First year mechanics taught through modelling in VPython

David Sands
Department of Physical Sciences
University of Hull
Hull
HU6 7RX
d.sands@hull.ac.uk

Abstract

This paper describes a development project carried out in 2008/9 aimed at developing model-based learning in mechanics for a first year physics module. Based on the work in the literature, VPython, the visual extension to the Python programming language, was chosen as the vehicle for developing the models. VPython is ideally suited to modelling mechanics for various reasons, including a class of variables called vectors which have all the properties of vectors in mathematics, the ease with which basic models in VPython can be constructed, and the instant feedback on the operation of the models afforded by their visual nature. Thus the emphasis is much more on the physics and the modelling rather than computation. It is shown how an analysis of students' understanding has revealed that Newton's third law of motion causes difficulties, leading to a greater emphasis on this concept in the modelling for 2009/10. In addition, a greater attention was given to the methods and techniques of modelling, especially spatial reasoning. The evidence for student reasoning in this way is presented.

Introduction

This paper describes the implementation and subsequent development of a first year course in mechanics based around modelling in VPython, the visual extension of the Python programming language. Developed with the help of development project funding from the UK Physical Sciences Centre, the aim was to design an alternative form of instruction which helped overcome the deficiencies in entry-level maths knowledge. It is widely recognised, for example, that, compared with a generation ago, the entry level knowledge of physics undergraduates has declined markedly due to changes in the teaching of both physics and maths in schools. Many students now lack in-depth knowledge of either differential calculus or Newtonian mechanics. In consequence, lectures on topics which draw heavily on mathematical principles, such as electricity and magnetism or mechanics, are hard to follow and hard to grasp. Moreover, even if mathematical knowledge and skills are taught alongside the physics, students are unlikely to be fluent and may find it difficult to transfer this knowledge from the mathematical domain to the physical context. Therefore the aim of this approach is to concentrate on physics concepts themselves through the construction of computer models.

Many students now lack in-depth knowledge of either differential calculus or Newtonian mechanics. In consequence, lectures on topics which draw heavily on mathematical principles, such as electricity and magnetism or mechanics, are hard to follow and hard to grasp.

The methodology is based on the work of both David Hestenes[1], and Ruth Chabay and Bruce Sherwood[2]. Hestenes is a strong advocate of the power of modelling to improve student learning, having previously identified the naive views that physics students often hold about mechanics, are tested by the FCI, or Force Concept Inventory[3]. Chabay and Sherwood have long advocated a computational physics approach based on VPython. Thus Hestenes' work provides the nature and structure of a model and the importance of qualitative reasoning and Chabay and Sherwood's pioneering efforts in computational instruction provide the basis for quantitative modelling. This flow of information from the qualitative to the quantitative appears to be quite general and an important aspect of computational modelling[4].

There is another reason, however, for wanting to combine these two different approaches to modelling: practicality. This course replaced a conventional 20-lecture, mathematically based course on classical, mechanics it was expected to feed into later modules that the students take. It is not possible, therefore, to tear up the curriculum and start afresh, which precludes adopting Hestenes' approach in its entirety. Hestenes[5] advocates the construction, through group discussion, of just a few models related to motion, but in the present work the physics content is largely fixed. It is desirable, however, to move away from conventional lectures. Hestenes has shown that FCI scores correlate with functional understanding of complicated Newtonian concepts and testing of thousands of students post-learning reveals that traditional, lecture-based learning

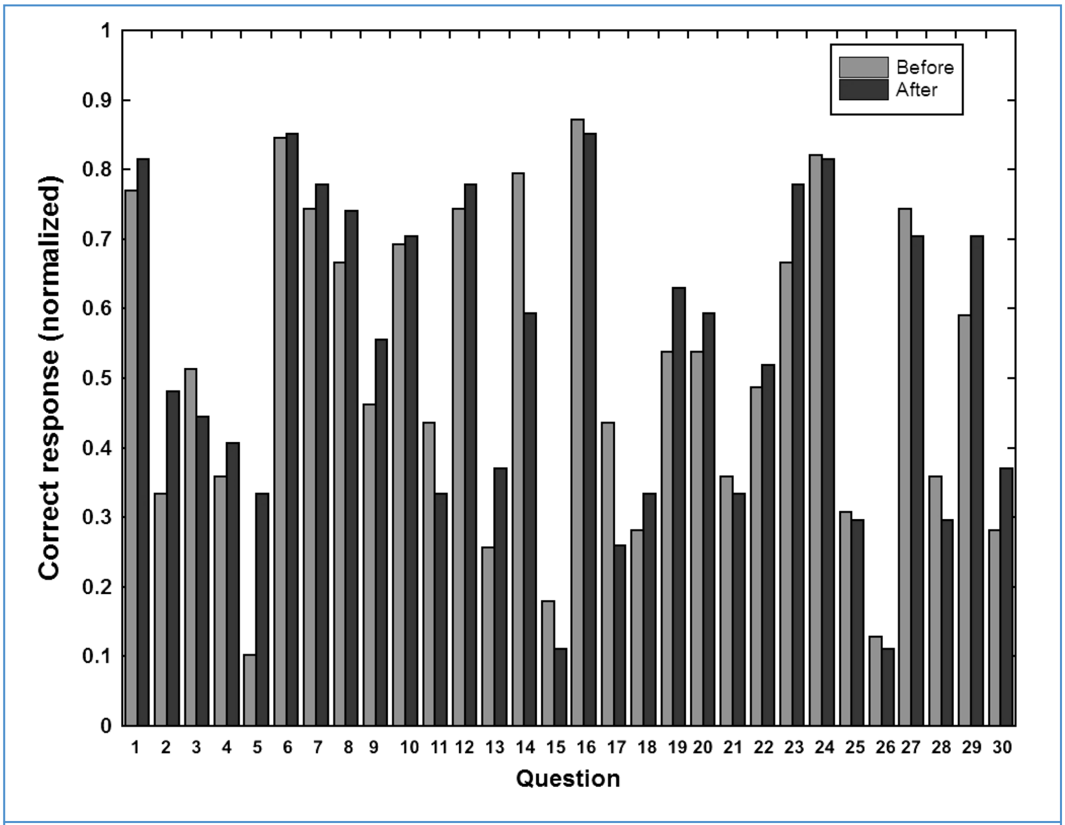


Figure 1: The percentages of correct answers broken down by question in both pre- and post-instruction FCI tests.

yields the least return[6]. Students taught using model-based learning, on the other hand, can use concepts to solve physics problems. This, then, defines the approach taken in this course material is delivered through a combination of lecture, demonstration and modelling exercises in which students reason about a problem and construct models in VPython. It will be shown in the following that whilst course content is important just as much attention has to be given to the modelling methodology and the methods and techniques of problem solving.

The Course

HEA UK Physical Sciences Centre funding was obtained to fund a student to write programmes in VPython over the summer of 2008. These were intended to serve as the basis for a series of visual demonstrations as well as providing a point of reference for students building their own models. The choice of programming language always presents a difficulty. It has been known for some time[7] that huge variations exist in aptitude and attitude towards computing. Conventional languages such as Pascal have been used in physics education[8] but extensive lines of set-up code can be required. Python, on the other hand, is rapidly becoming accepted as an easy-to-use and easy-to-read language for scientific computing[9] and VPython, the visual extension, is ideally suited to the construction of dynamical, vector-based models. Indeed, an essential element of VPython is a class of variables called vectors which have all the properties of vectors in mathematics, including *dot* and *cross* products. In addition, the visual representation, which is rotatable in all dimensions to give an accurate 3-D depiction, provides instant feedback by allowing the student to see instantly whether the physics is sensible and corresponds to experience.

The course was first delivered in 2008 to 51 students in their first year of a physics bachelor's degree over twenty conventional 50-minute lecture slots. The syllabus, exam structure and total contact time were all constrained by the existing programme specification. The delivery consisted of a mixture of demonstrations using VPython models, walk-through exercises, modelling sessions, and PowerPoint presentations to deliver some of the more formal ideas. These included motion under constant acceleration, such as ballistic trajectories, position dependent forces, such as electrostatic or gravitational attraction, and simple harmonic motion (SHM), including damping and forced harmonic motion. All four techniques were fully integrated in the delivery of a class. By way of assessment, students had to construct two models as well as sit a conventional examination in which the questions tested their knowledge of mechanics rather than Python programming. In addition, students had to construct a number of models in class and two models by way of assignment. These last two comprised exercises on ballistic motion and electrostatic forces.

The course was evaluated using the FCI both before and after instruction. A slight improvement in the post-instruction scores is evident, but the improvement is not as large as might have been hoped for. Further analysis revealed a number of possible factors, two of which stand out. First, the FCI measures understanding of Newtonian concepts and does not address learning in vectors or in SHM, both of which were important parts of the course. Second, as shown in Figure 1, the breakdown of FCI scores by question indicates that a significant prior knowledge of some Newtonian concepts already exists and in such a mixed class it is possible that the learning needs of students at both ends of the ability spectrum are not being addressed. It was decided therefore to make changes to the computational aspects of the course work for 2009 to emphasise those areas of mechanics that are clearly lacking. As indicated in Figure 1 by questions 5, 15 and 26, this involves identifying forces using Newton's third law. Two computational exercises were designed around reaction forces and a third on mutual electrostatic repulsion. The use and manipulation of vector quantities integral to these problems so in addition to the pre- and post course testing with the FCI, elements of the Vector Evaluation Test, or VET[10], were also employed.

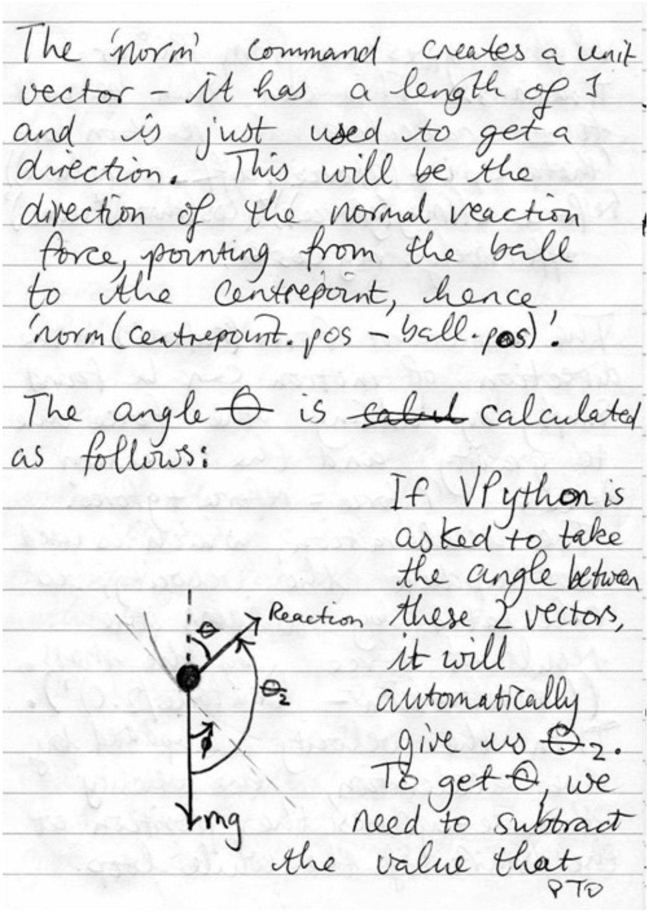
In addition to the above, the problem solving aspects of modelling have been emphasised to a much greater extent. Problem solving is an integral part of modelling, especially computational, as it is necessary first to understand a problem and its solution before constructing a computer model. However, students are neither mature problem solvers nor expert programmers and in response to some of the perceived difficulties support for both of these aspects has been enhanced in 2009/10. First, the formal time spent at the computer was increased to twenty hours and, second, the students were required to keep a diary of their modelling activities. As part of the formal instruction students are shown, by means of a walk-through exercise, how the use of diagrams can aid reasoning and lead to a solution of a problem prior to the generation of a computer model. In a paper co-authored with Tina Overton (this journal) the author has reviewed the literature on problem solving and shown the importance of spatial reasoning, especially the use of representations both to understand the problem and to reason through to a solution. The diary is therefore expected to contain problem representations, qualitative and quantitative reasoning, and information on encoding the problem in VPython.

The Outcome

The course has now run for two years and the outcomes were mainly positive in the first year with further improvements in the second. There was a small, but significant gain in FCI scores and many students like programming in VPython and commented that the course was enjoyable, though some struggled with VPython in the first year. As described above, a number of changes designed to address specific issues were made, including the design of modelling exercises that both emphasise Newton's third law and take advantage of one of the features of VPython, extended computational support through additional class hours, and a greater emphasis on the problem solving aspects of modelling through the use of a diary. In addition, the VET was used for the 2009 class to augment the pre- and post-course testing with the FCI, but at the time of writing a complete analysis of this testing is not available. However, there are indications within the modelling diaries that students have made significant gains in their understanding of vectors. By way of illustration, Figure 2 shows an extract from one of the diaries in which the problem of a mass sliding down a curved slide is addressed. Note that mention is made of a unit vector defined by differences in position (expressed in Python code), and that the diagram is clearly being used to aid the spatial reasoning about angles.

One of the interesting features of this kind of activity-based learning is that it affords an opportunity to observe a class at work in ways that other kinds of instruction do not. There are times during a computational class when students are working away without the need for help and on these occasions it is possible to watch the students at work either on their own or with another, or perhaps even explaining something to another student. A common feature observed in all these classes is the extensive use of hand gestures to depict motion, directions, or spatial relationships. The use of similar gestures using simulations of relative motion has been cited as strong evidence for spatial reasoning[11, 12]. Taken with the diaries and the fact that all but a handful of students out of a class of 60 produced at least one working programme, there is evidence of not only learning in mechanics but also the development of wider problem solving skills. This would not typically be an outcome of a conventional lecture-based course.

It seems reasonable to attribute this success to the requirement to keep a modelling diary. Chabay and Sherwood report that after ten years of activity in introductory computational physics instruction they have not achieved the full educational potential of modelling. A close examination of the work of Kohlmyer[13], a PhD student working with Chabay and Sherwood, reveals a possible reason. Although it was intended that the physics should be emphasised, in fact the emphasis was placed firmly on the computation through the use of problems that were too difficult initially and which had to be altered for subsequent students. In addition, reference to the literature on problem solving tended to concentrate on knowledge structures, by which is meant that a problem based around energy is intended to cue the use of potential energy, kinetic energy or work. It is fair to say that there is little or no reference to qualitative or spatial reasoning or the use of representations. By contrast, this feature of problem solving has been emphasised in the present work and the problems are at a level commensurate with the students' abilities and knowledge. That is not to say that some students did not encounter difficulties with programming. As described by Bishop-Clark[7], enormous personal differences in performance on computer programming tasks exist, and it is clearly an ongoing research issue to try to identify the reasons why some students struggle, but the emphasis on qualitative and spatial reasoning and the production of a diary means that students can still benefit from solving the problem even if a working computer model is not always an outcome.



The 'norm' command creates a unit vector – it has a length of 1 and is just used to get a direction. This will be the direction of the normal reaction force, pointing from the ball to the centrepoint, hence 'norm(centrepoint.pos – ball.pos)'.

The angle θ is ~~calcul~~ calculated as follows:

If VPython is asked to take the angle between these 2 vectors, it will automatically give us θ_2. To get θ, we need to subtract the value that

PTO

Figure 2: A page from a modelling diary illustrating the writer's appreciation of code, unit vectors, and the importance of a diagram (reproduced with permission).

Conclusion

A course in classical mechanics has been designed in which material is presented formally, using VPython models to demonstrate concepts, and students spend time constructing models in VPython. The reasons for choosing VPython over other languages have been discussed. The emphasis throughout has been placed on the physics and modelling rather than on the computation. The models themselves are based on Newton's third law of motion, the concept which the FCI indicates is most lacking in these students and it has been shown how students have been supported in both computation, through the provision of extra contact time, and in their modelling through the use of a diary in which students record their reasoning about the models. Finally, the evidence from both observation and from the diaries points to significant spatial reasoning by the students.

References

1. Hestenes, D (1987) *Toward a Modeling Theory of Physics Learning*, American Journal of Physics, **55**, 440-454.
2. Chabay, R and Sherwood, B (2008) *Computational physics in the introductory calculus-based course*, American Journal of Physics, **76**, 307-313.
3. Hestenes, D., Wells, M., & Swackhamer G (1992), Force Concept Inventory, Physics Teacher, **30**, 141-158
4. di Sessa, Andrea (1987): *The third Revolution in Computers and Education*, J. Res. Sci. Teach., **24** (4), 343-367.
5. Hestenes, D. (1997), *Modeling Methodology for Physics Teachers*, in The Changing Role Of The Physics Department In Modern Universities, Ed. Redish, E. F. and Rigden, J., 935-957, American Institute of Physics Part II.
6. Halloun, I. and Hestenes, D. (1985), *The Initial Knowledge State of College Physics Students*, American Journal of Physics, **53**, 1043-1055.
7. Bishop-Clark, Catherine (1995): *Cognitive Style, Personality, and Computer Programming*, Computers in Human Behaviour, **11** (2), 241-260.
8. Redish, E. F. and Wilson, J. M. (1993) *Student programming in the introductory physics course: M.U.P.P.E.T.*, Am. J. Phys., **61**, 222–232.
9. Borcherds, P.H. (2007) *Python: a language for computational physics,* Computer Physics Communications **177**, 199–201.
10. Redish, E. F. (2003) Teaching Physics with the Physics Suite, John Wiley & Sons, USA.
11. Nersessian, Nancy J (2008) *Creating Scientific Concepts*, Bradford Books.
12. Monaghan, James M. and Clement, John (1999) *Use of a computer simulation to develop mental simulations for understanding relative motion concepts*, Int. J. Sci. Educ., **21**, (9), 921– 944.
13. Kohlmyer, Matthew Adam (2005) *Student performance in computer modeling and problem solving in a modern introductory physics course,* PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

> A common feature observed in all these classes is the extensive use of hand gestures to depict motion, directions, or spatial relationships. The use of similar gestures using simulations of relative motion has been cited as strong evidence for spatial reasoning.