The use of the commercially available software package Mathematica, which is capable of both solving equations and visually presenting the results, is described to assist in the teaching of an introductory course in dynamics. Improving instruction within the classroom environment and assisting the students’ learning outside the classroom are goals of this work. The focus of the development presented herein is on the beginning portions of the course, concentrating on mathematical preliminaries and particle dynamics.

Mathematica, frequently employed to teach a variety of topics, is used to parallel and illustrate the formal development of equations in a manner consistent with the textbook. Mathematica lends itself to this purpose, as its syntax is not unlike that found in textbooks. The software also allows for user input thus allowing students to vary the parameters defining the problem and to see the variations in the final results. In addition to containing dynamics problems, the Mathematica notebooks also address the more fundamental, but often more abstract, development of critical mathematical relationships. (The term “notebook” refers to a Mathematica file.) For example, some relationships demonstrate the benefits of choosing particular coordinate systems while others help students understand the components used to describe motion along a path by highlighting the intrinsic triad, i.e., the normal, tangential, and binormal unit vectors. The relationships between position, velocity, and acceleration, as well as the effect of taking the derivative of a vector with respect to time are also presented. This paper will discuss our experiences in creating these Mathematica notebooks, will present some examples of notebooks we have created, and will provide advice for instructors wishing to create notebooks of their own. It is hoped these innovative approaches will help educators to better illustrate and will help students to more easily grasp fundamental concepts that are crucial in understanding dynamics.

Introduction

There are a great variety of tools and teaching methods available to lecturers who are providing instruction to engineering students in today’s colleges and universities. The choices made among these many options are often due to the particular backgrounds and
interests of an instructor. The students in the classroom also bring a diversity of experiences and learning styles to the student-teacher relationship. Keeping in mind the responsibility of both instructors and students to effectively communicate with one another as well as to adequately prepare for learning outside of the classroom, engineering instructors should be interested in considering different ways of presenting course material. The work associated with this paper concentrates on using Mathematica, a mathematical computer package produced by Wolfram Research, to

- present information necessary to gain an understanding of dynamics,
- present example problems similar to traditional homework problems,
- show the derivation of equations necessary for grasping fundamental theory as well as for solving typical problems,
- act as models that may be manipulated, allowing students to explore the relationships among various entities that define a given problem,
- emphasize formulating equations as opposed to simply obtaining a numerical solution; and
- make use of visual output, e.g., graphs and movies generated from the equations of motion.

The goal of this work is to produce, and to help others to produce, Mathematica notebooks that will supplement a traditional course in undergraduate dynamics by providing additional resources to the lecturer to be used within a classroom setting and by providing to students examples to emulate in their own work. To support this goal, the authors of this paper intend to relate their own experiences in creating notebooks to aid others in creating their own notebooks. To this end, a small number of concepts are presented with the hope that this will generate feedback from readers as to the types of concepts they would like to see addressed in this format. In addition, this paper includes a number of useful suggestions on the use of Mathematica, since these types of suggestions are quite helpful to a notebook author and are difficult to find in the literature. The authors wish to emphasize that this work is in its infancy and that our goal with this paper is not to present the completion of a study, but our experiences with the beginning of a study.

Since time is a valuable commodity (especially within the limitations of standard lecture periods), since students may have no background whatsoever with a particular software package, and since the aim of an undergraduate dynamics course is not instruction in a particular package’s syntax, it is necessary that there be no significant extra work required of students who will be utilizing these notebooks. Also, it is important to keep in mind that the notebooks in this work are aimed towards all students taking a dynamics course, e.g., not only honors students, so that the representations of ideas must be unambiguous. As much as possible, the notebooks should follow standard mathematical nomenclature.
so that students may conveniently apply what they see in their calculus and dynamics textbooks to what they see on the computer screen or projector. Equations should be easily evaluated and any graphical output or displays should likewise be straightforward. In addition, it would be helpful if there were simple ways of utilizing additional software packages offering alternatives for presenting information in formats not relying on the availability of Mathematica, e.g., QuickTime or PDF.

In the next section, some examples of the kind of work others have produced to aid engineering education will be presented along with the experiences of two of the authors in teaching an Interactive Dynamics course. There will then be a homework style example problem shown. This is the type of notebook students may develop for themselves to help in solving a typical problem. Following that section, some questions will be posed to assist a potential developer in defining the scope of an electronic presentation. The section entitled Implementation presents some ideas to assist in developing a Mathematica notebook, although this section is not intended to be a manual on the use of Mathematica. Several examples are presented in the section Examples that highlight the different types of information being passed on to the students and the different learning processes that may be emphasized using these presentations. There is also a concluding section relating some final thoughts.

Instructor Experience and Student Response

As alluded to in the Introduction, there are many ways electronic presentations may be used to improve the educational process. It is first worth mentioning a few examples from the literature. Some of these examples involve much more extensive use of mathematical and modeling packages, moving beyond a mere course supplement. For example, the world wide web may be used as a means of presenting both lectures and computer-generated examples to students completely apart from the traditional classroom setting.\(^4,12\) Mathematica may become central to the course, either by associating each lecture or topic with a given notebook,\(^10\) or even by creating an entire textbook based on the software package.\(^9\) If students are required to become proficient with Mathematica, they would then be able to create their own notebooks and to solve their own problems.\(^1,11\) Similarly to what is being described here, others have written software to create interactive modules for their students to explore and to better understand various physical systems.\(^5\)

Previous work has been done to encourage the use of computers in the classroom as part of an overall interactive environment to be used in teaching engineering and science courses.\(^3,6\) Interactive Dynamics (ID) was an experimental program for the creation of a new learning environment for the delivery of undergraduate dynamics. The objective of this course was to reform the delivery and content of undergraduate dynamics, the first engineering undergraduate course in dynamics. ID was designed to engage students in a collaborative environment in which all students have easy access to an array of modern technological tools to:
• analyze data (often generated in real time in class);
• observe graphic representations of the data; and
• construct as well as interact with computer simulations.

In ID, students spend large portions of their class time learning "actively" by working in small collaborative groups to analyze physical phenomena (sometimes captured on videotape), using elements of numerical analysis to study and visualize the motion of objects, and presenting their findings in reports required to be professionally written.

ID was therefore designed to be a guided, problem-based learning environment where students develop and sharpen their engineering as well as communication skills. Furthermore, this environment was designed to provide the students with a more realistic “workplace” experience of the analytical/computational component required by machine or process design. With this in mind, we used Mathematica extensively in the classroom to show how it can be used to remove the tedious algebra from the kinds of problems they were encountering in their homework. In addition, we used it to show the motion associated with many of the solutions obtained in class in the form of animations. We encouraged students to use Mathematica in their homework and required them to use it on the team-based projects that they did in class. As is to be expected when learning a new computer language, the students had some initial difficulties with the Mathematica syntax. However, the vast majority of them were able to quickly pick up the new syntax and the students’ feedback at the end of the course was generally positive. The one thing that we did find, almost with no exception, was that the students wished they had had an introductory tutorial on Mathematica the semester prior to that of ID or at the beginning of ID.

An Example Notebook for Student Use

Before continuing with the development of notebooks for classroom use, it is worth giving an example of how a student might develop a notebook for his or her own use. A typical dynamics system is shown in Figure 1. Instead of asking for a velocity at a particular configuration, e.g., when θ is specified, assume that the student is required to plot the velocity of the slider at C for a known θ̇ (where the dot refers to taking a derivative with respect to time) and known values of ℓ₁ and ℓ₂. Therefore, the first goal is to write an equation for the velocity at C in terms of the angle θ.

Figure 2 is a screen shot showing the development of just such an equation from a Mathematica notebook. The position of the slider at C, given as \( y_C = ℓ₁ \cos(θ(t)) + ℓ₂ \cos(ϕ(t)) \), is entered. Next, two equations are entered. The first one defines the kinematic constraint, \( ℓ₁ \sin(θ(t)) = ℓ₂ \sin(ϕ(t)) \), and the second one is a basic trigonometric identity, \( \cos(ϕ(t)) = \sqrt{1 - \sin(ϕ(t))^2} \). Other than naming these latter two equations, all three entries have a very recognizable form within Mathematica.
Figure 1. This is an image of the slider-crank homework style problem discussed. The lengths of each segment are given by $\ell_1$ and $\ell_2$ and the position of the slider at C is given by $y_C$. The angles $\theta$ and $\phi$ are defined as shown in the schematic drawing on the right.

One way of continuing is to make use of the `Solve` command to first solve for $\sin(\phi(t))$, $\cos(\phi(t))$, and $\dot{\phi}(t)$. These terms come from the kinematic constraint equation, the derivative of the kinematic constraint equation, and the trigonometric identity. This command is simply of the form `Solve[{equations},{unknowns}]`. The final step involves taking the derivative of the position function with respect to time (to get the velocity) and applying the solutions of the terms involving $\phi(t)$. This is accomplished through the replacing operator, symbolized by `/.`. The `FullSimplify` command simplifies the solution, which is called $v_C$ or the velocity at C in terms of $\theta(t)$, $\dot{\theta}(t)$, $\ell_1$, and $\ell_2$. The last three terms are taken to be given in the problem statement.

Mathematica may then be used to plot a variety of solutions for different values of $\dot{\theta}(t)$, $\ell_1$, and $\ell_2$. Three cases are shown in Figure 3. In each case $v_C$ is plotted as $\theta(t)$ goes from 0 to $2\pi$. The replacement operator is once again used to apply the numerical values of $\theta(t)$, $\ell_1$, and $\ell_2$ for the three cases. By only varying the value of $\ell_2$, the student is able to see the effect this has on this dynamical system. Nothing was done to improve the appearance of the output or automate the input for this problem in an effort to keep things as simple as possible. This example demonstrates both the similarity between Mathematica’s syntax and conventional mathematical nomenclature as well as some instances where basic knowledge of Mathematica would be necessary to solve certain useful problems.

**Defining the Notebook Contents**

It is helpful to consider the ultimate goal of a particular notebook before beginning to create the notebook. A number of questions will be posed to aid in this decision. How much time will be spent using the notebooks in this course? Is this a single demonstration or will...

---

*The typewriter font is used to designate commands, e.g., `Plot[Sin[x],{x,0,3.14}]`. 

Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition 
Copyright © 2003, American Society for Engineering Education
Figure 2. This is a screen shot showing the derivation of the solution to the homework style problem using Mathematica. Plots of the solution are shown in Figure 3. The first line of the notebook simply turns off Mathematica's built-in spell checker, which can be rather annoying.

Will these notebooks be a part of each and every lecture? The more frequent the use, the more consistent the appearance should be to avoid confusion—this may require more planning in the beginning.

Will these notebooks be used by the lecturer who is also the creator of the notebooks, another lecturer, the students outside of the classroom, or a combination of these possibilities? The answer to this question will have implications for how robust and how well documented the final product must be. In the context of this project, the notebooks must be sufficiently portable as the primary intention is for them to be used by any lecturer during a classroom lecture. The notebooks may also be considered as templates for the students to refer to in solving their own problems since the essential details of the notebooks will not be hidden.

Is the focus of the notebook on theory and mathematical foundations or does the notebook contain an example problem, e.g., a traditional homework problem? Presentation of theory tends to be more abstract and yet also fundamental to a student's understanding of the course. Therefore, this type of notebook may take more thought on the part of the author, as it
Figure 3. Mathematica is used here to plot three different solutions, for different ratios $\ell_1/\ell_2$.

may not be so obvious how to best demonstrate more theoretical ideas. An example problem should be solved in a recognizable manner, consistent with the textbook or lecture.

*How much and what type of user input is necessary to use this notebook?* At one extreme, the user does little more than look at a generated picture or view an animation. At the other extreme, the user may be required to enter numerical values for variables and subsequently evaluate the notebook to see the final results (and also the effect of choosing certain values). Somewhere between these extremes, the user may be able to select from certain pre-defined situations or set parameters within a given range. The choice made here will greatly affect the complexity of the resulting notebook.

*What type of output is desired?* The results for a given problem may be best described with still graphs (two- or three-dimensional) or animations. If the principal focus is on the observation of derived equations or relationships, there may be no graphical output produced whatsoever. At this point, one should also consider whether to use other software.
packages to present the output, as was mentioned before regarding the use of QuickTime.

Finally, other than looking at the results, is the student expected to do anything more? For example, the student may be expected to answer questions based on seeing the final output. The student may also be expected to use the notebook as a model for solving similar problems. The answers to all of these questions within this section will determine the type of notebook that will be produced. It is hoped that the variety of examples presented in the Examples section will encompass many of the issues raised here.

Implementation

In Mathematica, the contents of a notebook are either input cells, where commands are entered, or output cells, where results are displayed upon evaluation of input cells. In both cases, it is helpful to consider what parts of either the input or output the student should see and what parts should be hidden. Of course, the parts that are seen must abide by the requirements discussed in previous sections, namely, they must be unambiguous and helpful to the student. Commands that are hidden are typically those beyond the scope of what students would be expected to comprehend, e.g., complex graphics directives. Deciding what to hide and what to show is central to the purpose of the specific application.

Hiding a cell or cells is accomplished by placing the cell or cells within a given section and collapsing the section. The contents of a cell may also be hidden by highlighting the cell and selecting the Option Inspector\(^*\) from the Format menu. Under the General Properties of Cell Options set the CellOpen option to False as shown in Figure 4.

It is suggested that the latter technique be used for the more obscure syntax that a student has no need to see, or be confused by, while the former technique is appropriate when optional—and greater—detail is made available. When hiding commands, although the rules for clarity may be somewhat relaxed, effort should still be made to make the commands as understandable as possible.

There are additional ways to hide output. A semicolon may be added after a command line to suppress text output. Often it is useful to hide graphical output, only to display it later in the notebook. This is desired when one combines multiple graphics objects together into one final object, but there is no need to see each piece separately. Setting the DisplayFunction to Identity returns the graphics object without displaying it, while resetting the function to $DisplayFunction will then display the object, as shown in Figure 5.

The other case where hiding output is useful occurs when creating animations. Mathematica creates animations by creating a table of graphical cells in order of display and then collapsing the cells together such that, when they are selected, each cell is displayed in

\(^{*}\)The sans serif font is used to designate menu options, e.g., File.
time to produce a final animation effect. The following lines may be added after creating a table within the same cell to automate this collapsing process:

```mathematica
SelectionMove[EvaluationNotebook[], After, EvaluationCell];
SelectionMove[EvaluationNotebook[], Next, CellGroup];
FrontEndTokenExecute["OpenCloseGroup"]
```

Mathematica will still generate each frame; it will now collapse them together after finishing the process as shown in the example given by Figure 6.

Before moving on to the non-hidden parts of the notebook, it is worth mentioning the evaluation process within Mathematica. Instead of selecting each cell to be evaluated, it is possible to create a button with a function that will automatically evaluate the cells of interest. This is especially useful if the intention is to have students evaluate the notebooks themselves. In addition to simplifying the evaluation process, this also ensures the correct evaluation of cells as intended by the notebook author.

Regarding the visible input, it is important that the syntax is obvious and parallels that which is being used on the chalkboard and in the textbook. There should also be adequate explanation and directions given, making use of descriptive titles, text, and still figures showing variables. If the students are required to enter numerical values, the exact form of the entry should be shown as text so there is no confusion with the syntax. Even if the lecturer is the only user of the notebook, it is still helpful to provide sufficient comment

Figure 4. This is a screen shot showing the Option Inspector menu opened after selecting it under the Format menu item.
Figure 5. This is a screen shot showing the use of the DisplayFunction command to hide graphical output.

with the commands; if the notebook will be used by others, such comments are essential. Mathematica also provides a number of Style Sheets found under the Format menu to improve the look of the notebook.

Generating visually pleasing graphical output can often require more effort than generating the results themselves. Sophisticated coding may be required to handle all the possible variations in user input. It may be necessary to adjust the size of objects or the plot range based on the user’s input. A simple example of this was actually given in Figure 6 (the first few frames of the animation are shown in Figure 7). Seeing axes move is rarely a good thing in this situation and so it would be better to specify the same range for each image by setting `PlotRange→{{0,3.14},{-1,1}}`. It is a good idea to look at the work others have done, some shown here and much more available on the web, and carefully consider what one is trying to demonstrate. In the beginning, let Mathematica produce graphical objects for relatively simple problems. One may then continue to add complexity as one works more with these problems.

Examples

A number of examples are presented that are appropriate for an undergraduate course in dynamics. The first three examples focus on more abstract concepts: the idea of a closed path and the intrinsic triad of the normal, tangential, and bi-normal unit vectors along a particle’s path (Example I); the time rate of change of a vector (Example II); and the ideas behind selecting certain coordinate systems or components based on a particle’s
Figure 6. This image is taken from a notebook showing how the thirty-one images generated with the `Table` command appear collapsed together automatically using the specified three command lines. All command lines, including the `Table` command, occur within the same cell.

path and information of interest (Example III). The remaining three examples (Examples IV–VI) are more typical of homework problems, although in each case additional, more theoretical, results are also generated. These materials are located on the web at <http://lpcm.esm.psu.edu/DynamicsNotebooks/>.

Example I: The closed path and intrinsic triad  This is a rather simple example that at first generates a single image showing a two-dimensional function (in $x$ and $y$) that changes in time ($t$), Figure 8. When the function is projected through time onto an $x$-$y$ plane, the path appears to be closed. The path may also be plotted using time as the third axis to clearly see that there is a difference between the total distance travelled and the absolute position. This situation arises in many dynamics problems and will appear in Example IV.

A related function, shown in Figure 9, may also be used to teach about the intrinsic triad (described by a unit vector tangent to the path in the direction of the velocity vector, a unit vector normal to the path pointing in the direction of the radius of curvature, and a unit vector bi-normal to the path). This figure is generated by a vector tracing out a three-dimensional path in $x$, $y$, and $z$ through time, $t$. (Recall that Figure 8 showed a two-dimensional path in $x$ and $y$ through time.) The three components are defined by the following relationships:

$$x(t) = \cos \left( \frac{t}{2} \right) \quad y(t) = \frac{1}{3} \sin(t) + \frac{1}{2} \cos \left( \frac{t}{2} \right) \quad z(t) = t.$$
Figure 7. This shows some of the first images that make up the animation given in Figure 6. As can be seen, the range along the \( y \)-axis changes in each image.

The functions for \( x(t) \) and \( y(t) \) are identical to those from the first case, as is apparent from the input commands included in the image presented in Figure 8. For the three-dimensional example, the intrinsic triad’s unit vectors are determined within the notebook and the results are presented in an animation showing the changing triad as it winds along the cork-screw path. Figure 9 shows a single frame from the animation. The three unit vectors, labeled by Mathematica, are \( \mathbf{u}_T \) (tangent to the path), \( \mathbf{u}_N \) (normal to the path), and \( \mathbf{u}_B \) (bi-normal). During the animation, students will see \( \mathbf{u}_N \) flip as the instantaneous radius of curvature abruptly changes.

Example II: Time rate of change of a vector  

The next example is designed to give an understanding of the meaning behind taking the time derivative of a vector (shown by using a dot, \( \dot{\cdot} \), over the vector). In this case, for a vector that is a function of time \( t \), \( \mathbf{A}(t) \), the idea is to demonstrate that \( \dot{\mathbf{A}}(t) = \dot{\mathbf{A}}(t)\mathbf{u}_A + \mathbf{\omega} \times \mathbf{A}(t) \). The vector \( \mathbf{u}_A \) represents the unit vector in the direction of \( \mathbf{A} \) and the vector \( \mathbf{\omega} \) refers to the angular velocity of \( \mathbf{A} \). The scalar quantity \( \dot{\mathbf{A}}(t) \) is the time rate of change of the magnitude of \( \mathbf{A} \). Students most often encounter this situation when working with vectors describing position and velocity.

To reinforce the idea that the time derivative of a vector may be thought of as being the sum of two vectors, a notebook has been created to animate three different vectors simultaneously changing in time. A single animation cell is shown in Figure 10. A legend defining the vectors is shown in the bottom right-hand corner (in the actual notebook, these objects are shown in color). The top left-hand figure shows a vector of constant length rotating with a constant angular velocity, \( \dot{\mathbf{A}}(t) = 0 \). The top right-hand figure...
Definition of a Closed Path

\[ x = \cos \left( \frac{t}{2} \right); \]
\[ y = \frac{1}{3} \sin t + \frac{1}{2} \cos \left( \frac{t}{2} \right); \]

Figure 8. This figure is a still image from a notebook showing a path in two dimensions of space (x and y) and in time (t). The path is also shown projected in time such that it appears as a closed path.

shows a vector only changing in length at a constant rate, \( \omega = 0 \). By combining these two effects, the bottom left-hand figure shows a vector changing length while simultaneously rotating. The vector \( \omega \) is actually perpendicular to the page. Combining the three animations and the legend together is accomplished using the GraphicsArray command in Mathematica.

Example III: Choosing coordinate systems or components The purpose of this notebook is to help the student to think about the relationship between the choice of coordinate systems (or components) and a particle’s path. In the notebook, four different two-dimensional paths may be shown relative to a fixed coordinate system: a straight
Figure 9. This figure shows the intrinsic triad of the unit normal $\mathbf{u}_N$, tangential $\mathbf{u}_T$, and bi-normal $\mathbf{u}_B$ vectors along a three-dimensional particle path.

line, a circular path of constant radius, a parabolic path associated with a projectile in a gravitational field without drag, and a sinusoidal path. In addition to considering a particle’s path when choosing an optimal coordinate system, other factors related to the problem statement or situation may also be important. For example, one may want to find the radius of curvature at a point along the path or the motion of a particle may be viewed relative to a specified fixed position.

The examples just mentioned show that there are a variety of path geometries and secondary properties of interest a student will encounter in the course. Therefore, this notebook considers the four different paths by plotting them and superimposing unit vectors on a particle travelling each path in time. One example from the notebook, the parabolic path, is presented in Figure 11.

There are actually three different sets of unit vectors for this two-dimensional problem: one set associated with a cartesian coordinate system aligned with $x$-$y$ axes, one set associated with a polar coordinate system, and one set associated with normal-tangential components. Additionally, a vector describing particle position and a bold line representing the current radius of curvature are also included and change with time. All of the graphical data are generated from initial functions of time that describe the various particle paths. The instructor might use this notebook to point out the differences between coordinate systems, which are dependent on a particle’s position, and normal-tangential components, which are dependent on a particle’s trajectory. Also, the relationship between the normal component and the radius of curvature is made apparent.

Example IV: A free-fall problem  This is the first of the homework style problems and it is based on a model of a free-fall ride found in many of today’s amusement parks.
Rotation with constant angular velocity
Extension with constant velocity

Combining extension and rotation

Vector, components, and total derivative

Figure 10. A single image from the animation showing the effect of the time derivative of a vector is shown by this figure. The top two images show rotation and extension, respectively. The bottom image shows the combination of these two types of motion. There is also a legend, included as a still image, at the bottom right-hand side.

These rides consist of an elevator that takes people from the ground level to the top of a tower before the elevator is made to slide down as if the lifting cables were suddenly cut. The tower in question is, more often than not, an open rectilinear elevator shaft. Hence, the path of the elevator can be viewed as a straight line segment that is traveled over and over again. This relatively familiar example may be used in a variety of ways to effectively teach about the relationships between position, velocity, and acceleration in a one-dimensional problem. This problem is fairly complex since it incorporates realistic ramping of the velocity to begin the ride, to stop at the top of the elevator, and to finally come to rest at the ride’s end. Based on the chosen motion, the elevator may even drop below the starting point slightly before coming to rest at the original position. There is no drag incorporated into the actual free-fall, though.
Figure 11. This figure shows a screen shot with a particle following a parabolic path. It also includes the current radius of curvature, position vector relative to the origin, and unit vectors of a cartesian coordinate system ($u_x, u_y$), a polar coordinate system ($u_r, u_\theta$), and normal-tangential components ($u_n, u_t$).

It is possible to use this notebook in a qualitative manner, where the user simply enters a variety of parameters to describe the system, e.g., the elevator height and steady-state velocity. After evaluating the notebook, a number of graphs may be produced showing position, arc length (or length travelled), velocity, and acceleration as shown in Figure 12. In addition to simply observing or commenting on the graphs, the instructor may also go through the actual derivations needed to generate the results by working through the sections and subsections of the notebook, the headings of which are shown in Figure 13. By following through the derivations in such an organized manner, the student may also be sufficiently interested to or required to work out a similar problem by using this notebook as a template. An animation of the motion of the elevator corresponding to the time series plots shown in this paper is also included on the web site.

Example V: Some projectile problems  Three variations of the projectile problem are presented with this example. In each case, the initial velocity and angle of a particle are entered by the user. The three cases represent a particle in a uniform gravitational field subjected to no drag, to drag as a linear function of speed, and to drag proportional to the

Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2003, American Society for Engineering Education
Figure 12. These images are from the free fall problem, showing position, position and arc length, velocity, and acceleration versus time as taken directly from Mathematica.

speed squared. Based on the user’s choices, the particle path as well as position, velocity, and acceleration of the particle as a function of time are all shown in an animation (see Figures 14 and 15).

The students may be asked to consider the relationships between position, velocity, and acceleration by inspecting the shapes of the plots. The effect of drag both on the distance of travel and on the shape of the path of travel may also be observed. A feature of this notebook is the use of If statements to produce final output based on the options selected by the user.

The student may be encouraged to think of everyday experiences in which an object may be modeled as a particle projectile (since this example considers only particles, phenomena like spinning are not taken into account). Figures 14 and 15 show two screen shots of the animation sequence for particles with identical initial velocities. In the first case there is no drag, while in the second case drag is considered to be a linear function of velocity. The drag reduces the horizontal distance travelled by approximately one-third when compared with the case with no drag. One may clearly see the change in the shape of each trajectory as well.
Free-Fall Ride

A one-dimensional particle problem using the relationships between position, velocity, and acceleration (also involving differences between position and arc length and velocity and speed)

Notebook Initialization

Problem Statement

Calculations Set-up

- Ride Parameters
- Initial Climb
  - Position
  - Velocity
  - Acceleration
  - Plots
- Pause Before Falling
- Free Falling
- Final Slow-Down Phase

Summary Plots

- Position
- Velocity
- Acceleration

Finding the Minimum Station of the Elevator

Expressing the Arc Length

Position vs. Arc Length Plots

Figure 13. This image shows a screen shot from Mathematica listing the outline of the solution method used for the free-fall problem.

Example VI: Three particles interacting via an energy potential  In the interest of exposing engineering students to a variety of applications in which the principles of dynamics are essential, a simple case modeling the interaction of three nickel atoms is presented here. Within solid state physics, atomic interactions are often modeled using potential energy functions. A common potential, and the one used in this example, is the Lennard-Jones potential. This potential is a two-particle potential in that it describes the energy between two atoms as a function of the separation distance between the atoms. By simply taking the derivative of the potential with respect to the separation distance,
one obtains a function for the force acting between the two atoms. This force is either attractive or repulsive. This derivation, as well as plots of the energy potential and inter-atomic force, are given within the Mathematica notebook. Students have the opportunity to see the development of the equations and to nurture a qualitative understanding of the meaning of the equations from the visual output.

Once the force has been written as a function of separation distance, a series of ordinary differential equations are presented and solved that model the behavior of three interacting particles. The final step is to apply initial conditions. At this point, an animation showing the particles moving in time is produced. In addition to the current position of each of the three particles, every animation frame shows the particle path up to the current time step. One image taken from this animation is shown in Figure 16. It is interesting to note that the center of mass of this system remains constant throughout the motion of the three atoms.
Figure 15. This is a screen shot showing one frame from the problem modeling a projectile with drag as a linear function of velocity. The particle follows its path in the topmost figure. The second row animates position, velocity, and acceleration in the x-direction, while the third row shows position, velocity, and acceleration in the y-direction.

Figure 16. This view shows three particles subjected to a Lennard-Jones potential in two dimensions. The particle paths are traced behind each particle.
Conclusion

Due to the many options now available to instructors in engineering courses and to the diversity of experiences, strengths, and weaknesses of instructors and students alike, there are many approaches available for teaching traditional mechanics courses like dynamics. The use of mathematical computer packages, like Mathematica, provides a great opportunity for instructors to develop their own course materials to supplement their traditional teaching methods and tools and, hopefully, to better reach more students. There are many questions the lecturer must first answer before beginning the process of creating a Mathematica notebook; these questions relate to the scope of the project, the place of the notebook in the context of the course, who will be the actual notebook user, and where the notebook will ultimately be used.

Although this is clearly not a manual on the use of Mathematica, some ideas and suggestions were presented regarding notebook creation. A number of examples were included to represent the types of notebooks that may be created for an undergraduate course in dynamics. Some of these examples focused more on theoretical and abstract ideas that are important in a student’s understanding of dynamics. Other examples showed what could be done with more traditional problems by not only solving the problems, but showing additional, more fundamental, results as well. These examples should provide some ideas for instructors interested in using Mathematica or comparable software packages in their courses, whether to create occasional, smaller demonstrations or to utilize such notebooks as a regular feature of the course.

Acknowledgments

The authors would like to acknowledge the support provided by the National Science Foundation through the CCLI-EMD grant DUE-0127511. In addition, Francesco Costanzo wishes to acknowledge the support provided by the National Science Foundation CAREER program through grant No. CMS-9733653.

References


Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition
Copyright © 2003, American Society for Engineering Education


Biographies

Charles L. Randow is a graduate assistant and Ph.D. candidate in the Department of Engineering Science and Mechanics at Penn State. He received his M.S. from Virginia Tech in Engineering Mechanics. His current research is focused on the mechanics of nanostructures.

Andrew J. Miller is a graduate assistant and Ph.D. candidate in the Department of Engineering Science and Mechanics at Penn State. He earned a M.S. degree from Penn State in 1999. His research interests include mechanics of nanostructures, dynamics of mechanical systems, and parallel computing.

Francesco Costanzo came to Penn State in 1995 and is an Associate Professor of Engineering Science and Mechanics. He earned a Ph.D. degree in Aerospace Engineering from the Texas A&M University in 1993. His research interests include the mechanics of nanostructures and the dynamic crack propagation in thermoelastic materials.

Gary L. Gray came to Penn State in 1994 and is an Associate Professor of Engineering Science and Mechanics. He earned a Ph.D. degree in Engineering Mechanics from the University of Wisconsin–Madison in 1993. His research interests include the mechanics of nanostructures, dynamics of mechanical systems, and the application of dynamical systems theory.