

An educational system for modelling mechanics problems

0407847

Abstract

Newtonian mechanics is taught in schools and colleges as part of many physics and applied mathematics courses. Previous work showed that empirical modelling can produce useful teaching applications in this field by developing a simple simulation of masses moving under gravity in order to demonstrate a single physical law. There is a great deal of scope for this work to be extended by modelling dependencies between objects along with other physical properties such as friction and elasticity. This paper describes an attempt to build upon this previous work and create a set of tools along with an intuitive user interface which would allow a teacher to model a wide range of typical A-level standard mechanics problems quickly and easily. The problems encountered while doing this and the relative merits of using EM for this kind of project are discussed.

1 Introduction

This study was originally inspired by my former A-level mathematics teacher and the simple programs that he would write to illustrate Newtonian mechanics problems in classes. It occurred to me that many teachers would not have been able to write such models by hand due to lacking the necessary programming skills. I decided to investigate the suitability of using the EDEN empirical modelling language to implement a set of tools for teachers without pro-

gramming knowledge to construct simulations of such problems in the classroom. This task met with little success, as will be discussed, and for a number of reasons I do not think that EDEN is a suitable tool to attempt to construct such a tool, as hinted at in the revised abstract.

2 Review

2.1 Empirical Modelling Studies

There have been several relevant empirical modelling projects in previous editions of the Warwick Empirical Modelling Bulletin (WEB-EM). The two most similar physics related projects are the bicycle simulation in WEB-EM-01 (Bicycle Drive Chain Simulation), and the projectile simulation in WEB-EM-02 (An Educational Model for Newtonian Mechanics). These models are simple in their scope and model specific situations where the number of objects being modelled and the actions they can perform are small and well understood. These models demonstrate that simulations of mechanical and physics problems are possible in EDEN and can prove useful, but my intention was to construct a more generic framework which would allow users to construct their own models from a set of components.

While not a simulator of Newtonian physics, the ELS Eden based logic simulator from WEB-EM-03 is similar to my intended goal in that it allows a user to construct systems from a set of components, in this case electronic circuits made of logic gates, LEDs and similar. This type of free-form, construction set approach is closer to the concept I had in mind at the outset of this study. As will be discussed later, I encountered the same problems as the author of this particular study, to an even greater degree.

2.2 Other work

There are a number of non-EM related tools available which perform a similar task to the system I intended to construct. The Interactive Physics program from Design Simulation Technologies is the best example I have found. This program is aimed at my target users – secondary and sixth form level physics and mechanics students – and contains a comprehensive collection of objects along with the ability to connect them together in a vast number of ways to create interesting physics problems. This is precisely the type of tool I was originally intending to construct with EDEN. To experiment with this tool see:

<http://www.design-simulation.com>

Another system, aimed less at formal education but most definitely a fascinating piece of software, is “Phun!”. This ingenious and addictive program provides an accessible, intuitive and pleasingly cartoon like interface for constructing systems of surprising complexity using a variety of components. This program includes an interface and physics engine with such features as simple fluid dynamics, collisions, and the ability to create surfaces of any shape and size with a simple mouse movement, and as such is vastly beyond the level I would have ever hoped to achieve with this study even at its outset. However, it does illustrate very well the power and usefulness of well built, easy to use educational software in this particular field. Phun! is a very addictive piece of software simply to casually play with, and it is easy to imagine that if used in an educational context, it could encourage a level of engagement and understanding far greater than is possible by simply studying problems on a blackboard or using more formal educational software such as the previously mentioned Interactive Physics. See <http://www.vrlab.umu.se/research/phun/> for more information.

3 Problems

As stated at the start of this paper and in my original abstract, my intention was to construct a set of tools from which users would be able to construct physics simulations. After studying the problem in more detail, it became apparent that it would not be possible to achieve this goal given the time available, my knowledge of EDEN, and the unsuitability of EDEN for this task.

3.1 Complexity of the task

The original concept for this project involved modelling all, or a significant subset, of the objects that are considered in the Advanced-Level mechanics syllabus in the UK, which I am familiar with as a product of this education system. This would include elastic and inelastic strings, projectiles, rough and smooth inclined planes, collisions, and simple harmonic motion. After considering some possible systems several problems became apparent.

While constructing a specific mechanics problem in EDEN from scratch is quite possible and relatively straightforward, as demonstrated by the model I *have* constructed, when trying to anticipate all the possible ways in which these components can be put together and interact the task quickly becomes unmanageable. Firstly, as was found by the author of the EDEN Logic Simulator study, anticipating the way in which components even as simple as logic gates can be connected, in such a way that they interact correctly and circular dependencies are avoided, is extremely difficult. It quickly becomes impossible to do this procedurally with more than a few components. In comparison to the logic simulator study this problem is significantly worse with most mechanics problems since there are many more interactions to model than current flow and circuit logic states. Sticking to the intended goal of allowing any possible combination of components and correctly modelling all their interactions would mean anticipating the

effects of gravity, friction, momentum, elasticity, surface reaction and other effects while resolving all the relevant forces into the same basis vectors. Doing this in my manually constructed model was quite difficult, attempting to write code to generate all the objects, connect them together via a graphical user interface in the correct way, and then construct the underlying physics model dynamically, was a task far beyond my skills. The main problem in attempting to do this lay in the need to keep the number of assumptions made to an absolute minimum. When solving typical mechanics problems manually, it is possible to use the properties of the task before you to greatly simplify the amount of work you need to do. An example is the model that I actually constructed to show the way in which Empirical Modelling can be somewhat useful in modelling this type of problem. This model, discussed in more detail later, involves two blocks modelled as point masses, an inelastic string on a pulley, and an inclined surface. One block rests on the surface, the other hangs freely in space. In this particular case most of the forces involved act on the block which rests on the plane. A student solving this manually would model the forces involved as vectors, each with two components, one perpendicular to the plane and one parallel to it.

When trying to create a completely generic system it is not possible to build in this kind of intuitive problem solving and it is necessary to resolve all the forces in the horizontal and vertical directions, significantly increasing the complexity of the physics model you have to build.

3.2 Limitations of EDEN itself

The other source of the problems I encountered came was the nature of the EDEN language itself. EDEN is still very much a work in progress, with limited and sometimes incomplete documentation and lim-

ited functionality, as well as a number of irritating syntactic differences between the different notations – EDEN, DONALD, SCOUT etc. – which make up the EDEN software. The main limitation with EDEN, however - as was also found in the EDEN Logic Simulator study - is its lack of support for object-oriented programming. Attempting to build a system consisting of a number of similar components would have been made considerably easier, or perhaps at least feasible, with the power of object-orientation, and the lack of this ability within the EDEN language makes creating copies of an object an extremely painful process.

4 The Model

4.1 Features

Given the problems that I have outlined and the fact that my original goal was not achievable, I set out to build a model which would demonstrate the ways in which Empirical Modelling *can* be useful in this type of situation. The model I have constructed is one of a fairly typical mechanics problem. This consists of two blocks, modelled as point masses, a rough inclined plane, and a rope and pulley system connecting the two (fig 1).

The model includes a SCOUT GUI which allows a user to adjust various parameters within the model. The plane can be inclined at any angle within an arc of 178 degrees (i.e., any gradient except the vertical). This is adjusted using two buttons which increment (or decrement) the angle (fig 2). The user can also adjust the mass of the two blocks and the friction between the block and the plane. Finally, the user can enter a time in seconds and the interface will display the time, velocity and distance moved by the two blocks after that number of seconds has elapsed (fig 3). This model would be suitable for a student or teacher to check their answers for a similar textbook prob-

lem, or experiment with the way the different properties of the system interact with each other.

4.2 Limitations

There are a number of limitations to this model which I will now outline. Interacting with the GUI should not produce any incorrect results, but given the nature of EDEN and the fact that all the parts of the models internal state are accessible by entering commands into the interpreter it is possible to induce strange and incorrect results. This is due to some of the assumptions I have made within the physics model in order to simplify it. For example, the model assumes that the plane is never vertical or inverted. If one causes this to happen, the block which rests on the plane will not fall off, but will remain attached. The model resolves the forces acting on this block under the assumption that at least part of its weight will be directed into the plane. Also, the model assumes that the rope is parallel to the plane. Moving the pulley upwards will not cause it to lift the block off the plane, since the force of tension in this part of the rope is assumed to act in the direction of the plane. These are just two simple examples of the kind of assumptions which make modelling well defined problems like this easy, but open ended customisable systems very difficult.

4.3 Benefits of using EDEN

Being given the ability to apply some constraints to the system being modelled and make some assumptions about it made the process of developing this model relatively straightforward. While this still took some time as I was relatively inexperienced with the EDEN language, I doubt that I would have been able to implement this more quickly or easily in any of the standard programming languages that I am familiar with. When the system being modelled is well defined, EDEN is an adequate choice

for the task, and if EDEN was to be developed further, to the point where it has a slightly more robust set of features and comprehensive documentation, I feel it would be a very good choice for modelling a specific problem like this. Given the rough and ready state of EDEN and the lack of any kind of IDE to speak of, it is also surprisingly easy to debug code written in EDEN, due to the ability to change the model easily while it is running. This is another feature of EDEN which makes it quite suitable for this type of activity. Also, it is quite common for students to be given a number of problems which feature either a very simple or identical set of objects, but a different set of known and unknown quantities. The ability to access any part of the model relatively easily by entering commands in the EDEN interpreter is helpful when modelling this kind of problem, since it is possible for a user provided with a set of basic instructions to access any of the quite large number of forces acting within the system.

5 Further Work

5.1 Improvements to this model

There are a number of possible enhancements to this model. The model deliberately ignores physical concepts in order to remain consistent with the content of an Advanced Level mechanics course – for example, there is no air resistance and the friction model is rather simplistic. Currently these forces are simply not implemented, but it would make the model more potentially useful if it was capable of handling them. Also, a more realistic tension model capable of modelling elastic strings and lifting the block off the plane would be useful. Finally, the ability to define the length of the plane and the ropes and animate the model using EDENs clocking features would also be helpful. Currently the model does not do this as, from my recollection of studying this type of problem, it

would be more useful to simply access the acceleration, velocity and distance figures without cluttering the interface with definitions for the various lengths. However, it may be useful to some users to have the option.

5.2 Further modelling study

While I would not consider, or recommend, another attempt to achieve my original goal using EDEN, I believe it might be interesting to construct models of more of this type of mechanics problem using EDEN. While it is not a suitable tool to allow those without programming knowledge to construct their own models, nor is it suitable for constructing such a tool, those who do have some programming knowledge and skills might find EDEN to be quite a good choice for such activities, particularly if its documentation were improved.

6 Conclusions

While I have not managed to achieve the original goal of this study, I believe it has been an interesting exploration of the potential of Empirical Modelling, and EDEN in particular, in this area of educational software. EDEN is not particularly suited to developing applications like the one I originally envisaged, but for a programmer wishing to model a particular problem it is quite an effective tool to use.

7 Suggested Weighting

Model:Paper 60:40

Bibliography

An Educational Model for Newtonian Mechanics. (n.d.). Retrieved from The Second Warwick Empirical Modelling Bulletin:
<http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/web-em/02/newtonian.pdf>

Anon. (2007). ELS - An Eden Based Digital Logic Simulator. *The Third Warwick Empirical Modelling Bulletin*.

Beynon, W. M. (1997). *Empirical Modelling for Educational Technology*. Washington DC, USA: IEEE Computer Society.

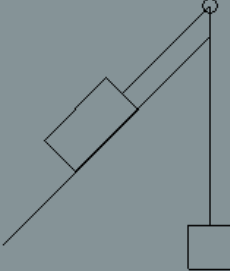
Bicycle Drive Chain Simulation. (n.d.). Retrieved from The First Warwick Empirical Modelling Bulletin:
<http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/web-em/01/bicycle.pdf>

Bodin, K., & Ernerfeldt, E. (2008, 02 12). *Phun - visual and interactive 2d physics*. (VRlab - Umea University) Retrieved 02 20, 2008, from
<http://www.vrlab.umu.se/research/phun/>

Design Simulation Technologies. (n.d.). *Interactive Physics*. Retrieved from
<http://www.design-simulation.com/IP/index.php>

Appendix: Screenshots

7x screen (tkeden 1.70)



Angle of incline (degrees)
+
-45.000000
-
Mass of resting block (kg)
1
Mass of hanging block (kg)
1
Coefficient of Friction
0.2
Time (seconds)
0
Friction: 1.414214 N
Normal Reaction: 7.071068 N

Resting block:
Acceleration: 1.514719 m/s/s
Velocity: 0.000000 m/s
Distance travelled: 0.000000 m

Hanging block:
Acceleration: 1.514719 m/s/s
Velocity: 0.000000 m/s
Distance travelled: 0.000000 m

Figure 1. Initial view of the model

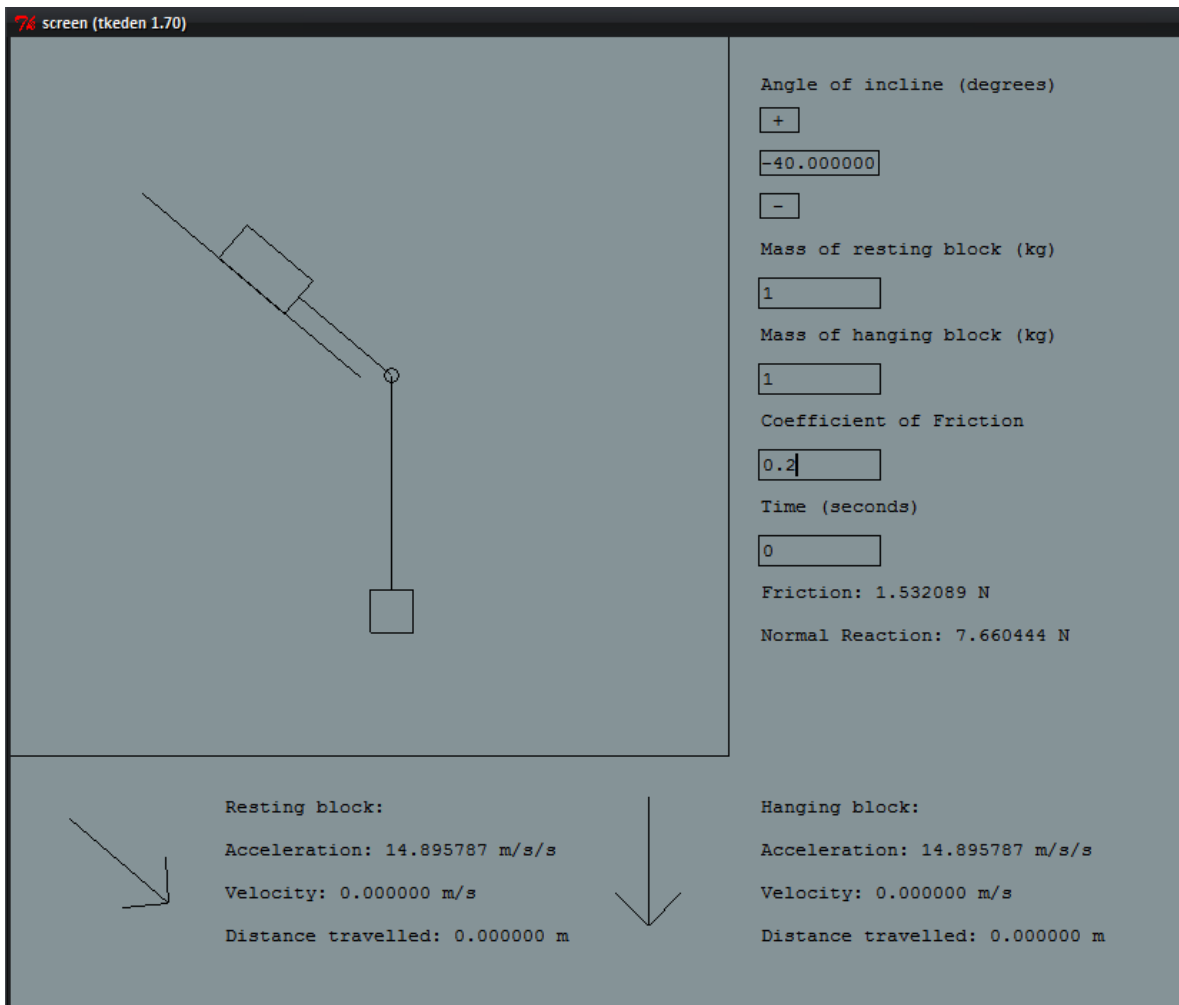


Figure 2. Model with plane inclined in the opposite direction

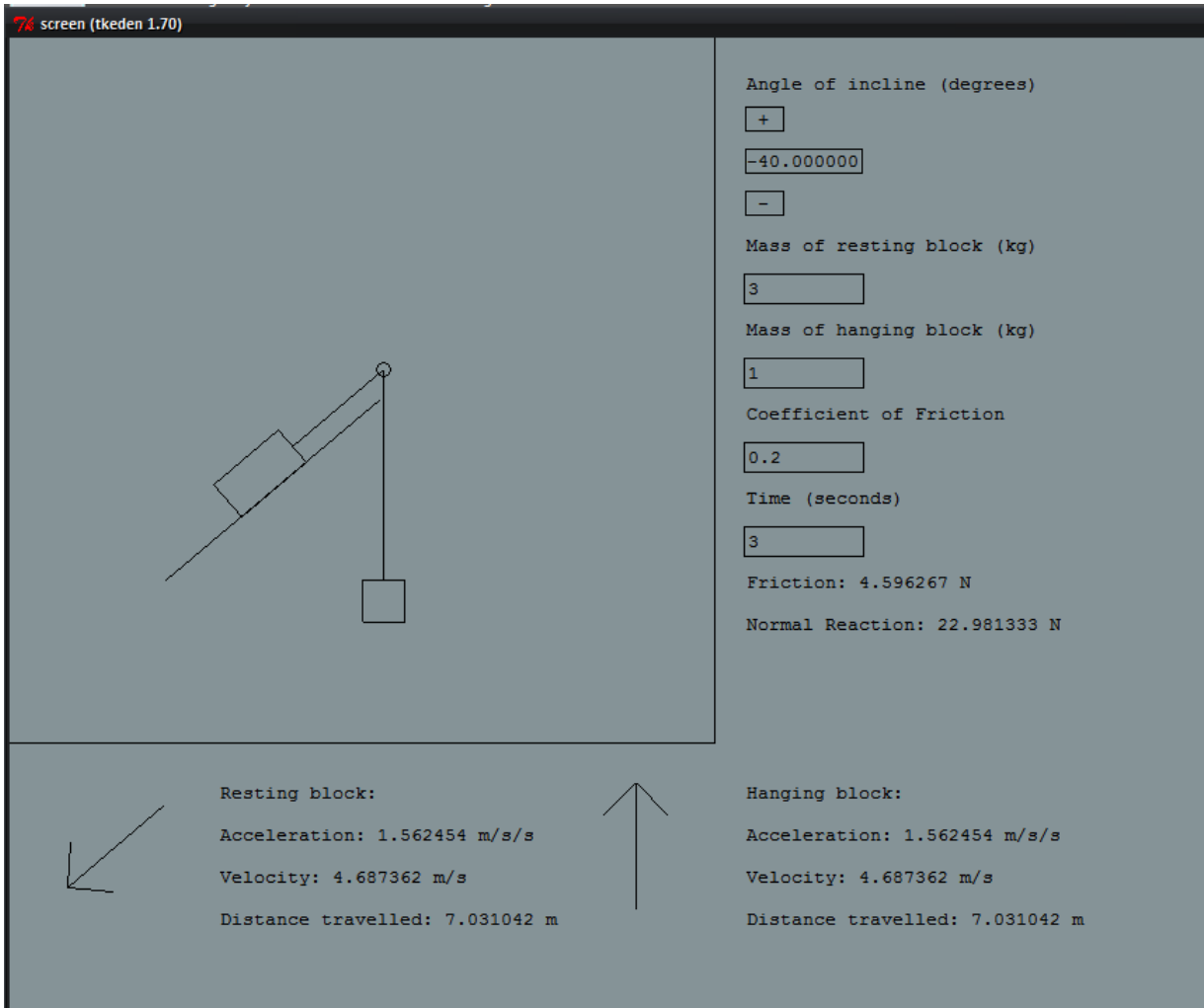


Figure 3: Shows blocks moving in the opposite direction, along with distance and velocity calculations