

Checking Students' Symbolic Math on a Computer

Benjamin Crowell, Natural Science Division, Fullerton College, 321 E. Chapman Ave.,

Fullerton, CA 92832

Conceptual Mathematics

This note describes an open-source computer program that allows students to check their answers to symbolic homework problems. Traditional introductory physics courses emphasize solving a problem to obtain a numerical result, but this skill is only weakly correlated with conceptual understanding.¹ We state the laws of physics in the language of algebra, but many students find it difficult to connect a purely symbolic equation with its interpretation; to them, an equation is nothing more than an algorithm for numerical computation.

We can build up our students' conceptual-math muscles using activities that develop the link between an equation and its meaning. In particular, we can assign purely symbolic homework problems, in which the student is asked to interpret the result. The student finds, for example, that the acceleration of a cart on an inclined plane is $g \sin \theta$. Now why is it that the mass of the cart has canceled out? What happens when θ is zero? When θ is 90 degrees? We find few purely symbolic problems in most introductory texts.² Why? One reason is that students resist them: they need help in overcoming an initial feeling that such problems are "impossible" or "theoretical." To build students' confidence with numerical problems, we give answers in the back of the book, but this approach is not as useful for symbolic problems. A printed numerical

answer provides no information about how it was derived, but given that the acceleration of a cart on an inclined plane is $g \sin \theta$, many students will be able to cook up a derivation that gives the right result. The danger is not so much that students will misrepresent their understanding as that they will overestimate it.

The Software

To attack this difficulty, I have written a program called Spotter, which allows students to check their symbolic (and numerical) results. The software can be used, copied, and modified freely, and full documentation is available online.³ The instructor installs the program and an answer file on a server, and students check their answers through a web browser. Common incorrect answers can be added to the answer file, along with an appropriate hint for the student. The idea was suggested to me by another open-source program, LON-CAPA.⁴

Developing Spotter involved two challenges. First, there is the problem of accepting mathematical expressions in machine-readable form from naive users. Since we cannot expect students to learn a programming language for this purpose, Spotter must be able to interpret input whose form is as close as possible to ordinary mathematical notation, a language with all the inconsistencies and ambiguities common to human languages. For example, the symbol \ln may stand for a natural logarithm, or it may indicate a variable l multiplied by a variable n . Spotter is able to do a relatively good job at resolving these ambiguities because its input is framed within a certain context: the software knows what variables are expected to occur and what units they possess.

Second, the software must be able to recognize answers that are mathematically equivalent. If the instructor has entered $a+7b$ in the answer file, Spotter must accept $7b+a$ as well. It does this by choosing random values for the variables and checking whether the two expressions give the same result. The random numbers may be unphysical, so there may be results such as the square root of -1, or the arcsine of 2; these cases are handled by analytic continuation of all functions into the complex plane. This analytic continuation method makes it unnecessary for the instructor to specify which values of the variables to sample.

Results

My students like the program's immediate feedback. I have often found it difficult to convince my students to attend my office hours, and a salutary effect of using Spotter has been that they now often come to ask for help in troubleshooting solutions that Spotter has warned them are wrong.

I was initially surprised to find that some students simply did not take advantage of the software. One of my goals had been to allow students to correct certain commonly occurring mistakes before their papers arrived on my desk, but some students were still turning in these wrong answers, which Spotter would have detected. I suspect that this resulted from procrastination: students completed their homework too late to be able to consult Spotter. I have had better results with a new policy, under which students are required to enter their answers to symbolic problems in Spotter, and they automatically receive credit for them. They write the details of their calculations in a notebook, which I collect periodically.

I was concerned that students would try to abuse the system by inputting expressions and searching for the right answer by trial and error. This has only rarely happened, presumably because they know they are required to show their work in their notebooks, and also because most problems would admit an astronomical number of answers, all of them equally plausible in the absence of any physical understanding. The software also imposes a minimum waiting time before a student can retry a problem.

Trying, Using, and Installing the Software

Readers who are interested in test-driving the software can do so through a web browser³ without installing any software on their own computers. This can be done anonymously, without having to log in. There is a demonstration page containing the full text of two simple mathematical problems. For samples of physics problems, there is also a page that allows users to check their answers to the problems in my own textbooks; the full text of the problems is in the books, which can be downloaded for free in Adobe Acrobat format.⁵

Teachers who want to write their own answer files for use by their students will need to read the online documentation,³ but the following is a brief summary of the steps involved. I will be glad to answer e-mail⁶ from users who need help with installation. First, one downloads and installs the Spotter software on a Unix server. (The software is written in the portable Perl language, but has not been tested on Windows.) Next, one writes the answer file in a format called XML, which is similar to the HTML language used for writing web pages. The XML file is placed on the server, and the students are given a World Wide Web address at which they can

access it. If the students will be recording their answers online for credit, accounts are set up for them, and they choose their own passwords.

References

1. Eric Mazur, *Peer Instruction: A User's Manual* (Prentice Hall, New Jersey, 1997), p. 7.
2. A notable exception is Paul G. Hewitt, *Conceptual Physics* (Addison Wesley Longman, Reading, MA, 2001), which also has a strong emphasis throughout on connecting equations to their interpretations.
3. <http://www.lightandmatter.com/spotter/spotter.html>
4. <http://www.lon-capa.org/>, described in E. Kashy et al., *Amer. J. Phys.* **61**, 1124 (1993). The current version of CAPA (now called LON-CAPA) has a facility similar to Spotter's for handling symbolic mathematical answers. Another noncommercial piece of software that does similar things is WeBWorK, <http://webwork.math.rochester.edu>.
5. <http://www.lightandmatter.com/area1.html>
6. <http://www.lightandmatter.com/area4author.html>