GENERATING AND MODELING DATA FROM REAL-WORLD EXPERIMENTS

Robert E. Kowalczyk and Adam O. Hausknecht
University of Massachusetts Dartmouth
Mathematics Department, 285 Old Westport Road, N. Dartmouth, MA 02747-2300
rkowalczyk@umassd.edu and ahausknecht@umassd.edu

We have developed the new software package *Image Data Modeler* that has the capability of importing a digital image of some physical phenomenon and using it to generate a set of data for modeling purposes. This software runs on computers with Macintosh OS X, Linux, or current versions of Microsoft Windows/NT/XP operating systems that have a Java J2SE 1.4.2 (or later) installed. This package is a prototype of a new cross-platform version of our Mac-only software package TEMATH. The goal of the software is to enhance the applied modeling aspect of our mathematics courses. Mathematics textbooks contain a multitude of applied examples and exercises for use in class and for student homework. Instructors diligently present a variety of applied examples to their classes and students solve their share of applied exercises for homework, however, students seldom experience the true nature and power of these applied problems. They often do not get an opportunity to develop a good intuition of the applied aspect of the problem nor do they get to appreciate the modeling power of the underlying mathematics. Our software is designed to bring applied problems to life in the mathematics classroom and to have students truly experience the applicability of mathematics. The instructor or student can use a digital camera to take a picture of some physical object, for example, a seashell or the capillary action of water between two glass plates; or they can take a video clip of a moving object or system, such as, the flight of a ball or the periodic motion of a spring-mass system. The resulting digital picture or the frames of the video can then be imported into our software for data analysis and modeling purposes. The tools included in the software can be used to measure, record and save data points from the digital image or the frames of the video. Students or the instructor can use regression (least squares) models or the theoretical models found in the textbook to fit the data.

The *Image Data Modeler* can be used in a demonstration mode by the instructor to provide a visual learning experience for their students, or students can use the software to test their own models. In either case, students become actively involved in these experiments and they provide an escape from the traditional all-lecture format. Classroom experiments of this type can be used to verify the accuracy of the theoretical models found in mathematics textbooks and to help build confidence in our students about the use of these models as problem-solving tools in the real world. Students develop new investigation and problem-solving skills and their physical and mathematical intuition.

Our software is free and can be downloaded from our web site (see below) by any potential user — student or instructor. Our goal is to make it easier for teachers to do real applied mathematics in the classroom and to make mathematics more enjoyable for the first- and second-year mathematics students. Figure 1 shows the four windows of the *Image Data Modeler* along with a model for the trajectory of a ball. Features of our software include the following:

- **The Graph window**: Used for displaying a digital image or a sequence of digital images extracted from a video. You can view the images as an animation or step through them frame-by-frame. Currently two tools are available in this window: the *Point* tool (used to mark and record the position of an object within the digital image), and the

Line tool (used to measure distances and slopes). The recorded data can be plotted along with a theoretical model or a least squares fit (linear, quadratic, cubic, exponential, natural exponential, natural logarithmic, power).

- **The Work window**: Used to store data sets and models. Currently, the following models are available: linear, quadratic, cubic, exponential, natural exponential, natural logarithmic, power, sine, logistic, and hyperbolic cosine.
- **The Data Table window**: Used to enter, edit, transform or generate data. Available operations on the data include: average, sum, max/min, correlation, standard deviation, forward differences, Riemann sums, and others.
- **The Report window**: Used to record the results of the Graph window's Point and Line tools. You can also enter text into this window to document the model being studied.
- **The Domain & Range window**: Used to enter the values for the plotting x-domain and y-range. It also tracks the coordinates of the position of the cursor in the Graph window and can be used to precisely enter points and endpoints of line segments.

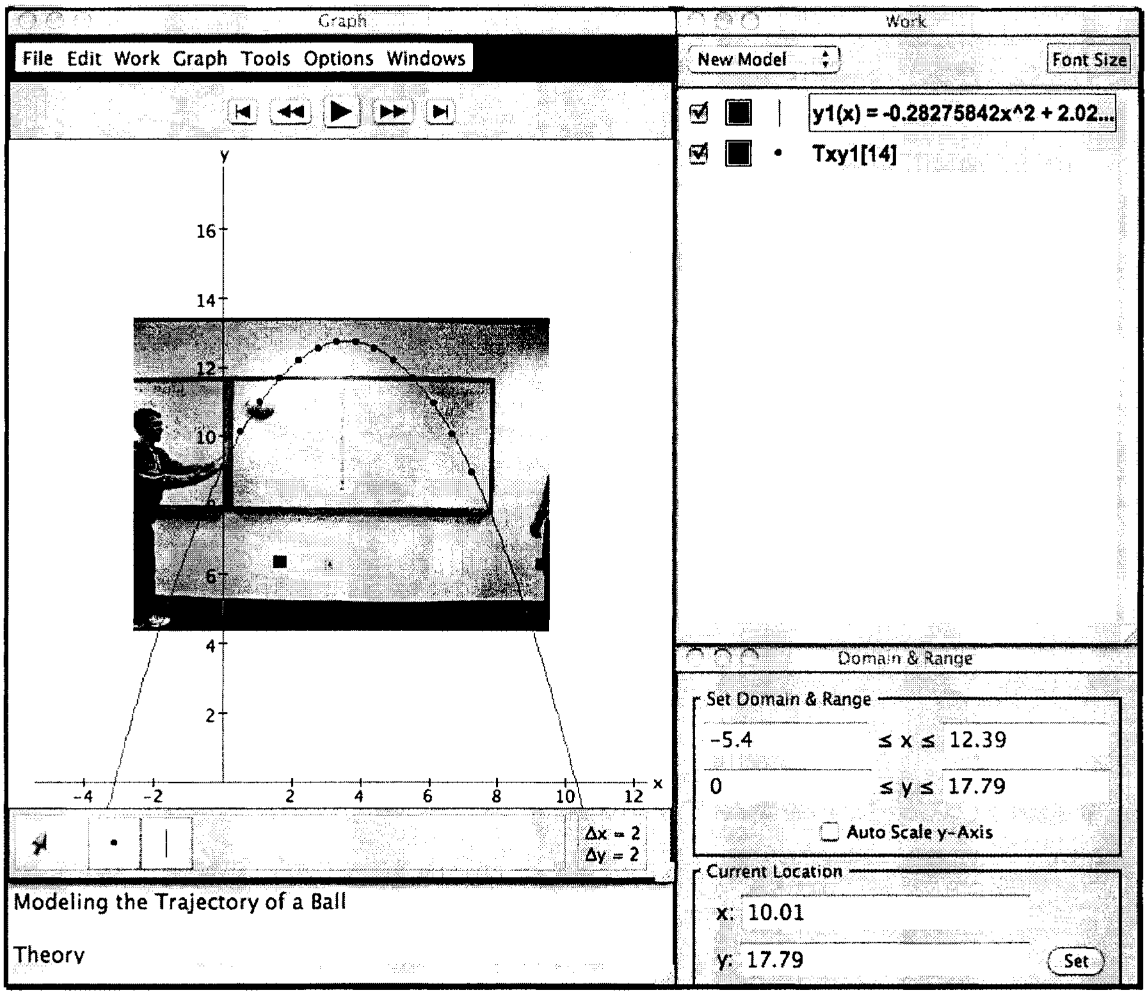


Figure 1 The Graph, Work, Report, and Domain & Range Window

Modeling the Trajectory of a Ball

Using a digital camera, we took a video clip of two of our calculus students tossing a ball. The video was then imported into a computer and converted to a sequence of frames. Our software has the capability of playing the sequence of images as a movie or stepping through the images one at a time. We placed a yardstick in the background of the video so that we could calibrate the axes for measuring the position of the ball in units of feet. As we stepped through the images, we used the Point Tool to mark the position of the ball in each frame of the video. These positions were saved into a data table. The first step in the analysis was to fit the data with a least squares quadratic fit. The fit was excellent and convinced the students that the trajectory is really a parabola (see Figure 2). The second part of the analysis was to convince students that the theoretical model they learned in their physics class is an accurate model[1]. This model for the trajectory is

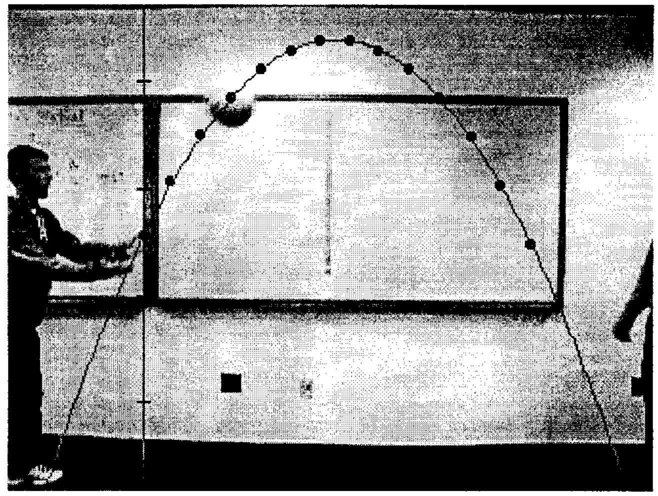


Figure 2 Modeling the Trajectory of a Ball

$$y = x_0 + \tan(\theta_0)x - \frac{gx^2}{2(v_0\cos(\theta_0))^2}$$

where θ_0 = initial angle, v_0 = initial velocity, x_0 = initial *x*-position, and g = acceleration due to gravity.

Using the Line Tool, we drew a line tangent to the path of the ball at the initial position. The slope of this line was 2.027. This value equals $\tan(\theta_0)$. We next measured the distance between the first two positions of the ball and divided it by 1/15 sec. (this is the time between frames of the video) to obtain an estimate of the initial velocity v_0. The *x*-coordinate of the first position was used to estimate x_0. These values were entered into the theoretical model and the trajectory $y = 9.14 + 2.027x - 0.228x^2$ was obtained. This model was nearly identical to the least squares fit and it was an excellent fit to the data.

Finding Volumes Using the Disk Method and Riemann Sums

In this experiment, we wanted to use the *Image Data Modeler* to calculate the volume of an object with a known volume in order to build confidence in our students about the underlying mathematical process. We took a digital picture of a ball and imported it into our software. To obtain accurate volumes, we carefully calibrated the axes in the Graph window to be in units of feet and centered the ball at the origin. Using the Point tool, we placed a sequence of points along the top circumference of the ball (see Figure 3). We next saved the points into a data table. The integral for finding the volume of revolution using the disk method and the Riemann sum approximation for the integral are given by[2]

$Volume = \pi\int[f(x)]^2dx \approx \pi\sum_{i-1}^{n}[f(x_i)]^2\Delta x_i$. Using the Transform command in the Data Table window, we squared the *y*-coordinates of the saved data points and then multiplied them by π. Finally, we used the Left and Right Riemann Sums commands to obtain estimates of the volume of the ball. The best estimate for the volume was 0.267 cubic feet. In order to determine how close this estimate was to the exact volume of the ball, we used a cloth tape measure to measure the exact circumference of the ball and then calculated the radius and volume of the ball. The calculated volume was 0.264 cubic feet. Thus, our estimate was off by only 1.1%. Building confidence in our students, we next estimated the unknown volume of a gourd (see Figure 4). Using the same technique as on the ball, we found the volume to be 7.75 cubic inches.

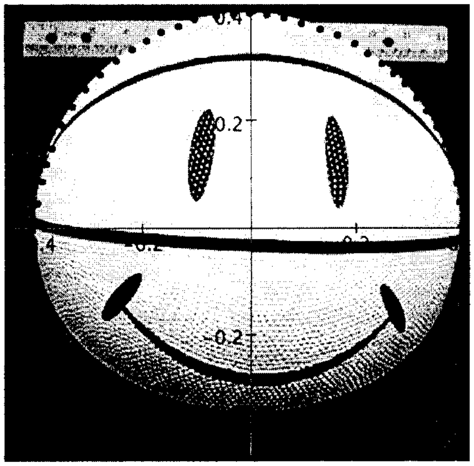


Figure 3 Estimating Volumes

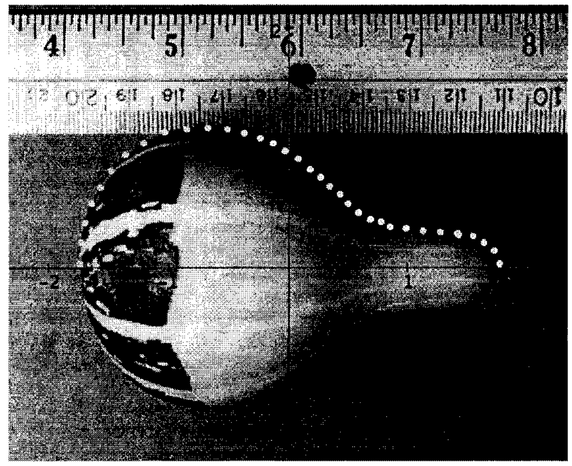


Figure 4 Approximating Unknown Volumes

Exporting Data to Microsoft Excel

Some of our engineering calculus students were studying drag in their engineering course so we decided to perform an experiment that included a drag component. Being near Halloween, we purchased a lightweight paper pumpkin. We took a video of one of our students dropping the pumpkin. Using our software, we stepped through the frames of the video and marked the position of the pumpkin in each frame. To determine if the pumpkin reached a terminal velocity, we fit the data with the quadratic model $y = -gt^2/2$. This model fit the data near the beginning of the fall, but towards the end of the fall it was obvious that the pumpkin had reached a terminal velocity. Using the Line Tool, we measured the slope of the line passing through the last three data points. This gave us the estimate of 11.09 ft/s for the terminal velocity (the axes were calibrated to units of feet). See Figure 5. Since we have not yet implemented a parser-evaluator in our software to evaluate general mathematical functions, we exported the data to Excel and plotted the data along with the theoretical model

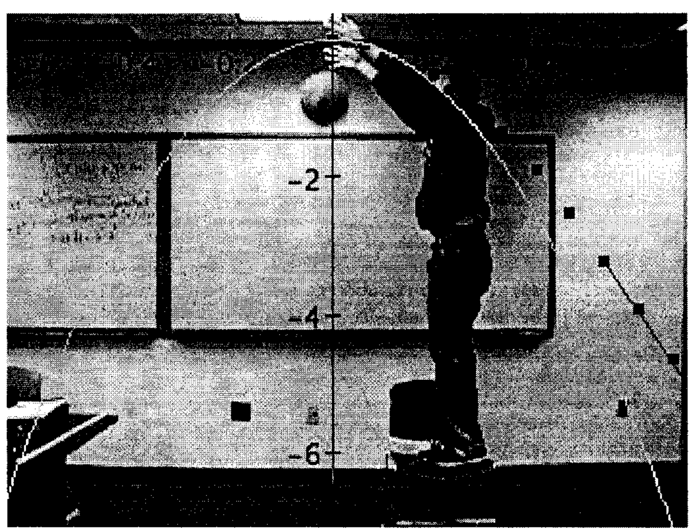
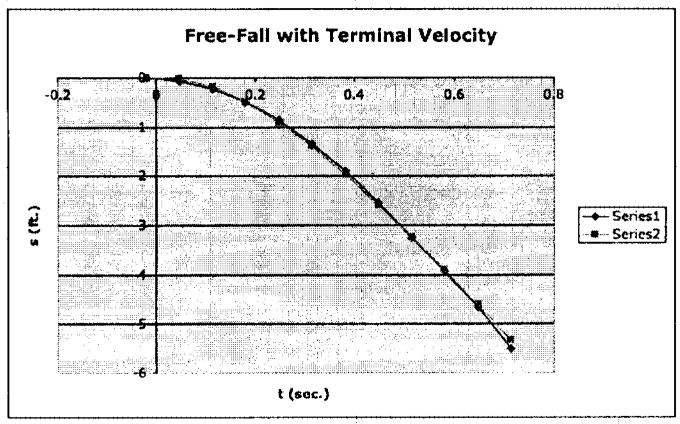


Figure 5 Modeling Free-Fall With Drag

$$s(t) = -\left(v_{term} t + \ln\left(\frac{1 + e^{-2gt/v_{term}}}{2} \right)\left(\frac{v_{term}^2}{g} \right) \right),$$

where v_{term} = measured terminal velocity and g = acceleration due to gravity. Note how well the model fits the data (see Figure 6). This experiment enabled us to integrate calculus with engineering and to demonstrate to our students the power of these mathematical models.

	t	Measured Data	Theoretical Data	
to =	-0.02		11.09	= Vt
	-0.02	-0.0105	-0.0064364	
	0.0467	-0.073684	-0.0349604	
	0.1133	-0.231579	-0.2031913	
	0.18	-0.515789	-0.4995439	
	0.2467	-0.863158	-0.9058647	
	0.3134	-1.3368	-1.401322	
	0.38	-1.905263	-1.9657296	
	0.4467	-2.536842	-2.5815633	
	0.5134	-3.231579	-3.234717	
	0.58	-3.926316	-3.914418	
	0.6467	-4.652632	-4.6127358	
	0.7134	-5.505263	-5.3239784	



Theoretical Solution $s(t) = -\left(v_{term} t + \ln\left(\frac{1 + e^{-2gt/v_{term}}}{2} \right)\left(\frac{v_{term}^2}{g} \right) \right)$

Figure 6 Data Exported to Excel for Modeling

Future Plans

We are the authors of TEMATH, a Mac-only application, and have been asked numerous times for a PC/Windows version. This is our first attempt at converting TEMATH to a cross-platform application. The complete conversion is probably a few years away. However, we will add additional tools and features incrementally and would appreciate feedback from users as we move toward the goal of a full conversion. The current version of the *Image Data Modeler* will be available at our TEMATH web site

www2.umassd.edu/TEMATH

You can also download a copy of TEMATH and its documentation, application files, and games from this web site.

Bibliography

[1] D. Halliday, R. Resnick, J, Walker, Fundamentals of Physics, John Wiley & Sons, Inc., 1997.

[2] R. Larson, R. Hostetler, B. Edwards, Calculus, Houghton Mifflin Co., 2003.

[3] Robert Kowalczyk and Adam Hausknecht, *TEMATH - Tools for Exploring Mathematics Version 2.1.1*, 2003.