IOPscience iopscience.iop.org

Home Search Collections Journals About Contact us My IOPscience

Using computational simulations to confront students' mental models

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2014 Phys. Educ. 49 195

(http://iopscience.iop.org/0031-9120/49/2/195)

View the table of contents for this issue, or go to the journal homepage for more

Download details:

IP Address: 193.136.24.142
This content was downloaded on 12/03/2014 at 15:01

Please note that terms and conditions apply.

Using computational simulations to confront students' mental models

R Rodrigues[1] and P Simeão Carvalho[1,2]

[1] IFIMUP, Universidade do Porto, Rua do Campo Alegre s/n, 4169-007 Porto, Portugal
[2] Departamento de Física e Astronomia, Faculdade de Ciências da Universidade do Porto, Rua do Campo Alegre s/n, 4169-007 Porto, Portugal

E-mail: rrodrigues.iutl@gmail.com and psimeao@fc.up.pt

Abstract
In this paper we show an example of how to use a computational simulation to obtain visual feedback for students' mental models, and compare their predictions with the simulated system's behaviour. Additionally, we use the computational simulation to incrementally modify the students' mental models in order to accommodate new data, and receive visual feedback for those modifications.

Introduction

We use conceptual models to abstract or simplify physical systems [1–3]. When students solve problem-based questions they use simple mental models to predict the functioning of some physical systems and give their answers [4, 5]. Like all simplifications, these models have flaws, and to be able to use them correctly one needs to be able to recognize when the models fail, and the limitations to their application. Cox *et al* [6] maintain that giving intentionally incorrect simulations to students helps to promote fruitful discussions and enhance student understanding. Another approach that we explore here is to increase the complexity of a physical model to describe a phenomenon by comparing the model predictions with the real behaviour of the system. Participative discussions between students are expected because, in general, many events have to be considered and possible physical explanations need to be explored.

The physical system

A ball is dropped from a given height into a large (bottomless) body of water. The ball's density is lower than the water's density.

Students should already have been introduced to the concept of gravitational and buoyant forces and the concept of friction, although it may not be necessary to know exactly how the drag force depends on the speed and geometry of the ball. They should be able to interpret the application of these forces to simple systems, i.e. to draw free body diagrams and to recognize that materials less dense than a fluid will float on it.

The software

To build the computational animation, we used VPython [7], the general purpose programming language, and a 3D graphics module called 'visual' developed by David Scherer. VPython allows one to easily create 3D simulations and it includes vectorial representation and operations, making it

0031-9120/14/020195+06$33.00 © 2014 IOP Publishing Ltd

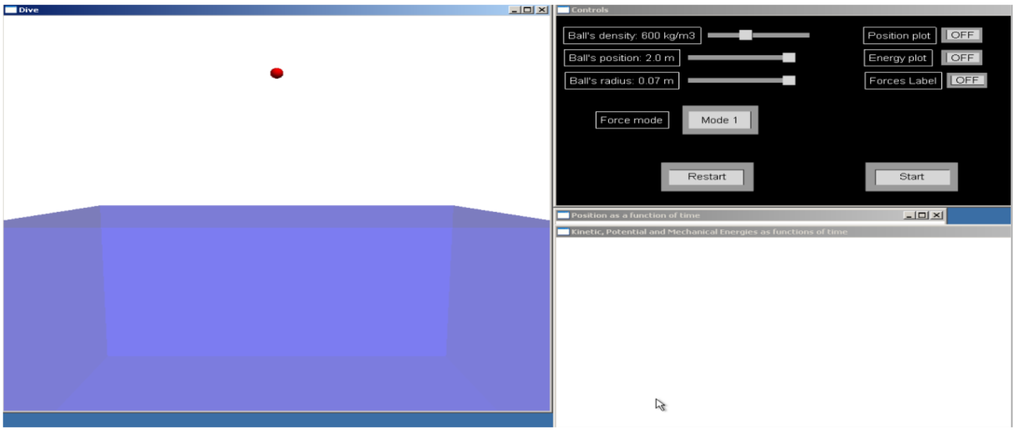

Figure 1. Image of the front panel of the simulation. On the left side we can see the animation running. On the right side there are buttons and sliders to control physical parameters, modes of operation and displayed graphs.

very useful for educational purposes. The simulation[3] allows users to change several parameters and the initial conditions of the motion, such as the ball's density, initial height or radius.

It is possible to run the simulation using different 'mental models'; i.e. we can consider that (i) gravity is the only force acting on the ball; (ii) only gravity and buoyancy forces act on the ball; (iii) gravity, buoyancy and water drag forces all act on the ball. After selecting these parameters, while running the simulation it also allows the generation of graphs for position versus time as well as kinetic, potential and mechanical energy versus time plots of the ball's motion.

Three forces acting on the ball are considered in the simulation: the downwards gravitational force given by

$$\vec{F}_g = m\vec{g}, \tag{1}$$

where m is the ball's mass and $\vec{g}$ is the gravitational acceleration; the upwards buoyant force given by

$$\vec{F}_b = -\rho_l V\vec{g}, \tag{2}$$

where ρ_l is the water density and V is the ball's volume; and the water drag force given by

$$\vec{F}_d = -\tfrac{1}{2}\rho_l v^2 C_d A\hat{v}, \tag{3}$$

where v is the ball's speed, C_d is the drag coefficient, equal to 0.42 for a rough sphere [8], A is the cross-sectional area of the sphere and $\hat{v}$ is the unitary velocity vector.

For simplicity, we have considered that the sphere can be treated as a point mass; this assumption does not compromise the main goals of the animation.

[3] This can be downloaded from www.fc.up.pt/pessoas/psimeao/VPhyton/dive.py.

Teaching strategies

A good exploration of the simulation is necessary to understand students' mental models, and provide opportunities for conceptual changes in their minds.

To begin with, the teacher can start by showing the simulation's front panel and explain to the students that it describes a ball being dropped into the water (figure 1). Knowing that the ball is less dense than water, they should consider the ball's motion after being dropped. Students may be organized into small groups to be able to discuss their predictions.

The simulation should be set with the default values: ball's position 2 m, ball's radius 0.07 m, ball's density 600 kg m^{-3}, and 'Mode 1' should be selected (only the gravitational force is considered). All plots and force labels should be switched off at this point.

As the simulation is going to run in 'Mode 1', the teacher must advise the students that the animation may not be totally correct, and their goal is to find out what is wrong and why. On running the simulation, they will see the ball passing through the water as if it was not there. Here, the discussion should be directed to identification of what causes the ball's behaviour, which force is present that makes the ball move downwards and sink, and what should happen to the ball in order to reverse the ball's direction and move it towards the surface.

It should be pointed out to the students that the simulation model is not wrong, but merely incomplete. The model fails to predict the ball's behaviour underwater, but it describes the ball's motion above it reasonably well. If our needs were limited to making predictions above the water level this model would be reasonable.

To describe the usual behaviour of a ball below the water level, a new question may be raised: 'besides the gravitational force, what other force could we add to the simulation model to make it more realistic?' Our program allows the buoyant force to be added to the simulation model with a single click (activating 'Mode 2'), but other suggestions from the students can also be added easily in the source code, if needed, as Python is an interpreted language. Before running the simulation with the buoyant force included in the model, the students should be asked to make predictions again, or confirm whether their initial predictions will now be correct. Other requests to the students may include a sketch for a position versus time plot of the motion, or calculation of the maximum depth the ball will reach.

Generally speaking, the ball's motion is described by the expressions [9]

$$y = y_0 + v_{0y}t + \tfrac{1}{2}a_y t^2, \tag{4}$$

$$v_y = v_{0y} + a_y t. \tag{5}$$

Choosing $y = 0$ m for the water surface and $v_0 = 0$ m s^{-1}, the equations of motion that describe the ball's motion while it is outside the water are

$$y = 2 - \tfrac{1}{2}gt^2, \tag{6}$$

$$v_y = -gt, \tag{7}$$

which means that the ball reaches the water with a speed of -6.3 m s^{-1}. Underwater, the total force ($\vec{F}_t$) acting on the ball is the sum of the gravitational and buoyant forces,

$$\vec{F}_t = \vec{F}_g + \vec{F}_b, \tag{8}$$

$$\vec{F}_t = \rho_b V\vec{g} - \rho_l V\vec{g} \tag{9}$$

and the total acceleration is

$$\vec{a} = (\rho_b - \rho_l)\vec{g}/\rho_b, \tag{10}$$

which, for the initial conditions chosen, results in $a_y = 6.5$ m s^{-2}. Therefore, the equations of motion that describe the motion while the ball is inside the water are

$$y = -6.3t + 3.25t^2, \tag{11}$$

$$v_y = -6.3 + 6.5t. \tag{12}$$

Therefore, the ball goes to a maximal depth of -3.0 m. If the students calculate this depth or sketch the position versus time plot, then they should compare their results with those of the simulation, which can be done by activating the position plot in the simulation panel.

After watching the simulation, the differences between the simulation and the predicted values should be discussed. Typically, students believe that the inclusion of the buoyant force will result in a motion where the ball ends at rest on the water's surface, but a perpetual motion is observed instead. It should be verified that the perpetual motion is expected in this model, for example, by finding the ball's velocity when it returns to the surface from underwater (to compare with the ball's velocity when it first reaches the water surface) and the ball's maximum height after jumping out of the water (to verify whether the ball jumps to the initial height).

Further inquiry should be engaged about the implications of the conservation of energy. The simulation assumes that the gravitational and buoyant forces are conservative (figure 2). However, students may notice that the ball's motion in the simulation seems 'unnatural', and in real life there is a loss of energy. This suggests that at least one other force (a non-conservative one) must be considered.

At this point, the students should notice that the simulation is not 'wrong'. In fact it is better than before, because previously the ball kept sinking and now it predicts correctly that the ball will return to the surface, although without a realistic motion.

Which non-conservative force should be assumed in the simulation, to describe a realistic motion?

It is now time to discuss the impact of the fluid (water) friction on the ball's motion. This friction acts as an interaction opposite to the ball's velocity, called the drag force, that always decreases the

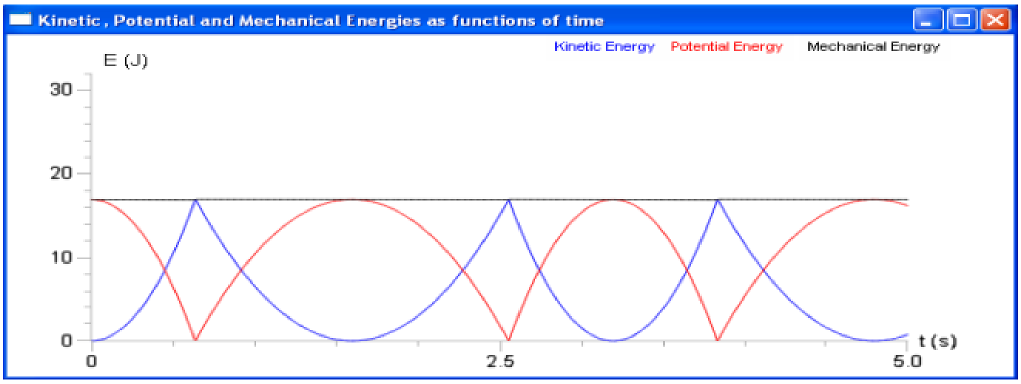

Figure 2. Plots of the kinetic, potential and mechanical energies from the simulation. Conservation of energy is observed when only gravitational and buoyant forces are considered in the physical model.

```
# gravitational force
Fg = ball.mass * g

# buoyancy force
if MODE == 1: # in this mode no buoyancy is considered
    Fb = vector(0,0,0)
else:
    Fb = -liquid.rho*ball.volume*g

# drag force
if MODE == 3:
    Fd = -k*ball.velocity.mag**2*ball.velocity.norm()
else: # in these modes (1 and 2) no drag is considered
    Fd = vector(0,0,0)

if ball.pos.y > 0:
    # Total force acting on the ball, when it is OUTSIDE the liquid
    Ft = Fg
else:
    # Total force acting on the ball, when it is INSIDE the liquid
   Ft = Fg+Fb+Fd
```

Figure 3. Detail of the source code of the simulation. The code with green background represents the insertion of the drag force in the ball's motion inside the liquid.

speed of the ball. Students must be told that this drag force depends on the ball's speed, in the form

$$F_d = -kv^2. \quad (13)$$

It can also be instructive to show the simulation's source code, so that the students are able to see where all the three forces are considered in the simulation, in particular the drag force (figure 3, with green background).

Before running the simulation again, the students should be prompted to predict, besides the ball's general motion, whether the addition of the drag force will change the ball's maximum depth in the water, and how. Additional tasks may include sketching the kinetic, potential and mechanical energy versus time plots.

On activating 'Mode 3' and running the simulation, one obtains a much more realistic motion of the ball. The students' energy graphs and the simulation energy graphs must then be compared. In particular, a line of inquiry may be followed asking 'why is the mechanical energy constant while the ball is outside the water, and briefly nearly constant underwater (for example, in figure 4, between 1.0 and 1.3 s)?' The fact that the drag force (the only non-conservative force in the model) is proportional to the speed squared implies that when the sphere's velocity is zero the drag force is also zero, so no work is done. This should be verified in figure 4 by noticing that the points underwater where the mechanical energy is apparently constant are those where the sphere's kinetic energy is nearly zero.

The discussion could end by enumerating some of the flaws that the final model still contains. Some of these flaws are: (i) the missing air drag force; (ii) the buoyant and drag forces always have constant magnitudes once the sphere is totally or partially submersed; (iii) the drag force is considered to be a function of speed squared; (iv) the water is considered to be undisturbed by the sphere's impact; (v) ultimately, even the constant gravitational force is an approximation. Some of these flaws are easy to eliminate, like the inclusion of the air drag force, but others are practically impossible to consider, such as the water surface disturbance. Therefore, a simulation will never represent the physical system exactly. The important point is the balance between the model's exactness and its complexity and recognizing when the model

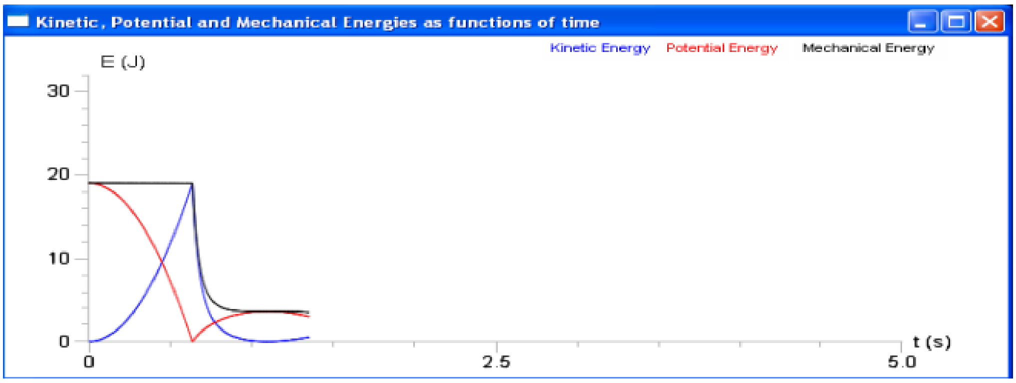

Figure 4. Plot of the kinetic, potential and mechanical energies from the simulation when ‘Mode 3’ is activated. Conservation of energy is not observed when the drag force is considered in the physical model. Other parameters are: ball’s position 2 m; ball’s radius 0.07 m; ball’s density 675 kg m^{-3}.

fails and when it predicts the system behaviour accurately. This should be highlighted in the final discussion with the students.

At higher level, an additional task may be requested in predicting how the sphere’s radius impacts how long the ball takes to reach equilibrium. Considering that the drag force is proportional to the effective contact area of the ball, while the ball’s mass depends on the ball’s volume, i.e. $\vec{F}_d \propto A$ and $m \propto V$, it results that the ball’s acceleration is $\vec{a} \propto R^{-1}$. Therefore, a larger sphere will experience a smaller acceleration due to the drag force. The buoyant and gravitational forces are both proportional to the sphere’s volume, so the contribution of these forces to the acceleration of the ball is independent of its radius. Therefore, we conclude that a smaller sphere reaches equilibrium faster than a larger sphere.

Conclusions

This paper shows an example of how to use simulations described by an incomplete set of physical laws to confront students’ mental models. An inquiry approach follows naturally when using this kind of simulation, which provides an immediate visual feedback useful for the generation of conceptual changes. This approach has the advantage of never reproducing incorrect physical simulations, but only incomplete descriptions of phenomena due to simple physical models whose complexity has to be increased. This gives an excellent opportunity for a challenging instruction of what to include in the model, in order to better describe the physical system’s behaviour.

When students work in this way, they are really doing problem solving and applying their knowledge to novel situations, and at the same time they are discussing the effects of each component of the incomplete model. In the example given in this paper, the separate effects of the gravitational force, the buoyant force and the drag force are clearly visible. Moreover, the use of interpreted simulations, therefore with open code such as the one here reported, allows the examination/study of the model used in the simulation and a simple change of parameter, if necessary.

One of the authors has already used this simulation with secondary school students. His experience is that teachers see their students becoming much more involved in physics, either because of the funny situations in the incomplete simulation (e.g. the jump of the ball out of the water in the absence of the drag force) or by the struggle to find the right way to reproduce the real phenomena (e.g. to conclude that the drag force slows the ball down and prevents it from jumping).

Acknowledgments

The authors are indebted to ‘Projeto Norte-070124-FEDER-000070’ and ‘Projeto PEC 259, Física Interativa: divulgação *low cost*, Agência Nacional para a Cultura Científica e Tecnológica’ for supporting this work.

Received 20 December 2013, accepted for publication 7 January 2014
doi:10.1088/0031-9120/49/2/195

References

[1] Hestenes D 1987 Toward a modeling theory of physics instruction *Am. J. Phys.* **55** 440–54

[2] Etkina E, Warren A and Gentile M 2006 The role of models in physics instruction *Phys. Teach.* **44** 34–9

[3] Vergnaud G 1990 La théorie des champs conceptuels *Recherches en Didactique des Mathématiques* **10** 133–70

[4] Gentner D and Stevens A L (ed) 1983 *Mental Models* (Hillsdale, NJ: Lawrence Erlbaum)

[5] Johnson-Laird P N 1983 *Mental Models* (Cambridge: Cambridge University Press)

[6] Cox A J, Junkin W F, Christian W, Belloni M and Esquembre F 2011 Teaching physics (and some computation) using intentionally incorrect simulations *Phys. Teach.* **49** 273–6

[7] Scherer D 2000 VPython—3D programming for ordinary mortals (available at www.vpython.org)

[8] Munson B R, Young D F, Okiishi T H and Huebsch W W 2010 *Fundamentals of Fluid Mechanics—International Student Version* 6th edn (New Jersey: Wiley)

[9] Halliday D, Resnick R and Walker J 2001 *Fundamentals of Physics* 6th edn (New York: Wiley)