



POSTPROCESSING IN AUTOMATED GRADING SYSTEMS, PART 2

By Peter A. Knipp and S. Raj Chaudhury

In this second installment on postprocessing, the authors discuss a few simple grading algorithms for handling a wide array of homework and test questions that are not otherwise handled well by the standard algorithm.

In this second installment of our three-part series on postprocessing in automated grading systems, we demonstrate the usefulness of our approach for homework and tests, particularly for those in upper-level courses in science and engineering. Postprocessing uses a full programming environment to simulate a real (human) grader's actions. In the first installment,¹ we reviewed the standard algorithm used in automated assignment grading—namely, to determine if the student's submission is within a particular percentage of the key—and we introduced the manner in which postprocessing extends beyond that algorithm to increase a grading system's versatility. We also presented the specific variables and functions used for postprocessing in WebAssign (www.webassign.net/info). In this installment, we'll focus on the uses of postprocessing in lecture courses.

Complicated Student Submissions

Off-the-shelf question formats for grading systems frequently can't handle the complexity of a given homework problem (for example, "Create the position x versus time t graph for a dog that runs from $x = 2$ m to 10 m with a constant speed of 4 m/s and then sits for 3 s"). In these cases, the student must construct an appropriate response—for example, by submit-

ting a small quantity of numbers, as in Figure 1—which is computer gradable (www.pcs.cnu.edu/~pknipp/cise/supplemental). Consequently, the student must visualize his or her response prior to submitting it. We, as instructors, have addressed both these needs by using a two-part question (see Figure 2). In the first part, the student repeatedly resubmits to finalize his or her response, which echoes back to the student after each submission as perhaps an elaborate text string or a graphically rich result. In the second part, the student signals that he or she has finalized the submission. Postprocessing serves a dual purpose for this type of question because it creates an image of the student's response prior to submission, and it grades the submission.

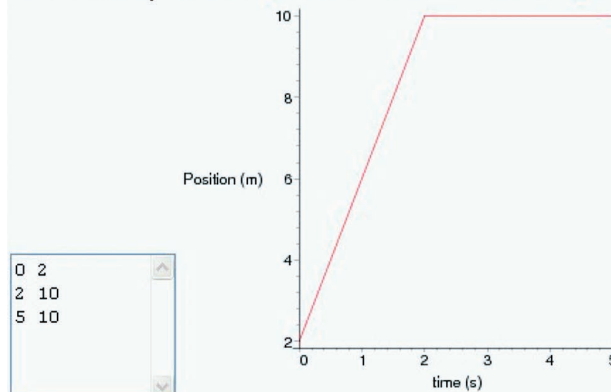
Grading Multiple Parts Simultaneously

Although most assignments involve independent parts, there exist at least two circumstances in which instructors like to grade two or more question parts simultaneously. When using WebAssign in each of these circumstances, we handle the first parts of the question in a different manner from the last part, as follows: we set the value of \$MARKOFF to equal 1 in the Question Editor to remove the discrete correctness indicator—that is, a red \times or a green \checkmark —that the stu-

dent would otherwise see. We also set the value of Points to equal zero in the Assignment Editor. The system should then manipulate the value of \$THIS_SCORE in the last part of the question in order to assign an appropriate grade to the totality of the question parts.

In one circumstance, an instructor might ask a student to specify all the quantum numbers (n, l, m_l) for a particular state of the hydrogen atom. Usually, a homework system displays this question as a set of three independent answer boxes. Unfortunately, this enables the student to work on one answer box at a time. Because the quantum numbers are integers that are usually small, a student could easily guess his or her way to the correct solution with repeated attempts. (Instructors usually regard the option of multiple submissions as one of the most beneficial settings in online grading systems.) To discourage such a mindless problem-solving strategy in this case, the instructor grants the student nonzero credit only if all parts submitted are correct (see Figure 3). If the probability that a student answers the j th part correctly ($1 \leq j \leq N$) is P_j in an N -part question, then the question's mean grade is $\sum_j P_j$, if all parts are graded independently—this expectation value decreases to $N \prod_j P_j$ if the system uses an *all-or-none* grading strategy. If instructors want to

Enter (t,x) coordinate pairs in the box below in order to create a graph of a dog that runs from $x = 2$ m to 10 m with a constant speed of 4 m/s and then sits for 3 s.



Click below when you are content with your graph.



provide a bit more feedback to the student regarding errors made in a multi-part question that uses the all-or-none strategy, they can use a hint¹ to inform the student as to the number of question parts they answered incorrectly.

In a second circumstance that calls for simultaneous grading, instructors ask questions that have nonunique solutions and require multiple answer boxes. Consider this example: “Specify the x - and y -components of a vector whose magnitude equals 13 .” Figure 4 shows how instructors can use postprocessing in this circumstance in which the function F is $|x^2 + y^2 - 13^2|$ for this specific example. The computer never grades the student’s choice of the x -component by itself. As long as the student submits a corresponding y -component that gives the final desired result, the computer grades the submission as correct. Note that the scenario presented in the preceding paragraph is a special case, with $F(x, y) = |X - x| + |Y - y|$.

Accepting Unspecified Amounts of Data

Often, we want students to enter more than one number in a single answer box—for instance, we might not want to alert the student to the number of necessary data (“Specify all prime numbers between 10 and 20 .”). We use two approaches for grading such submissions, and they differ in terms of awarding partial credit. The lenient approach awards a fractional grade proportional to the number of correct quantities. The strict approach (analogous to the type of postprocessing diagrammed in Figure 3) gives nonzero credit only if all data are correct. The instructor can include a hint¹ in conjunction with the strict approach to indicate the fraction of incorrect data the student submits.

Figure 1. Screenshot of a WebAssign question showing a graph created from student input. The student enters numbers in the left-hand “essay box,” and the automated system uses these numbers to generate the graph. At the student’s request, the system subsequently grades this submission.

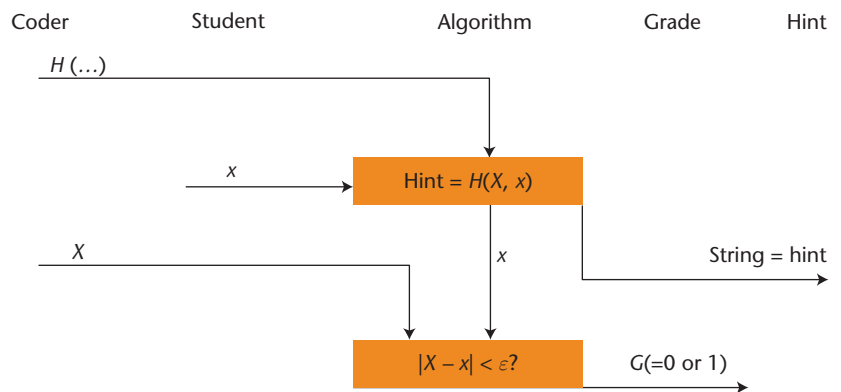


Figure 2. Flow chart for a hint to show the student how to format the submission. In the first part of this question, the system evaluates the function H to translate the student’s primitive submission x into a comparatively richer form that’s echoed back as a hint. In the second part (the only part that receives a score), the student simply indicates that the submission is ready for grading.

Data Quality

For most questions that students encounter in an introductory course in science or engineering, the instructor’s goal is for each student to obtain an answer x very close to the key X (for example, by including three or

more digits displayed on the calculator). Accordingly, the homework system awards a grade G of 1 if $|X - x|$ is arbitrarily small or a grade of 0 otherwise. In some circumstances, this grading method sets needlessly rigid expectations for students who strive

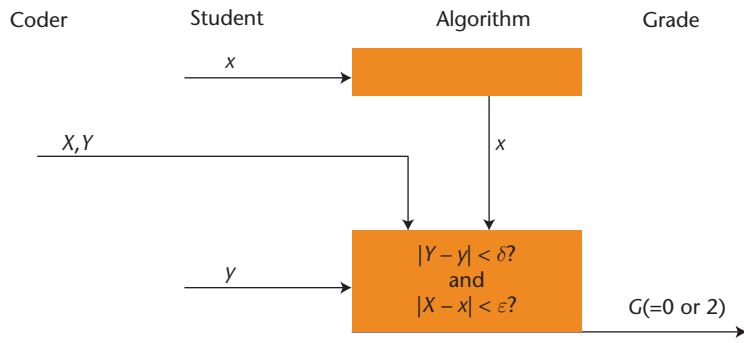


Figure 3. Flow chart for a problem type that discourages guessing. The upper box simply captures the student’s submission x and passes it to the bottom box; the latter generates a nonzero grade only if all the student’s submissions (x, y, \dots) are sufficiently close to the corresponding keys (X, Y, \dots).

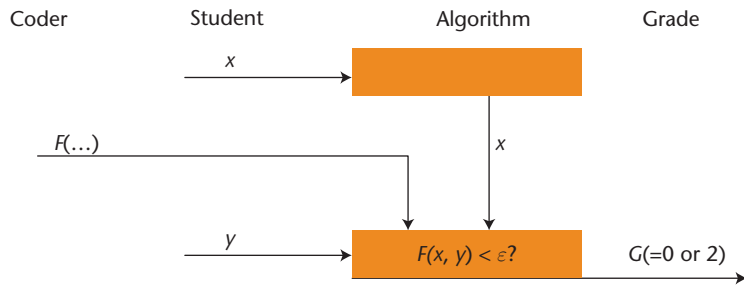


Figure 4. Flow chart for a question involving nonunique solutions x and y . The upper box simply captures the student’s submission x , and the bottom box generates a nonzero grade only if the student’s submissions x and y satisfy the question’s statement, as manifested by the fact that a given function F is sufficiently small if x and y are its inputs.

to achieve a highly accurate value for x —for instance, x might involve a definite integral that the student calculates numerically. In this case, we feel it’s appropriate to use a continuous function to generate G ($0 \leq G \leq 1$), such as $G = (1 + |X - x|)^{-1}$. When using WebAssign for this, the value of \$THIS_SCORE should be set to 1, and the system should pass G ’s value from the answer field of the Question Editor to the Conditional Points area of the Assignment Editor, where it should multiply \$POINTS (the part’s weight) to produce the grade for that part. Additionally, it makes pedagogical sense to set \$MARKOFF to 1. Note that this grading scheme enables a student to optimize mindlessly his or her grade

through the use of repeated submissions and standard algorithms for finding the maximum values of functions. Therefore, take care when setting an assignment’s administrative characteristics that include this question—that is, by rationing the quantity of each student’s submissions.

Awarding Appropriate Credit for Sequential Questions

A sequential question consists of N steps configured so that the output of at least one intermediate step provides an important input for at least one later step in the sequence. It’s often impossible for a student to obtain a correct answer to some steps after making a mistake in any of the preceding steps. The average grade for such an N -part


question is the sum of the average grades for each part:

$$P_1 + P_1P_2 + \dots + P_1P_2 \dots P_N,$$

whose ratio to the maximum possible score ($= N$) is small if N is large. Although this standard grading approach might provide an important lesson to the student—a chain is as strong as its weakest link—instructors often use an alternative approach when manually grading this type of question. This alternative approach recognizes student effort and penalizes students only for the particular step or steps in which they made mistakes. If generated in this fashion, the average grade increases to $\sum_j P_j$, so this algorithm provides a more accurate assessment of the student’s work. Postprocessing lets us implement this grading strategy for online homework systems (see Figure 5). Note that the implementation diagrammed in Figure 5 could confuse students in the following two ways unless instructors explain these issues in advance:

- The computer marks early questions as incorrect but marks subsequent parts as correct (which the student calculated from submissions to the earlier parts).
- The grade for a later part or parts might change from correct to incorrect in response to a student’s modification of a submission to an earlier part (without a corresponding modification of a later part). This *nonlocal* grade change could be problematic in a time-constrained situation such as a quiz or test.

Several instances in which assessment tasks that students encounter in science or engineering courses, particularly upper-level ones, aren’t

handled well by standard implementation of automated grading systems. Postprocessing handles these tasks more appropriately and easily mirrors a human grader's simple—yet laborious—actions. In our last installment of this series, we'll focus on such tasks in laboratory courses in which assessment always poses a greater challenge. 

Reference

1. P.A. Knipp and S.R. Chaudhury, "Postprocessing in Automated Grading Systems, Part I," *Computing in Science & Eng*, vol. 11, no. 3, pp. XX-XX.

Peter A. Knipp is an associate professor of physics at Christopher Newport University. His research interests include surface phys-

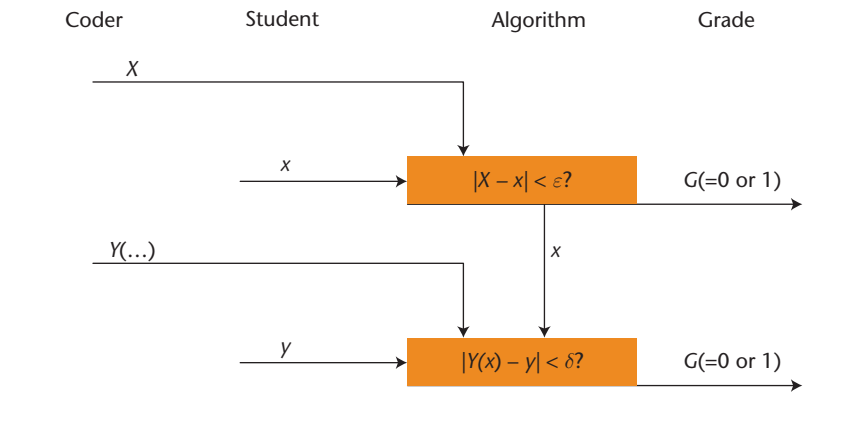


Figure 5. Flow chart for a two-part sequential question. The grading algorithm for each question part uses a key whose calculation is based on the student's submissions to previous question parts, if any. This generalizes easily for a question having three or more sequential parts.

ics, 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 University of California, Los Angeles. Contact him at schaudhury@auburn.edu.