

Integrating Computational Modelling in Science, Technology, Engineering and Mathematics Education

Rui Gomes Neves

Unidade de Investigação Educação e Desenvolvimento (UIED)

Departamento de Ciências Sociais Aplicadas, Faculdade de Ciências e Tecnologia

Universidade Nova de Lisboa

Monte da Caparica, 2829-516 Caparica, Portugal

Email for correspondence: rgn@fct.unl.pt

Jorge Carvalho Silva

Centro de Física e Investigação Tecnológica (CEFITEC)

Departamento de Física, Faculdade de Ciências e Tecnologia

Universidade Nova de Lisboa

Monte da Caparica, 2829-516 Caparica, Portugal

Vítor Duarte Teodoro

Unidade de Investigação Educação e Desenvolvimento (UIED)

Departamento de Ciências Sociais Aplicadas, Faculdade de Ciências e Tecnologia

Universidade Nova de Lisboa

Monte da Caparica, 2829-516 Caparica, Portugal

Modelling is a central aspect of the research process in science, technology, engineering, and mathematics (STEM), which occurs in the cognitive context of an interactive balance between theory, experiment, and computation. The STEM learning processes should then also involve modelling in environments where there is a balanced interplay between theory, experiment, and computation. However, an adequate integration of computational themes in STEM high school and undergraduate university curricula remains to be achieved. In this chapter, we present an approach to embed computational modelling activities in the STEM learning processes which may be fruitfully adopted by curricula at secondary and introductory university levels, as well as be a valuable instrument for the professional development of teachers. To illustrate, we consider the example of physics.

Introduction

Science, technology, engineering and mathematics (STEM) are evolving structures of knowledge which are symbiotically interconnected. On one hand, science is based on hypotheses and models, leading to theories, which have a strong mathematical character as scientific reasoning, concepts and laws are represented by mathematical reason-

ing, entities and relations. On the other hand, scientific explanations and predictions must be consistent with the results of systematic and reliable experiments, which depend on technological developments as much as these depend on the progress of science and mathematics (see, e.g., Chalmers, 1999; Crump, 2001; Feynman, 1967). The creation of STEM knowledge is a dynamical cognition process which involves a blend of individual and collective actions where modelling occurs with a balance between theoretical, experimental and computational elements (Blum, Galbraith, Henn & Niss, 2007; Neunzert & Siddiqi, 2000; Schwartz, 2007; Slooten, van den Berg & Ellermeijer, 2006). In this process, computational modelling plays a key role in the expansion of the STEM cognitive horizon through enhanced calculation, exploration and visualization capabilities.

Although clearly related to real world phenomena, STEM knowledge is thus built upon abstract and subtle conceptual and methodological frameworks which in addition evolve following historical dependent paths. These epistemological and cognitive features make science, technology, engineering and mathematics difficult fields to learn and to teach. An approach to STEM education that aims to be effective and in phase with the rapid scientific and technological development should then be based on pedagogical methodologies inspired in the modelling processes of STEM professional activities. Meaningful learning (see, e.g., Mintzes, Wandersee & Novak, 2005) should then occur when students go through balanced interactive explorations of the different cognitive phases associated with the modelling cycles of STEM research, starting from a qualitative contextualization phase, setting the stage for the definition, exploration, interpretation and validation of the relevant mathematical models, and ending with the communication of modelling results and the development of generalizations. In particular, it is of crucial importance to promote an early integration of computational modelling in learning environments reflecting the exploratory and interactive nature of modern research (Ogborn, 1994).

However, even in technologically advanced countries, computers, computational methods and software, as well as exploratory and interactive learning environments, are still not appropriately integrated in the majority of STEM education curricula, a problem cutting across all educational levels from basic primary school to higher university degrees. As a consequence, most of these curricula are generally outdated and tend to transmit to students a sense of detachment from how STEM professional activities are developed today. These are factors that contribute to the appearance of negative views about the education process and to an increase in student failure.

Physics education is a good example to illustrate this problem. Consider the general physics courses taken by first year university students. These are courses which usually follow a traditional lecture plus laboratory instruction approach and cover a large number of physics topics which students find particularly difficult. Due to a lack of understanding of the necessary fundamental concepts and methods in physics and math-

ematics, the number of students that fail on the course examinations is usually very high. What is worse is that many of those students that do actually succeed also reveal several weaknesses in their understanding of elementary physics (Halloun & Hestenes, 1985a; Halloun & Hestenes, 1985b; Hestenes, 1987; McDermott, 1991).

Research in physics education has shown that this situation can be improved when students are involved in the learning activities as scientists are involved in research (Beichner et al., 1999; Mazur, 1997; McDermott, 1997; McDermott & Redish, 1999). This is not a surprising result. Scientific research in physics is an interactive and exploratory process of creation, testing and improvement of mathematical models that describe observable physical phenomena. It is this cognitive process that leads to an inspiring understanding of the rules of the physical universe. As a consequence, physics should be expected to be more successfully taught in interactive and exploratory environments where students are helped by teachers to work as scientists do. In this kind of class environment knowledge performance is better promoted and common sense beliefs as well as incorrect scientific ideas can be more effectively fought.

The scientific research process is supported by a continuously evolving set of analytical, computational and experimental techniques. The same should be true for research based learning environments. Consequently, another important aspect of these learning environments is the possibility to balance the role of computational modelling methods and tools. This would set the learning process in phase not only with modern scientific research where computation is as important as theory and experiment, but also with the rapid parallel development of technology.

Several attempts have already been made to introduce computational modelling in research based learning environments. The starting emphasis was on professional programming languages such as Fortran (Bork, 1967) and Pascal (Redish & Wilson, 1993). Although more recently this approach has evolved to Python (Chabay & Sherwood, 2008), it continues to require that students develop a working knowledge of programming, a time consuming task which can hinder the process of learning physics. The same happens with professional scientific computation software such as Mathematica or Matlab. To avoid overloading students with the detail of programming notions and syntax, and focus the learning process on the relevant physics and mathematics, several computer modelling systems were created, for example, the Dynamical Modelling System (Ogborn, 1985), Stella (Richmond, 2004), Coach (Heck, Kadzierska & Ellermeijer, 2009), EJS (Christian & Esquembre, 2007) and Modellus (Neves, Silva & Teodoro, 2011; Neves & Teodoro, 2010; Teodoro & Neves, 2011).

An adequate integration of computational modelling methods and tools in STEM learning environments is, thus, both a curricular and a technological development problem. In this work, we discuss how Modellus (a freely available software tool created in Java which is able to run in all operating systems, see the software webpage at <http://modellus.fct.unl.pt>) can be used as a central element in an approach

to develop exploratory and interactive computational modelling learning activities relevant for STEM education. These activities can be adopted by curricula at the secondary and introductory university levels, and used as a valuable instrument for the corresponding professional development of teachers. To illustrate, we consider computational modelling with Modellus to teach physics, namely introductory mechanics, and discuss its impact on the student learning processes.

Modellus: interactive and exploratory computational modelling for STEM education

The construction of STEM knowledge requires unambiguously clear declarative, operational and conditional specifications of abstract concepts and of the relations among such concepts. Of crucial importance in the understanding of the resulting models or theories is the interpretation process which involves operational familiarization and connection with the relevant referents in the observable universe (Reif, 2008). In education, as in research, computers, computational methods and software are cognitive artefacts (Norman, 1991; Teodoro, Schwartz & Neves, 2012) which may amplify the learning cognition horizon due to more powerful calculation, exploration and visualization capabilities. Indeed, with an appropriately equipped computer it can become easier for mental actions to treat abstract conceptual entities as real objects (Papert, 1980). Moreover, on screen real time interaction gives students the opportunity to use the computer as powerful intellectual mirror for their own cognitive activity (Schwartz, 1989). As a consequence, these artefacts can play a key role in enhancing the operational familiarization and the connection with the real world referents which must necessarily be involved in the STEM learning processes. To be able to fulfil such a potential key role, computational methods and tools should be used not only to display text, images or simulations but as mathematical modelling tools integrated in learning environments which reflect the exploratory and interactive nature of modern research. In addition, the modelling process should be focused on the meaning of models and avoid cognitive overload due to too much programming and specific software knowledge.

To meet this educational challenge it is not enough to simply choose a subset of programming languages and professional computational software. It is necessary to develop computer software systems with computational modelling functionalities which contribute to nurture the progressive growth of solid STEM cognitive competencies. Among the systems already created, Modellus stands out as a computational modelling tool for STEM education because of the following main advantages: 1) an easy and intuitive creation of mathematical models using standard mathematical notation; 2) the possibility to create animations with interactive objects that have mathematical properties expressed in the model; 3) the simultaneous exploration of multiple representations such as images, tables, graphs and animations; 4) the computation and display of math-

ematical quantities obtained from the analysis of images and graphs.

These are features that allow a deeper cognitive contact of models with the relevant real world referents and a deeper operational exploration of models as objects which are simultaneously abstract, in the sense that they represent relations between mathematical entities, and concrete, in the sense that they may be directly manipulated in the computer. In a word, Modellus allows a deeper reification (Teodoro, Schwartz & Neves, 2012) of abstract mathematical objects. Because of these characteristics, computational modelling activities built with Modellus can be readily conceived as exploratory and interactive modelling experiments performed by students in collaborative groups or individually. They can also be designed with an emphasis on cognitive conflicts in the understanding of STEM concepts, on the manipulation of multiple representations of mathematical models and on the interplay between the analytical and numerical approaches applied to solve STEM problems.

As a domain general environment for modelling, Modellus can be used to develop STEM learning activities which range from the exploration of existing models to the development of new ones (Bliss & Ogborn, 1989; Schwartz, 2007). As much as possible, such modelling activities should consider realistic problems to maximise the cognitive contact with the real world referents. This is a challenge because more realistic problems are generally associated with more complex analytic solutions which are beyond the analytic capabilities of students at the secondary or first to second year university levels. With Modellus and numerical methods, which are conceptually simpler and yet powerful, the interactive exploration of models for more realistic problems can start at an earlier age, allowing students a closer contact with the model referents, an essential cognitive element to appreciate the relevancy and power of models, necessarily a partial idealized representation of their referents.

Clearly, the development of the appropriate computational modelling activities for STEM research inspired learning environments is bound to call for a richer set of modelling functionalities which are not yet available in Modellus. These events are seeds for technological evolution which should be accomplished by a Modellus enhancement program. Currently under development and set to appear in forthcoming versions of Modellus is, for example, the following set of new functionalities: spread sheet, data logging and curve fitting capabilities, advanced animation objects like curves, waves and fields, 3D animations and graphs, creation of a physics engine for motion and collisions, video analysis and cellular automata models.

The simultaneous development of new functionalities to meet appropriate teaching goals is important because it reduces the cognitive load associated to an unnecessary proliferation of tools. However, there is a learning stage where it is advantageous to allow some diversity in the use of computer software tools and complement Modellus with other available tools. Indeed, in a research based STEM learning environment one of the objectives is to make a progressive introduction to professional STEM com-

putation methods and software. For example, Excel is a general purpose spread sheet where modelling is focused on the algorithms. In addition, it already allows data analysis from direct data logging. On the other hand, Mathematica and Matlab have powerful symbolic computation capabilities. Using these different tools to implement the same algorithm is an important step to learn the meaning of the algorithm instead of the syntax of a particular tool. If more realistic simulations are needed, Modellus animations can be complemented, for example, with EJS.

Computational modelling learning activities: an illustrative example from rotational dynamics

Let us consider a computational modelling activity about rigid body rotational dynamics and angular momentum, a general physics topic which in a course program for first year university students should be introduced after computational modelling activities covering vectors, kinematics and Newton's fundamental laws of motion, including simple numerical and analytical solutions (Neves, Silva & Teodoro, 2009, 2011; Neves & Teodoro, 2010).

A rigid body is a system of particles whose relative distance does not change with time. When a rigid body rotates around a fixed axis, each one of its particles has a circular motion around the axis which is characterised by a rotation angle θ , an angular velocity ω and an angular acceleration α . The kinetic rotation energy is the sum of the kinetic energies of all the particles of the body and is given by $K = I\omega^2/2$, where I is the moment of inertia of the body relative to the rotation axis. From a dynamical point of view, the rotational motion of a rigid body around a fixed axis is characterised by two vectors, the angular momentum of the rigid body and the moment of the sum of all the forces acting on the body, the latter also called the net applied torque. Newton's laws of motion imply that the instantaneous rate of change of the angular momentum is equal to the net applied torque. If the rotation axis coincides with the Oz axis, the angular momentum is given by $\vec{L} = L\vec{e}_z = I\omega\vec{e}_z$, where \vec{e}_z is the unit vector of the Oz axis. The corresponding moment of the sum of all applied forces is $\vec{\tau} = \tau\vec{e}_z = I\alpha\vec{e}_z$. A real world system which may be considered as a rotating rigid body is a wind turbine. With Modellus it is possible to model the action of the wind on the rotor blades and analyse their motion using at the same time different representations such as graphs, tables and object animations. A mathematical model example in SI units is given in Figure 1. In this model the fundamental equations of the rotational motion are written in the form of Euler iterations. Students are thus taught to apply this numerical method in a new realistic context, extending the applicability range of knowledge already acquired with a previous analysis of analogous numerical solutions of Newton's equations in translational motion settings (Neves, Silva & Teodoro, 2009; Neves & Teodoro, 2010). In this new application, students can determine the angular velocity and the rotation angle knowing the net applied torque, the moment of inertia of the system and the

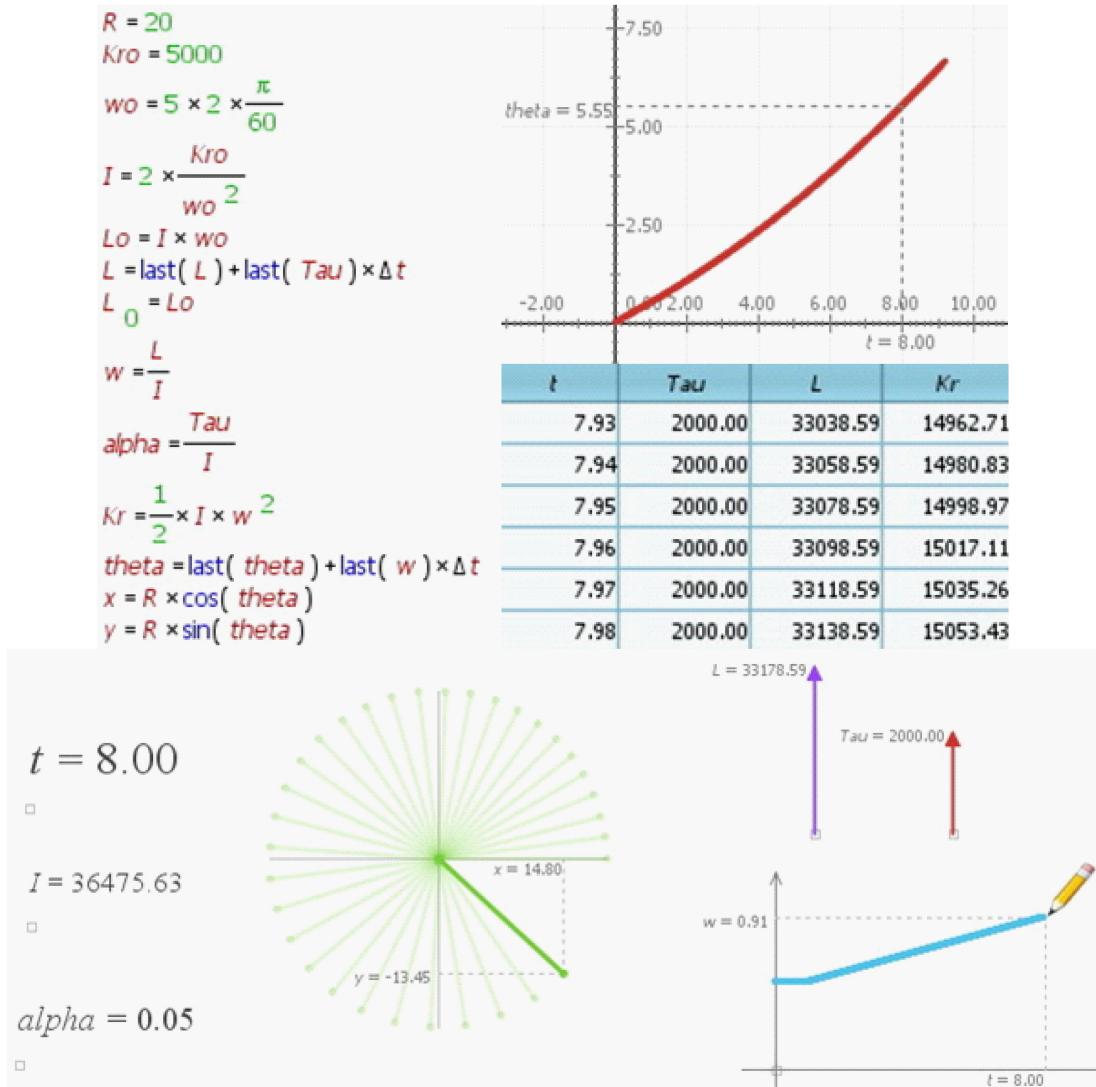


Figure 1: The rotational motion of a wind turbine: equations as seen in the Modellus Mathematical Model window, an example of a coordinate-time graph of the Graph window, selected variable values of the Table window and the Modellus animation where the bar stroboscopic effect shows the acceleration due to the net applied torque.

motion initial conditions.

The model animation is constructed with three objects: a bar representing the rotor blade, a vector representing the angular momentum and a vector representing the net wind torque (see Figure 1). Because the coordinates of the net torque are independent variables and the model is iterative, students can manipulate this vector at will and in real time control the motion of the rotor blade. With this activity students can confirm that the choice of the time step is an important one to obtain a good simulation of the motion and that this is the same as determining a good numerical solution of the equations of motion. Simultaneously, students can construct and visualize graphs and tables of model variables (see Figure 1). While exploring the model, students can determine, for example, what are the values of the angular momentum, the angular velocity and

the rotational kinetic energy, 8 seconds after increasing the net wind torque to 2000 Nm (see Figure 1). The possibility to change the mathematical model and immediately observe this action on the animation, graphs and tables is a powerful cognitive element to enhance the students learning process. Students can also change the model. Introducing a vector to represent the wind force they can explore the effect of the wind direction on the net applied torque and on the motion of the rotor blades.

Field actions, discussion and outlook

In this paper we have shown how Modellus can be used as a key element of an approach to develop exploratory and interactive computational modelling learning activities for research based STEM education. As an example, we have discussed the modelling of the rotational dynamics of a wind turbine.

Since 2008 this and other computational modelling activities created in the Modellus environment have been tested on the field during the implementation of our approach in the general physics and biophysics courses offered to the biomedical engineering and informatics engineering students at FCT/UNL (Neves, Silva & Teodoro, 2009, 2010, 2011; Neves & Teodoro, 2010, Teodoro & Neves, 2011). Other Modellus based activities with mathematical physics models relevant to introduce meteorology have also been developed and implemented in courses gathering students from several undergraduate university degrees, namely, environmental engineering, marine sciences and biology (Neves, Neves & Teodoro, 2011). For other educational applications and evaluation tests of Modellus based computational modelling activities see, e.g., Araújo, Veit & Moreira (2008), Dorneles, Araújo & Veit (2008) and Teodoro (2002).

In the general physics and biophysics courses, interactive collaborative learning environments were built organising students in teams of two or three, one team for each classroom computer. For example, the 2010 biophysics course for biomedical engineering involved 28 groups, 2 teams of 3 and 26 teams of 2, and the 2011 general physics course for informatics engineering involved 56 groups, 53 teams of 2 and 3 teams of 3. During each class, the student teams worked on a set of Modellus based computational modelling activities built around a small number of physics problems connected with easily observed real world phenomena. The teams were instructed to analyse and discuss the problems on their own using the physical, mathematical and computational modelling guidelines provided by the course documentation. To ensure a good working pace with appropriate conceptual, analytical and computational understanding, the students were continuously monitored and helped during the exploration of the activities. Whenever it was felt necessary, global class discussions were conducted to keep the pace, to introduce new themes, or to clarify doubts on concepts, reasoning or calculations. Online support in class and at home was provided in the context of the Moodle platform where links to class, homework and assessment documentation were provided.

All computational modelling activities were supported by interactive PDF documents which explained the fundamental modelling ideas, problem solving processes and challenges to solve using text, images and embedded movies. For each course a specific thematic sequence of PDF documents was created where each individual PDF document contained 4, 5 or 6 computational modelling activities. To increase the degree of digital interactivity, free space was included to allow students the opportunity of inserting multimedia based comments related with their modelling actions, such as text answers or comments to the proposed modelling activities, and associated schemes, diagrams, images, or movies.

The evaluation procedures associated with the computational modelling course components involved group and individual evaluation. For each PDF document, all student groups had to complete an online test written in the Moodle platform answering a set of questions contained in the PDF document. For selected modelling activities the groups also had to present PDF documents with solutions and their corresponding multimedia justifying comments. The individual evaluation elements were more dependent on the specific course. For example, in the 2010 biophysics course for biomedical engineering all students had to do a final class test with computational modelling problems to be solved with Modellus. In the 2011 general physics course for informatics engineering all students had to do a paper and pencil diagnostic test at the beginning of the course and a similar final test at the end. At the end of the courses, students also answered a questionnaire to evaluate Modellus and the computational modelling activities.

In these introductory physics courses, the computational modelling activities were successful in identifying and resolving many of the student difficulties in key physical and mathematical concepts. For example, in the 2010 biophysics course the average grade was 70% in the standard 0 to 100% scale and all students were able to pass on the computational modelling component. In the 2011 general physics course the average grade was 78% and only 4 students failed to pass. Having real-time on-screen correspondence between the animations with interactive objects and the object's mathematical properties defined in the model, and the possibility of simultaneously manipulating several different representations were central to achieve these results. In the Modellus modelling environment, students also had the opportunity to create and explore models and animations, and not just act as simple browsers of computer simulations. In addition, students were able to solve Newtonian models based on first order ordinary differential equations applying simple numerical methods, such as the Euler and Euler-Cromer methods, and understanding the conceptual and operational differences existing between numerical solutions and analytical solutions. Students were also able to explore advanced mathematical concepts such as integration in the context of real world physics problems, prior to the introduction of the corresponding analytic techniques.

As shown by the average results of Likert scale questionnaires, and consistently since 2008 (Neves, Silva & Teodoro, 2009, 2010, 2011; Neves & Teodoro, 2010; Teodoro

& Neves, 2011), the majority of students manifested positive opinions about the computational modelling activities in the Modellus environment. In these questionnaires, students gave their opinion about a set of statements about the Modellus based computational modelling learning activities (Neves, Silva & Teodoro, 2011). A Likert scale starting at -3 and ending at +3 was used, where -3 stated complete disagreement and +3 complete agreement. To choose zero meant not having a preferred opinion. Defining the average opinion of a student as the average over all answers given by the student to the questionnaire statements, the results obtained for the 2010 biophysics course showed that 69% of the students had a positive opinion, averaging either 1 or 2, 14% had a negative opinion, averaging either -1 or -2, while the rest averaged no preferred opinion. For the 2011 general physics course 48% had a positive opinion, 20% a negative opinion and 31% averaged no preferred opinion. As shown by the average over all student answers for the corresponding questionnaire statements, students showed clear preference for the collaborative work in groups and considered Modellus helpful and user friendly in the processes of learning mathematical physics models. The digital PDF documents with guiding interactive text, images and movies as well as free space to insert multimedia answers or comments were also considered to be useful. However, the questionnaire results also indicate that students felt cautious about the new way of learning physics with Modellus and computational methods. For example, as perceived by students, the introduction of computational knowledge and technologies in the process of learning physics persists in meaning a heavier content load to master for which the available time is not sufficient. More details about these and other research actions and their effective learning outcomes will be present in forthcoming publications.

Acknowledgements

We would like to thank Judah Schwartz for valuable discussions and comments. This work was supported by UIED, FCT/UNL and Fundação para a Ciência e a Tecnologia (FCT), Programa Compromisso com a Ciência, Ciência 2007.

References

- Araújo, I., Veit, E., & Moreira, M. (2008). Physics student's performance using computational modelling activities to improve kinematics graphs interpretation. *Computers and Education*, 50, 1128-1140.
- Beichner, R., Bernold, L., Burniston, E., Dail, P., Felder, R., Gastineau, J., Gjertsen, M., & Risley, J. (1999). Case study of the physics component of an integrated curriculum. *Physics Education Research, American Journal of Physics Supplement*, 67, 16-24.
- Bliss, J., & Ogborn, J. (1989). Tools for exploratory learning. *Journal of Computer Assisted learning*, 5(1), 37-50.
- Blum, W., Galbraith, P., Henn, H.-W., & Niss, M. (Eds.) (2007). *Modelling and applications in mathematics education*. New York, USA: Springer.
- Bork, A. (1967). *Fortran for physics*. Reading, Massachusetts, USA: Addison-Wesley.
- Chabay, R., & Sherwood, B. (2008). Computational physics in the introductory calculus-based course. *American Journal of Physics*, 76, 307-313.
- Chalmers, A. (1999). *What is this thing called science?*. London, UK: Open University Press.
- Christian, W., & Esquembre, F. (2007). Modeling physics with Easy Java Simulations. *The Physics Teacher*, 45(8), 475-480.
- Crump, T. (2002). *A brief history of science, as seen through the development of scientific instruments*. London, UK: Robinson.
- Dorneles, F., Araújo, I., & Veit, E. (2008). Computational modelling and simulation activities to help a meaningful learning of electricity basic concepts. Part II - RLC circuits. *Revista Brasileira de Ensino da Física*, v. 30, n. 3, 3308, 1-16.
- Feynman, R. (1967). *The character of physical law*. New York, USA: MIT Press.
- Halloun, I., & Hestenes, D. (1985a). The initial knowledge state of college physics students. *American Journal of Physics*, 53, 1043-1048.
- Halloun, I., & Hestenes, D. (1985b). Common-sense concepts about motion. *American Journal of Physics*, 53, 1056-1065.
- Heck, A., Kadzierska, E., & Ellermeijer, T. (2009). Design and implementation of an integrated computer working environment. *Journal of Computers in Mathematics and Science Teaching*, 28, 147-161.
- Hestenes, D. (1987). Toward a modelling theory of physics instruction. *American Journal of Physics*, 55, 440-454.
- Mazur, E. (1997). *Peer instruction: A user's manual*. New Jersey, USA: Prentice-Hall.
- McDermott, L. C. (1991). Millikan Lecture 1990: What we teach and what is learned-closing the gap. *American Journal of Physics*, 59, 301-315.

- McDermott, L. C. (1997). *Physics by inquiry*. New York, USA: Wiley.
- McDermott, L., & Redish, E. (1999). Resource letter: Per-1: Physics education research. *American Journal of Physics*, 67, 755-767.
- Mintzes, J., Wandersee, J., & Novak, J. (Eds.) (2005). *Teaching science for understanding: A human constructivist view*. Burlington, MA: Elsevier Academic Press.
- Neunzert, H., & Siddiqi, A. (2000). *Topics in industrial mathematics: Case studies and related mathematical models*. Dordrecht, Holland: Kluwer Academic Publishers.
- Neves, R., Neves, M. C., & Teodoro, V. (2011). Improving learning processes in meteorology with computational modelling. *Geophysical Research Abstracts Vol. 13*, EGU2011-6122.
- Neves, R., Silva, J., & Teodoro, V. (2009). Computational modelling with Modellus: An enhancement vector for the general university physics course. In A. Bilsel & M. Garip (Eds.), *Frontiers in science education research* (pp. 461-470). Famagusta, Cyprus: Eastern Mediterranean University Press.
- Neves, R., Silva, J., & Teodoro, V. (2010). Computational modelling in science, technology, engineering and mathematics education. In A. Araújo, A. Fernandes, A. Azevedo & J. Rodrigues (Eds.), *Proceedings of the EIMI 2010 Conference: Educational interfaces between Mathematics and Industry* (pp. 387-397). Bedford, MA: Centro Internacional de Matemática & Comap Inc.
- Neves, R., Silva, J., & Teodoro, V. (2011). Improving learning in science and mathematics with exploratory and interactive computational modelling. In G. Kaiser, W. Blum, R. Borromeo-Ferri, & G. Stillman (Eds.), *International perspectives on the teaching and learning of mathematical modelling: ICTMA14 - Trends in teaching and learning of mathematical modelling* (Vol. 1, pp. 331-341). Dordrecht: Springer.
- Neves, R., & Teodoro, V. (2010). Enhancing science and mathematics education with computational modelling. *Journal of Mathematical Modelling and Application*, 1, 2-15.
- Norman, D. (1991). Cognitive artefacts. In J. Carroll (Ed.), *Designing interaction: Psychology at the human-computer interface* (pp. 17-38). Cambridge: Cambridge University Press.
- Ogborn, J. (1994). Modelling clay for computers. In B. Jennison & J. Ogborn (Eds.), *Wonder and delight: Essays in science education in honour of the life and work of Eric Rogers 1902-1990* (pp. 103-114). Bristol: Institute of Physics Publishing.
- Ogborn, J. (1985). *Dynamic Modelling System*. Harlow, Essex: Longman.
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York: Basic Books.
- Redish, E., & Wilson, J. (1993). Student programming in the introductory physics course: M.U.P.P.E.T. *American Journal of Physics*, 61, 222-232.
- Reif, F. (2008). *Applying cognitive science to education: thinking and learning in scientific*

- and other complex domains.* Cambridge, USA: MIT Press.
- Richmond, B. (2004). *An introduction to systems thinking with Stella*. Lebanon, NH: ISEE Systems, Inc.
- Schwartz, J. (1989). Intellectual mirrors: A step in the direction of making schools knowledge-making places. *Harvard Educational Review*, 59, 217- 222.
- Schwartz, J. (2007). Models, simulations, and exploratory environments: A tentative taxonomy. In R. A. Lesh, E. Hamilton & J. J. Kaput (Eds.), *Foundations for the future in mathematics education* (pp. 161-172). Mahwah, NJ: Lawrence Erlbaum Associates.
- Slooten, O., van den Berg, E. & Ellermeijer, T. (Eds.) (2006). *Proceedings of the International Group of Research on Physics Education 2006 Conference: Modeling in physics and physics education*. Amsterdam: European Physical Society.
- Teodoro, V. D. (2002). *Modellus: Learning physics with mathematical modelling*. Unpublished PhD Thesis. Lisboa, Portugal: Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.
- Teodoro, V., & Neves, R. (2011). Mathematical modelling in science and mathematics education. *Computer Physics Communications*, 182, 8-10.
- Teodoro, V., Schwartz, J., & Neves, R. (2012). Cognitive artifacts, technology, and physics learning. In N. M. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 572-576). Dordrecht: Springer.