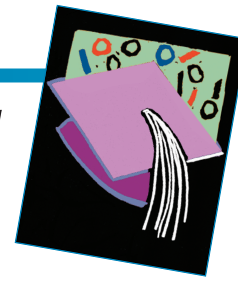
Editors: Michael Dennin, mdennin@uci.edu
Steven F. Barrett, steveb@uwyo.edu



POSTPROCESSING IN AUTOMATED GRADING SYSTEMS, PART 3

By Peter A. Knipp and S. Raj Chaudhury

Postprocessing provides crucial flexibility when instructors use automated systems to assess student performance in science laboratory courses.

This final segment in our three-part series on postprocessing in automated grading systems demonstrates the method's usefulness for laboratory assignments, which routinely involve complications such as large numbers of measurements, the presence of experimental uncertainty, or many steps of analysis. Each of these complications usually forces instructors either to devote inordinate human resources to grading tasks or to base students' grades upon their participation rather than their performance.

In part 1 of our series,[1] we reviewed the standard algorithm for automated homework grading and introduced how postprocessing uses a programming environment to go beyond that algorithm to extend the automated system's power. We also presented the specific variables and functions used in WebAssign's postprocessing implementation (www.webassign.net/info). In part 2,[2] we showed postprocessing's effectiveness in math and science lecture courses, especially upper-level ones. In this last segment, we'll discuss some unique circumstances that arise when assessing student performance in laboratory classes and show how postprocessing addresses those challenges.

Equipment Randomization

Randomization is an important feature of online homework systems, because it prevents students from copying other students' correct submissions. The randomization usually occurs in the software itself, originating from something unique, such as the student's username. Randomization offers a similar benefit in lab courses, but it offers additional benefits as well. For example, school staff members often have difficulty maintaining multiple identical copies of experimental equipment for students. As a result, distributing dissimilar apparatus to students creates a type of randomization that was difficult to grade before the inception of automated grading systems. Because this randomization occurs outside of the software, students themselves must pass the information to the grading system. In Figure 1, we show how a two-part question addresses this need in a straightforward way. The student submits the randomized information in the first question, and the second question postprocesses it. (The source code for this and other examples we refer to here is available at www.pcs.cnu.edu/~pknipp/cise/supplemental, along with the figures from Parts 1 and 2 of this series. A live version of this "WebAssignment" is available within the WebAssign database as Assignment ID 591552; WebAssign access if free to all instructors at accredited educational institutions. See www.webassign.net for details.)

This strategy blends the topologies that we diagrammed in part 2 of our series (figures 4 and 5).[2] We handle the first question's grading in at least two different ways, depending on circumstances. For example, if we provide each student with a length of string to construct a simple pendulum, and if the string lengths L vary from 1.2 to 1.3 meters, then the computer marks the first question—"What is the length of your string?"—as "correct" if 1.2 m and 1.3 m bracket the student's submission. If the equipment's physical characteristics vary much more widely (as in spring constants, for example), we

- label each item with an integer, or *code number;*
- create a lookup table in the software to determine the item's characteristic; and
- require the student to specify the item's code number in the first question.

Figure 1 shows both approaches.

Nonunique Measurements

Often, students can't control the phenomena that they encounter in laboratories. For example, a student using an airtrack to collide two gliders probably can't control their initial speeds with much accuracy. Even when students can easily control an experiment, instructors often want to give them freedom in how they exercise that control.

 1521-9615/09/$26.00 © 2009 IEEE

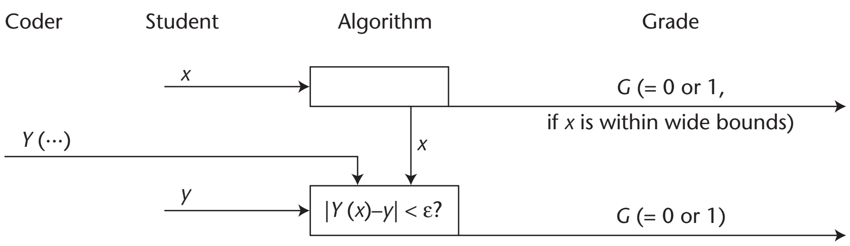


Figure 1. Schematic diagram of randomization approach for laboratory course equipment. The question is divided into two parts: students first submit the (randomized) equipment's specification(s) (x), and then submit the main experimental result(s) (y). The grade (G) for the second part is nonzero only if y is sufficiently close to a key whose value is based on the value of x.

For example, when students study a block sliding down an inclined plane, we might let them choose the specific angle of the plane's incline. In this case, we want to correlate our grading of a student's separate measurements (such as the plane's slope and the block's acceleration), rather than to grade these factors independently. Obviously, the possible measurement combinations aren't unique. In previous parts of our series, we discussed how to use postprocessing to grade a single question whose answers are nonunique,[1] and how to grade a series of questions in a correlated manner,[2] either simultaneously or sequentially.

Here, we let students choose one or more independent variables (such as the plane's slope), and ask them to measure the resulting dependent variable(s) (such as the block's acceleration). They must then submit this array of data to the online system (typically in multiple answer boxes). Later in this installment, we discuss how to assign credit so that it rewards careful experimental technique in such a laboratory setting.

Handling Experimental Data Sets

The measurements for an experiment generally involve more than one datum. Even for a given experiment, we don't always have advance knowledge of data quantity; in part 2, we discussed how to postprocess submissions of unspecified amounts of data.

If a student's data are represented most naturally as a one-dimensional horizontal array with an unknown number of elements, then WebAssign's "fill-in-the-blank" question type is probably the most logical choice. If, however, the data are considered more naturally as either vertical or two-dimensional, then WebAssign's "essay" question-type is probably the best choice. A student might generate sets of two or more columns—such as lists of (x,y)—when exporting data from a spreadsheet or from a computer-aided data-acquisition system, such as Pasco's DataStudio or Vernier's Logger Pro. The student could then submit the data to a WebAssign essay box using a simple copy-and-paste procedure. In a circuits experiment at CNU, each of our students routinely submits sets of several hundred (x,y) pairs to WebAssign in this manner. Part 2 explained two simple approaches for grading these submissions in homework assignments. As we now describe, we usually grade such submissions in a more sophisticated way in lab courses.

Data Quality

Perhaps the biggest algorithmic challenge for an automated grading system in a laboratory setting is the presence of experimental uncertainty. It's often difficult for instructors to estimate in advance the extent to which measurement accuracy will fluctuate from one experiment or student to the next. Given this, we use a continuous function to generate the student grade G $(0 \le G \le 1)$ based on the data's "quality."[2]

Previous authors[3] have utilized such a continuous grading strategy for "challenge laboratories" in which students use their analysis of a series of measurements to generate predictions that they subsequently test. Here, we consider the fact that a student commonly measures N data points as ordered pairs that are "fit" by a model having one or more parameters (a, b, and so on). The standard way to perform such a fit minimizes the value of χ^2, defined as[4]

$$\chi^2(a,b,\ldots) \equiv \sum_{j=1}^{N} \frac{[y_j - f(x_j; a,b,\ldots)]^2}{\sigma_j^2},$$

where σ_j is the experimental uncertainty of the j–th data point, and we derive the function $f(x; a, b, \ldots)$ from a particular theoretical or empirical model. The student (often aided by commercial software) minimizes the value of χ^2 by setting to zero its first partial derivatives with respect to each of the fitting parameters (a, b, …). For reasonable experimental results, this minimized value of χ^2 is on the order of N. In some circumstances (such as the linear fit of position versus time data for a glider's motion on a level, frictionless airtrack) the fitting parameters don't have any a priori values, but in others (such as the linear fit of velocity versus time data for a freefall experiment) they do. If the parameters have no a priori values, then our choice for the grade becomes:

$$G = \frac{1}{1 + Z(\chi^2)_{\min} / N}$$

where $Z = O(\langle G\rangle - 1^{-1})$, and $\langle G\rangle$ is the instructor's target value for the average student grade G. If the parameters do have a priori values,

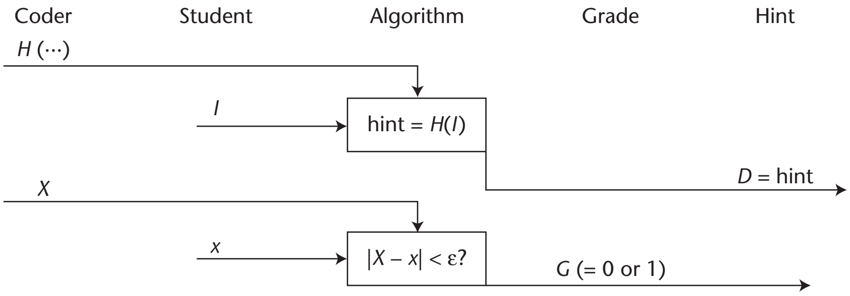


Figure 2. Flow chart of a simulated-experiment problem type. In the question's first part, the student repeatedly adjusts and submits the experiment's independent variable(s) *I*, to which the computer responds each time with the simulated dependent variable(s) *D*. In the second part, the student eventually submits for grading the value(s) *x* for the system's physical characteristic(s).

then we choose *G* to equal either $1/[1 + Z\chi^2(a,b, \ldots)]$ or $1/[1 + Z(\chi^2)_{\min} + A|a - a_{\text{fit}}| + B|b - b_{\text{fit}}| + \ldots]$. For the latter option, we choose the relative sizes of *Z*, *A*, *B*, …, to emphasize the relative importance of the student's obtaining data for which the fit values a_{fit}, b_{fit}, … are numerically close to the a priori values.

Data Analysis

Many laboratory assignments include both the acquisition and analysis of raw data. We grade the raw data using methods we just described; for the student's analysis, we base the key on the raw data he or she submitted. Hence, the raw data are important inputs for grading the student's analysis. It makes sense to base students' analysis grades solely on their calculations, rather than also considering their raw data's quality.

In Part 2, Figure 5 showed how postprocessing lets the system use the student's answer to earlier parts (in this case, the raw data) to establish the key for later parts (here, the analysis). Hence, our automated algorithm for grading the analysis without being prejudiced by the raw data is analogous to awarding partial credit in a sequential homework question when the student makes a mistake in an early part of the question. This lets us easily loosen the software's error-checking tolerance for the raw data, while retaining a tight tolerance for the analysis. Because of error propagation, retaining a tight tolerance for analysis is practical only if we use postprocessing to employ the student's earlier submissions to calculate the keys for those later parts.

Simulating a Physical System

The widespread use of simulations designed in environments such as Interactive Physics (www.design-simulation.com) or via standalone software such as Physlets (webphysics.davidson.edu/Applets) or PhET simulations (phet.colorado.edu) now lets students conduct challenging simulations rather than actual experiments. Using WebAssign's computational engine and display capability, we can create these simulations.

We simulate a physical system that has independent variable(s) *I* and dependent variable(s) *D*. As Figure 2 shows, the purpose of such a simulation is to challenge the student to determine some physical characteristic(s) *X* of a system, after the student has performed a systematic series of simulated measurements. Each measurement consists of the student's choosing the value(s) for *I* and then reading the corresponding value(s) for *D*. WebAssign has the ability to assign arbitrarily rich feedback to the intuitively named variable `$HINT`,[1] and we use this "hint" to communicate *D*'s value to students. When using WebAssign in this way, the values of the flags `$MARKOFF` and `$HINT ON CORRECT` should equal 1 (that is, "true") for the question's first part, for which we set the `points` to zero within the `Assignment Editor`.

After completing enough virtual experiments—that is, after submitting the question's first part enough times, each with (a) different value(s) for *I*—the student should be able to analyze the data and determine the value(s) of the system's physical characteristic(s).

An example of such an exercise is to simulate the AC current-voltage characteristics of a circuit with a resistor combined either in series or in parallel with an inductor or a capacitor. By monitoring the circuit's impedance (the sole dependent variable) as a function of the voltage source's sinusoidal driving frequency *f* (the sole independent variable) for at least two values for *f*, students should be able to characterize fully the circuit's nature, including its topology, its resistance (*R*), and its nonresistive circuit element's identity.[5] Figure 3 shows a screenshot of this example.

In our introductory physics lab courses for science and engineering majors, we've used WebAssign to automatically assess approximately 80 percent of our lab activities. In most cases, this automation hasn't required any change in the lab activity itself. Students still work in teams of two or three in a 165-minute lab session. The automation has substantially increased grading precision, letting us detect even the most subtle student mistakes (such as their implementing strategies that are susceptible to round-off-error accumulation). Such automation also allows for randomization of equipment and/or experiments among students,

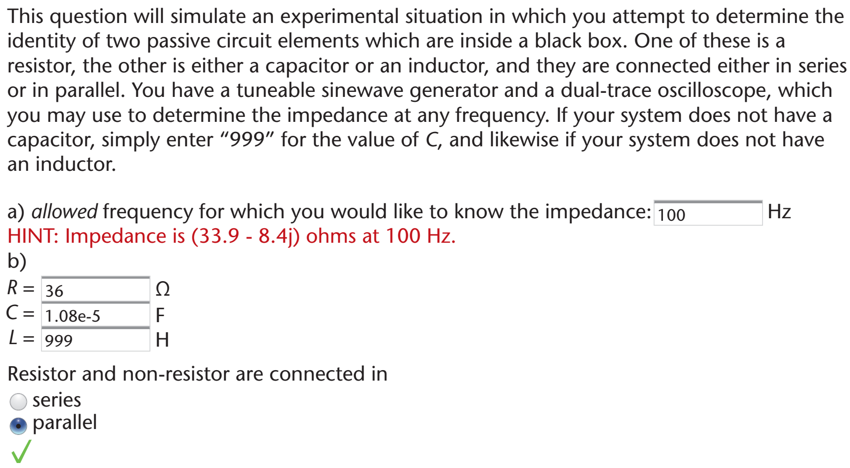


Figure 3. Screenshot from WebAssign of a simulated-experiment problem. This example involves the AC current-voltage characteristics of an unknown combination of a resistor and another circuit element.

which improves the quality of student collaboration—something we encourage in the laboratory.

Students appreciate the instant feedback that they receive, especially when it pertains to their raw measurements. Further, students typically stick with each part of the lab activity until they get it right. The result? Students work more slowly than in previous years, but their final products are of a much higher quality.

References

1. P.A. Knipp and S. Raj Chaudhury, "Post-processing in Automated Grading Systems, Part 1," *Computing in Science and Eng.*, vol. 11, no. 3, 2009, pp. 64–67.
2. P.A. Knipp and S. Raj Chaudhury, "Post-processing in Automated Grading Systems, Part 2," *Computing in Science and Eng.*, vol. 11, no. 4, 2009, pp. 82–85.
3. A.J. Greer and J. D. Bierman, "Challenge Laboratories," *Phys. Teach.*, vol. 43, no. 8, 2005, pp. 527-529.
4. P.R. Bevington and D.K. Robinson, *Data Reduction and Error Analysis,* 3rd ed., McGraw-Hill, 2003.
5. J.W. Nilsson and S.A. Riedel, *Electric Circuits,* 8th ed., Pearson Prentice Hall, 2008.

Peter A Knipp is an associate professor of physics at Christopher Newport University. His research interests include surface physics, semiconductor nanostructures, and learning technologies. Knipp has a PhD in physics from University of Chicago. Contact him at pknipp@pcs.cnu.edu.

S. Raj Chaudhury is the associate director of the Biggio Center for the Enhancement of Teaching and Learning at Auburn University. His research interests include learning technologies, science education, and faculty professional development. Chaudhury has a PhD in physics from the University of California, Los Angeles. Contact him at schaudhury@auburn.edu.