, it is unclear which part of the computational modelling was most significant in the achievements of Workshop Physics.

According to Schecker, Klieme *et al* ([47], p. 21), the studies in the 1990s on model-building systems carried out in Germany were less focussed on conveying subject-specific content knowledge, rather on promoting skills like systemic thinking in general ([48]; [49], p. 337). While some results from laboratory studies suggest that such software could be used to teach systemic thinking in the classroom, field studies in various subjects [50] showed no such effects ([47], p. 21).

Since then, modelling systems are primarily used to teach physical content knowledge in schools. Bethge and Schecker conducted several months of testing in various physics courses (grades 11 to 13) in Bremen [2, 31, 34, 51]. The only major study in Germany on the teaching of physical content knowledge in schools is the project ‘Physics Learning with Modelling Systems’, funded by the German Research Foundation (DFG), which was carried out between 1996 and 1999 at the University of Bremen and the Institute for Educational Research in Bonn ([47], p. 3; [52], p. 230). In two advanced physics courses in the eleventh grade, the STELLA modelling software was used during a fifth of the teaching time—mainly in small groups working with computers. In contrast, the two advanced courses of the control group had no computer use in their lessons. One hypothesis was that students who have worked with the modelling software achieve a higher level of conceptual physical competence than their traditionally taught peers. In particular, the assumption was that this higher competence would translate into a higher ability for conceptual-qualitative and semi-quantitative analyses. However, this hypothesis was not confirmed in the study ([47], p. 11; [53], p. 87).

A further hypothesis was that students who repeatedly used the modelling software in the field of mechanics would be able to apply the problem-solving strategy learned through modelling to new situations that do not necessarily require any modelling. To test this hypothesis, experimental interviews were conducted, in which the students were asked to describe and explain the motion of a cart in an experimental demonstration. The results showed that the physically correct argumentation regarding mechanical forces was significantly more frequent in the experimental group than in the control group ([47], p. 12; [54], p. 73). However, the final survey at the end of the eleventh grade showed no significant advantage of the experimental group in regard to non-mechanical forces, which were taught in the second half of the eleventh grade ([47], p. 12; [54], p. 71 + 73). Encouragingly though, the study confirmed the hypothesis that students in the experimental group use familiar substructures like the standard model when modelling new tasks in mechanics ([47], p. 13). However, the hypothesis that students in the experimental group have higher abilities in systemic thinking in the domain of mechanics compared to their traditionally taught peers was not backed by the study.

Overall, the DFG study has shown that the development of Newtonian ideas through teaching with the STELLA model-building software is not improved to the extent that was previously expected. Although semi-quantitative skills for describing and predicting motion sequences were promoted, no differences to traditionally taught classes could be found with regard to students’ basic understanding of key concepts of Newtonian mechanics and equation-oriented quantitative tasks ([47], p. 25). According to this study, the effects on the

development of systemic thinking are also limited to the semi-quantitative domain. Schecker, Klieme *et al* therefore come to the following conclusion: '*In the teaching concept investigated, model-building systems have only proven to be effective to promote a better physical understanding, but not as a means of promoting overarching competences*' ([47], p. 25). While graphic-oriented modelling software can therefore promote a general engagement with physics and its methods, the effects of modelling on an increase in content knowledge remain limited ([55], p. 151). However, a strong argument in favour of using modelling software in mechanics teaching is that it supports students in developing a Newtonian perspective on mechanical forces.

In a study by Sander [34] of 13 pre-service physics teachers in a German university, the modelling software STELLA was used in connection with real experiments during eight two-hour laboratory sessions during the students' first semester. It was found that students tended to discuss theoretical aspects more when working with the modelling software compared to them just conducting experiments. It has to be noted, however, that the students discussed even more theoretical aspects during the usual lab interview with their supervisor ([34], p. 121). Overall, the use of the modelling system promoted a conceptual-qualitative study of physics ([34], p. 215). Modelling has been shown to be a promising way to consolidate and deepen knowledge, but not to develop new conceptual knowledge. Although individual ideas and strategies were pursued, the conscious formulation of hypotheses, for example, occurred only to a limited extent. In contrast to original expectations, however, the intensive interaction between modelling and experiment was not adequately stimulated. In the study, this is explained by limitations of the software. In particular, it was criticised that it was not possible to overlap the measured values and simulation results in the same window using the modelling software STELLA. It is assumed that this limitation of the software particularly favoured superficial comparisons regarding the curve progression ([34], p. 243).

Another study, in which six students in the experimental group worked on two experiments with the STELLA modelling software, was carried out by Hucke and Fischer [56]. It was shown that the students are more concerned with physical relationships when modelling, but that there were only few changes in concept maps ([56], p. 252). A study by Tinker [33] furthermore shows that it also plays a role where modelling software is used. In the study, the modelling software STELLA was used in mathematics for students aged 14 and over to introduce basic concepts of differential and integral calculus. The study reports difficulties in dealing with the terms state and rate and in interpreting the resulting graphs, and therefore concludes that STELLA is not an ideal tool for learning these concepts. For physics, the author of the study expects problems with inflows and outflows that are controlled by valves, since they have no concrete meaning in physics (what flows into the stock 'velocity'?).

Using teacher-centred instruction, Wilhelm ([1], p. 64–83) used computational modelling (with 'VisEdit') in several classes in grade eleven after a traditional mechanics course. It was found that the students concentrated on specific motion functions, but did not know the basic definitions $v = \Delta x / \Delta t$ and $a = \Delta v / \Delta t$. Moreover, they did not know how to deal with multiple forces, since the equation $F = m \cdot a$ was interpreted as having only a single acting force. In addition, they always indicated forces without the necessary sign to indicate their direction. However, the use of the software was highly praised by the students. Moreover, a survey showed that they were more likely to think that physics had something to do with their everyday life. Significant changes could also be seen in the concept maps that students were asked to draw before and after the teaching unit. Before the teaching unit, velocity was the primary physical quantity for the students and only a few of them were able to correctly state what quantities determine the acceleration. In addition, one third wrongly stated that the acceleration and mass had an effect on the force, which was probably wrongly inferred from the formula $F = m \cdot a$. After the teaching unit, acceleration became the key quantity in the

students' concept maps. The 'sum of the acting forces' was also important and the students stated much more frequently that forces and mass determine acceleration. Based on these findings, it can be assumed that the students' developed a deeper structural understanding of dynamics. In another study ([1], p. 211–215), in which computational modelling was used in mechanics lessons in grade eleven alongside the content, similar positive results were reported.

The Modeling Instruction curriculum is a research-based program for high school science education reform supported by the National Science Foundation in the United States that tried to make school physics more student-centred while using the computer as an essential scientific tool [57]. Scientifically testing the limits of physical models was a focus of the curriculum. Data on 20 000 students showed that the emphasis on models and its focus on inquiry helped the students to acquire a better understanding of physics than in the traditional approach [57], also encouraged them in participating in class [58], improved their perception of the nature of science [59] and promoted their self-efficacy [60]. While numerical computation and computational modelling was originally not part of the curriculum Caballero *et al* successfully included real-world problems in the framework by using 'VPython' with ninth-graders [40]. They found that high school students are able to engage in computational thinking in the context of physics and are capable of using numerical computation. Due to the time students had to spend to learn and relearn 'VPython' they recommend a tighter integration in each modelling cycle of the Modeling Instruction curriculum and plan to provide more scaffolding. They also found that the students' success was closely tied to their ability to connect physics and computation knowledge [61], meaning that students that presented both an iterative-local and a force-causal view in the interviews were most likely to succeed in creating a working model. One third of the students were able to construct a model of a new physical system and thus complete the programming assignment. It remains to be seen if a more scaffolded code and better integration into the course lead to a higher rate of success for high school students.

In another study with 1357 students in university introductory physics also using the 'VPython' environment in 2012, where 11 out of the 13 mechanics laboratories included a computational modelling activity and students additionally solved 13 computational modelling homework problems, Caballero *et al* found that after completing said tasks 60.4% of all students were able to model a novel problem successfully [62]. A detailed cluster analysis additionally showed that the most common problems while creating the model were related to calculating the net force acting on the object. In a different study of Caballero and Pollock students carried out self-chosen modelling projects with Mathematica and remarked that the transition from physics on paper to physics on a computer was very important for their understanding of physics [63]. Additionally, most students saw the computational tasks as a boost for their confidence and motivation.

Based on these positive results, Irving *et al* developed a practice focused learning environment (P^3) for university students structured around computational modelling [64] that is built on matter and interactions (M and I) [65]. P^3 introduces modern problem-solving tools and focuses on core fundamental principles instead of certain seemingly isolated formulas. The course consists of 30 complex real-world problems. Computational modelling is used in seven of them. Their first results showed an average normalised gain of 0.60 in the FMCE test for the P^3 environment, while traditional introductory physics lead to an average normalised gain of 0.10–0.35. They also reported a slightly positive shift in CLASS-scores, which measures the cognitive attitudes towards nature of science. It is also used as an indicator whether students like physics. CLASS-scores normally tend to go down during other introductory physics courses [64]. Other universities successfully used 'Modellus' to create a new

course based on computational modelling [66]. The use of 'Modellus' also improved the performance in the interpretation of kinematics graphs [67]. Burke and Atherton designed a project-based computational physics course that was focused on expert practice using Mathematica and Python and analysed the initial implementation. They found that their students drastically improved in carrying out these projects during the semester and rated the course highly [68]. Similarly, in an interview study carried out by Hawkins *et al* most of the participating students held the opinion that their use of 'VPython' to model three different problems helped them to learn physics [69]. Furthermore, they structured students' statements regarding the theme 'Computation helps to Learn Physics' into categories. The most popular statements were that the practice of computation ('thinking like a physicist') helps to learn physics and computation helps to build a conceptual understanding of physics.

Work from Benacka investigates the motivating effects of using spreadsheet modelling of real-world physics problems [70, 71]. For example, 97% of high school students that modelled one of three mechanics problems, that are normally not part of high school physics, with 'Excel' using the Euler Method found the lesson very interesting [72].

5. Recommendations for implementing computational modelling

As reported, there have been many different attempts to implement computational modelling in schools and universities. Based on the research results it is a reasonable assumption to make, that computational modelling can be used to contribute to an authentic and successful physics curriculum.

Due to the vastly different goals that are pursued in the different settings where computational modelling can be implemented (i.e. high school physics and introductory physics in university) there is no one optimal solution for computer-based modelling work in physics classrooms.

Regarding the software, research shows that students in secondary schools have more difficulties using open programming languages than university students and the time to learn to program might not always be available. Therefore, easy-to-use software is needed, which enables high school students to focus on physics rather than computing. Therefore, it seems reasonable to recommend software that does not require the knowledge of a computer language [3]. Although there is software that tries to expand on the shortcomings of older programs, there are no recent studies of their usability for school pupils. In contrast, at a university level, programming environments like 'VPython' have proven to be successful, specifically in introductory physics. Furthermore, as programming skills are becoming increasingly important for further study in physics and engineering, this supports the implementation of them in the early undergraduate curriculum.

Generally, there seems to be the consensus that computational modelling is a useful tool to tackle authentic and complex problems. Therefore, it is necessary to not only include the computer in traditional physics classes but to restructure the curriculum to make optimal use of the possibilities of computational modelling. Then, it is possible to improve conceptual and content knowledge of students and also to create an interesting physics curriculum that highlights aspects of nature of science and how modern scientists work.

6. Desiderata for further research

While there have been some curriculum changes, especially in universities, using computational modelling that have already produced promising results, the work yet to be done is

twofold. Firstly, more research in secondary schools is necessary. There are very few recent research projects with school pupils and, due to the different goals and constraints, research results in universities are only partly applicable to secondary schools. The existing studies highlighted some of the problems that exist using modelling systems in schools and partly explained the rather small effects with some shortcomings of the then existing software. Research is therefore needed to find out if recent software, which tries to improve upon the reported difficulties, is better suited to be used by pupils in schools.

Secondly, there is a need to understand how and why students learn while using the computer for modelling physical systems. Though there have been quite a few studies that showed some results of the use of the computer for modelling a physical system, it is not yet known how exactly the interaction between student and computer works. Conceptual knowledge of physics, a set of mathematical tools and knowledge about the computational algorithms all play an important role in computational modelling. It is of great interest to find out how these parts interact, while the student learns [63]. This knowledge would help to better design programs and curricula that are well suited for their respective audiences.

7. Summary

Computational modelling is the construction of a network of physical quantities using the computer. It therefore serves a different purpose in teaching than the usage of ready-to-use animations. Computational modelling itself is an old idea in mechanics teaching, however the way it is implemented in physics lessons has changed. These changes were partly driven by research that suggested students to be more active in their learning process. The advent of new software also made better implementations of computational modelling in physics classes possible. Research results on computational modelling show that it can promote the understanding of Newtonian mechanics, although some far-reaching expectations have not been fulfilled, especially in secondary schools. While some studies came to the conclusion that students better understand the relationships of physical quantities in Newtonian mechanics through computational modelling, it remains unclear, if the understanding of the physical quantities themselves can also improve. Generally, it seems to be clear that computation alone does not guarantee a better learning experience for students.

In particular, different programs have different strengths and weaknesses. Programming environments like ‘VPython’ seem to be a good choice for university students, whereas programs designed for secondary schools try to reduce requirements needed to use said software. Further research is therefore necessary to analyse whether improved software, which are more intuitive to use and allow for example comparisons with measurement data such as video analysis, can better foster learning in the physics classroom.

Acknowledgments

We thank Jan-Philipp Burde and Thomas Weatherby for their help reviewing the paper in English.

ORCID iDs

Jannis Weber  <https://orcid.org/0000-0001-9874-235X>

Thomas Wilhelm  <https://orcid.org/0000-0002-7533-7009>

References

- [1] Wilhelm T 2005 Konzeption und Evaluation eines Kinematik/Dynamik-Lehrgangs zur Veränderung von Schülervorstellungen mit Hilfe dynamisch ikonischer Repräsentationen und graphischer Modellbildung *Studien zum Physik- und Chemielernen* Vol 46 (Berlin: Logos-Verlag) (<https://opus.bibliothek.uni-wuerzburg.de/frontdoor/index/index/docId/3310>)
- [2] Schecker H 1998 *Physik-Modellieren. Grafikorientierte Modellbildungssysteme im Physikunterricht* (Stuttgart: Klett-Verlag)
- [3] Schecker H 1993 The didactic potential of computer aided modelling for physics education In: ed D Ferguson *Advanced Educational Technologies for Mathematics and Science NATO ASI Series, Serie F: Computer and Systems Sciences* Vol 107 (Berlin, Heidelberg, New York: Springer) S 165–207
- [4] Reusch W, Gößwein O and Heuer D 2000 Grafisch unterstütztes Modellieren und Messen—VidEdit und PAKMA In:*Praxis Der Naturwissenschaften—Physik* **49** 32–26
- [5] Lück S and Wilhelm T 2011 Modellierung physikalischer Vorgänge am computer. Modellbildungssysteme als Unterstützung zum Verständnis physikalischer Strukturen In:*Unterricht Physik* **22** 26–31
- [6] Schecker H 1985 Das Schülervorverständnis zur Mechanik, Eine Untersuchung in der Sekundarstufe II unter Einbeziehung historischer und wissenschaftlicher Aspekte *Dissertation* Universität Bremen
- [7] Chabay R and Sherwood B 2008 Computational physics in the introductory calculus-based course — In:*Am. J. Phys.* **76** 307–13
- [8] Bethge T 1992 Mechanik in der Sekundarstufe II—Ein Kurskonzept unter Nutzung von Software-Werkzeugen *Zur Didaktik der Physik und Chemie, Probleme und Perspektiven, Vorträge Auf Der Tagung Für Didaktik Der Physik/Chemie in Hamburg* ed K H Wiebel (Alsbach: GDCP, Leuchtturm-Verlag) pp 152–4
- [9] Wilhelm T 2000 Der alte Fallkegel—modern behandelt In:*Praxis der Naturwissenschaften—Physik* **49** 28–31
- [10] Buffler A, Pillay S, Lubben F and Fearick R 2008 A model-based view of physics for computational activities in the introductory physics course In:*Am. J. Phys.* **76** 431–7
- [11] Redish E and Wilson J 1993 Student programming in the introductory physics course: M.U.P.P.E. T. In:*Am. J. Phys.* **61** 222–32
- [12] Mikelskis-Seifert D and Kaspar L 2011 Modellieren in der Physik—im Alltag und im Unterricht In:*Naturwissenschaften im Unterricht* **22** 4–12
- [13] Caballero M and Merner L 2018 On the prevalence and nature of computational instruction in undergraduate physics programs across the United States *Phys. Rev. Phys. Educ. Res.* **14** 020129 In:
- [14] Duarte Toedoro V and Gomes Neves R 2010 Mathematical modelling in science and mathematics education In:*Computer Physics Communication* **182** 8–10
- [15] Schecker H 1993 Learning physics by making models In:*Phys. Educ.* **28** 102–6
- [16] Guzdial M 1994 Approaches to classroom-based computational science *Recreating the Revolution. Proc. of the Annual National Educational Computing Conf. (Boston)* pp 217–25
- [17] Hücke L 1999 Handlungsregulation und Wissenserwerb in traditionellen und computergestützten Experimenten des physikalischen Praktikums *Studien zum Physik- und Chemielernen* Vol 8 (Berlin: Logos-Verlag)
- [18] Bork A 1978 Computers as an aid to increasing physical intuition In:*Am. J. Phys.* **26** 796–800
- [19] Peters H 1980 The CONDUIT series in physics In:*Comput. Educ.* **4** 1–9
- [20] Bork A 1980 Physics in the Irvine educational technology center In:*Computers & Education* **4** 37–57
- [21] Kane D and Sherwood B 1980 A computer-based course in classical mechanics In:*Comput. Educ.* **4** 15–36
- [22] MacDonald W, Redish E and Wilson J 1988 The M.U.P.P.E.T. manifesto In:*Computer in Physics* **2** 23–30
- [23] Greenberg M D 2012 Ordinary differential equations. Retrieved from <https://ebookcentral.proquest.com>
- [24] Eisberg R 1978 *Mathematische Physik für Benutzer programmierbarer Taschenrechner—Mit 16 Programmen, 29 Beispielen und 92 Aufgaben* (München: R. Oldenbourg Verlag GmbH)
- [25] Baker J and Sugden S 2003 Spreadsheets in education—the first 25 years In:*Spreadsheets in Education (eJSiE)* **1** 18–43

- [26] Arganbright D 1984 The electronic spreadsheet and mathematical algorithms *The College Mathematical Journal* **15** 148–57
- [27] Hsiao F 1985 Micros in mathematics education—uses of spreadsheets in CAL In:*Int. J. Math. Educ. Sci. Technol.* **16** 705–13
- [28] Laws P 1991 Calculus-based physics without lectures In:*Phys. Today* **44** 24–31
- [29] Wilhelm T 2018 Zur Geschichte der mathematischen Modellbildung im Physikunterricht In:*Plus Lucis* **4-2018** 4–10
- [30] Herman R 2009 Spreadsheet physics: examples in meteorology and planetary science In:*Am. J. Phys.* **77** 1124–9
- [31] Schecker H 1992 Computer als Werkzeuge im Physikunterricht *Zur Didaktik der Physik und Chemie, Probleme und Perspektiven, Vorträge Auf Der Tagung für Didaktik Der Physik/ Chemie in Hamburg, September 1991, Gesellschaft für Didaktik der Chemie und Physik (GDCCP)* ed K H Wiebel (Alsbach: Leuchtturm-Verlag) pp 149–51
- [32] Forrester J W 1961 *Industrial Dynamics* (Cambridge, MA: MIT Press)
- [33] Tinker R F 1993 Modelling and theory building: technology in support of student theorizing ed D L Ferguson *Advanced Educational Technologies for Mathematics and Science NATO ASI Series, Serie F: Computer and Systems Sciences* Vol 107 (Berlin, Heidelberg, New York: Springer) pp 91–113
- [34] Sander F 2000 Verbindung von Theorie und Experiment im physikalischen Praktikum. Eine Empirische Untersuchung zum handlungsbezogenen Vorverständnis und dem Einsatz grafikorientierter Modellbildung im Praktikum *Studien zum Physiklernen* Vol 13 (Berlin: Logos-Verlag)
- [35] McDermott L, Rosenquist M and van Zee E 1987 Student difficulties in connecting graphs and physics: Examples from kinematics In:*Am. J. Phys.* **55** 503–13
- [36] Bleichner R 1994 Testing student interpretation of kinematics graphs In:*Am. J. Phys.* **62** 750–62
- [37] Thornton R and Sokoloff D 1998 Assessing student learning of Newton's laws: the force and motion conceptual evaluation and the evaluation of active learning laboratory and lecture curricula In:*Am. J. Phys.* **66** 338–51
- [38] Scherer D, Dubois P and Sherwood B 2000 VPython: 3D interactive scientific graphics for students In:*Computing in Science & Engineering* **2** 56–62
- [39] Christian W and Esquembre F 2007 Modelling physics with easy java simulations In:*The Physics Teacher* **45** 475–80
- [40] Caballero M, Burk J, Aiken J, Thoms B, Douglas S, Scanlon E and Schatz M 2014 Integrating numerical computation into the modelling instruction curriculum In:*The Physics Teacher* **52** 38–42
- [41] Wenzel M 2018 Computereinsatz in Schule und Schülerlabor. Einstellung von Physiklehrkräften zu Neuen Medien *Studien zum Physik- und Chemielernen* Vol 251 (Berlin: Logos-Verlag)
- [42] <https://did-apps.physik.uni-wuerzburg.de/Newton-II> und <https://did-apps.physik.uni-wuerzburg.de/Fluxion>
- [43] Mayer R 2017 Using multimedia for e-learning In:*J. Comput. Assist. Learn.* **33** 403–23
- [44] <https://physlets.org/tracker/>
- [45] Laws P, Willis M and Sokoloff D 2015 Workshop physics and related curricula: a 25-year-history of collaborative learning enhanced by computer tools for observation and analysis In:*The Physics Teacher* **53** 401–6
- [46] Laws P 1991 Workshop physics: learning introductory physics by doing it In:*Change: The Magazine of Higher Learning* **23** 20–7
- [47] Schecker H, Klieme E, Niedderer H, Ebach J and Gerdes J 1999 Physiklernen mit Modellbildungssystemen. Förderung physikalischer Kompetenz und systemischen Denkens durch computergestützte Modellbildungssysteme, Abschlussbericht zum DFG-Projekt, Institut für Didaktik der Physik an der Universität Bremen und Max-Planck-Institut für Bildungsforschung Berlin (http://www.idn.uni-bremen.de/pubs/DFG_PMS_Ab.pdf)
- [48] Hassell D J 1987 The roll of Modelling activities in the humanities curriculum, with special reference to geography: an investigate study *Dissertation for Associateship for Education* London, King's College
- [49] Ossimitz G 1996 Können Schüler im Mathematikunterricht systemisch denken lernen? ed K P Müller *Beiträge Zum Mathematikunterricht—Vorträge auf der 30. Bundestagung für Didaktik Der Mathematik Vom 4. Bis 8. März 1996 in Regensburg* (Hildesheim: Franzbecker-Verlag) pp 337–40

- [50] Klieme E and Maichle U 1994 *Modellbildung Und Simulation im Unterricht Der Sekundarstufe I —Auswertung Von Unterrichtsversuchen Mit Dem Modellbildungssystem MODUS* (Bonn: Institut für Bildungsforschung)
- [51] Schecker H and Niedderer H 1991 Über den Beitrag des Computereinsatzes zur Entwicklung physikalischer Kompetenz *Zur Didaktik Der Physik und Chemie, Probleme Und Perspektiven, Vorträge Auf Der Tagung für Didaktik Der Physik/Chemie in Weingarten, September 1990, Gesellschaft für Didaktik der Chemie Und Physik (GDGP)* ed K H Wiebel (Alsbach: Leuchtturm-Verlag) pp 147–9
- [52] Schecker H 1998 Physiklernen mit Modellbildungssystemen—Forschungskonzeptionen ed H Behrendt *Zur Didaktik der Physik und Chemie, Probleme und Perspektiven, Vorträge Auf Der Tagung für Didaktik Der Physik/Chemie in Potsdam, September 1997, Gesellschaft für Didaktik Der Chemie Und Physik (GDGP)* (Alsbach: Leuchtturm-Verlag) pp 230–2
- [53] Schecker H and Gerdes J 1999 Messung der Konzeptualisierungsfähigkeit in der Mechanik—Zur Aussagekraft des Force Concept Inventory In:*Zeitschrift für Didaktik der Naturwissenschaften* **5** 75–89
- [54] Schecker H and Gerdes J 1998 Interviews über Experimente zu Bewegungsvorgängen In: *Zeitschrift für Didaktik der Naturwissenschaften* **4** 61–74
- [55] Sander F, Schecker H and Niedderer H 2001 Wirkungen des Einsatzes grafikorientierter Modellbildung im physikalischen Praktikum In:*Zeitschrift für Didaktik Der Naturwissenschaften* **7** 147–65
- [56] Hücke L and Fischer H E 1999 Die Verbindung von Theorie und Praxis in traditionellen und in computergestützten Praktikumsexperimenten. Fallstudien zur Effektivität des physikalischen Anfängerpraktikums ed R Brechel *Zur Didaktik der Physik und Chemie, Probleme und Perspektiven, Vorträge auf der Tagung für Didaktik der Physik/Chemie in Essen* (Alsbach: GDGP, Leuchtturm-Verlag) pp 250–2
- [57] Jackson J, Dukerich L and Hestenes D 2008 Modeling instruction: an effective model for science education *Science Education* **17** 10–7
- [58] Brewster E, Sawtelle V, Kramer L, O'Brien G, Rodriguez I and Pamelá P 2010 Toward equity through participation in Modeling Instruction in introductory university physics *Phys. Educ. Res.* **6** 010106
- [59] Brewster E, Kramer L and O'Brien G 2009 Modeling instruction: Positive attitudinal shifts in introductory physics measured with CLASS In:*Phys. Educ. Res.* **5** 013102
- [60] Sawtelle V, Brewster E and Kramer L 2010 Positive impacts of modeling instruction on self-efficacy *AIP Conf. Proc.* **1289** 289–92
- [61] Aiken J, Caballero M, Douglas S, Burk J, Scanlon E, Thoms B and Schatz M 2012 Understanding student computational thinking with computational modeling In:*AIP Conf. Proc.* **1513** 46–9
- [62] Caballero M, Kohlmyer M and Schatz M 2012 Implementing and assessing computational modeling in introductory mechanics In:*Phys. Rev. ST Phys. Educ. Res.* **8** 020106
- [63] Caballero M and Pollock S 2014 A model for incorporating computation without changing the course: an example from middle-division classical mechanics In:*Am. J. Phys.* **82** 231–7
- [64] Irving P, Obsniuk M and Caballero M 2017 P3: a practice focused learning environment *Eur. J. Phys.* **38** 055701
- [65] Chabay R and Sherwood B 2002 *Matter & Interactions I. Modern Mechanics* (New York: Wiley)
- [66] Gomes Neves R, Carvalho Silva J and Duarte Teodoro V 2010 Computational modelling with modellus: an enhancement vector for the general university physics course arXiv:1006.4662
- [67] Araujo I, Veit A and Moreira M 2010 Physics students' performance using computational modelling activities to improve kinematics graphs interpretation In:*Comput. Educ.* **50** 1128–40
- [68] Burke C and Atherton T 2017 Developing a project-based computational physics course grounded in expert practice In:*Am. J. Phys.* **85** 301–10
- [69] Hawkins N, Obsniuk M, Irving P and Caballero D 2017 Examining thematic variation in a phenomenographical study on computational physics *Physics Education Research Conf. 2017 (Cincinnati, OH)* pp 168–71
- [70] Benacka J 2015 Projectile general motion in a vacuum and a spreadsheet simulation In:*Phys. Educ.* **50** 58–63
- [71] Benacka J 2015 Spreadsheet application showing the proper elevation angle, points of shot and impact of a projectile In:*Phys. Educ.* **50** 342–7
- [72] Benacka J 2016 Numerical modelling with spreadsheets as a means to promote STEM to high school students In:*Eurasia Journal of Mathematics, Science & Technology Education* **12** 947–64