Computer-based Mathematics and Physics for Gifted Remote Students

Raymond Ravaglia, Richard Sommer, Marc Sanders, Gary Oas
Education Program for Gifted Youth
Stanford University
USA
{ravaglia, sommer, sanders, oas}@epgy.stanford.edu

Charles DeLeone
Department of Physics
University of California Davis
USA
deleone@solid.ucdavis.edu

Abstract: Since 1985 the Education Program for Gifted Youth (EPGY) at Stanford University has been developing a series of stand-alone multi-media computer-based distance-learning mathematics and physics courses from the elementary school through the university level. Because these courses are used in situations where students do not have access to regular classroom instruction, we have had to make essential use of the computer both as an instructional agent and as a tool for media-distributed learning. We discuss our experience over the last fourteen years developing software and delivering effective instruction.

Introduction

The Education Program for Gifted Youth (EPGY) at Stanford University is a continuing project dedicated to developing stand-alone multi-media computer-based courses and offering these to remote advanced middle-school and high-school students through Stanford Continuing Studies. Since 1992, EPGY has taught advanced placement calculus and physics to over 600 advanced middle school and early high school students. The present enrollment in all EPGY courses is over 1700, with almost 160 in such university-level courses as Multivariable Calculus, Differential Equations, Linear Algebra, Number Theory, Optics, Thermodynamics and Modern Physics. (See www-epgy.stanford.edu for a complete list of courses offered.) EPGY is presently in the third year of a development effort funded by the Alfred P. Sloan Foundation to create 14 additional university-level courses in mathematics and physics. When these courses are put into place, it will be possible for an ambitious student who starts calculus in his or her seventh grade year, and who takes one course per quarter during the academic year, to finish high school five courses short of a degree in mathematics and five courses short of a degree in physics.

The EPGY course software, unlike traditional applications of computers in education, is intended to be the primary means of instruction, and not merely to supplement a regular class. In fact, as all EPGY students are physically remote, students never meet face to face with the instructor or each other. Because of this it has been necessary for us to think about the different ways of using the computer to facilitate student learning, and not to just treat the computer as an expensive broadcast tool. The EPGY course environment consists of interactive multimedia exposition, on-line exercises using symbolic computation, and automated reasoning to check student work. The course environment also contains facilities for collecting extensive data on all aspects of actual student usage of the software. Every time a student views a lectures, answers a question or proves a theorem, all relevant information concerning this event is preserved. The students send this information by e-mail to Stanford where it is automatically seeded into a database. This data collection allows course developers to isolate those areas in the course where students are having the most difficulty, so that they can refine them by adding new material to explain common mistakes, or by adding more detailed explanations to areas generating the most questions. It also makes it possible to see which features of the software prove useful to students and which are sources of frustration. In this panel discussion we focus on some of the features of the EPGY courses that have proven effective with students.

Multimedia Presentation of Proof-Saturated Mathematics

In the last year, as we have begun to focus on the development of courses beyond the level of the first year of calculus, we have been forced to find effective ways to present mathematical proofs to students. The need to do this stems from a variety of factors ranging from the complexity of the subject matter to the wide variation in the preparedness of the students taking the courses. An example (see Figure 1) from the Abstract Algebra course illustrates our approach to teaching abstract, rigorous, proof-saturated mathematics in this technology-enhanced environment.

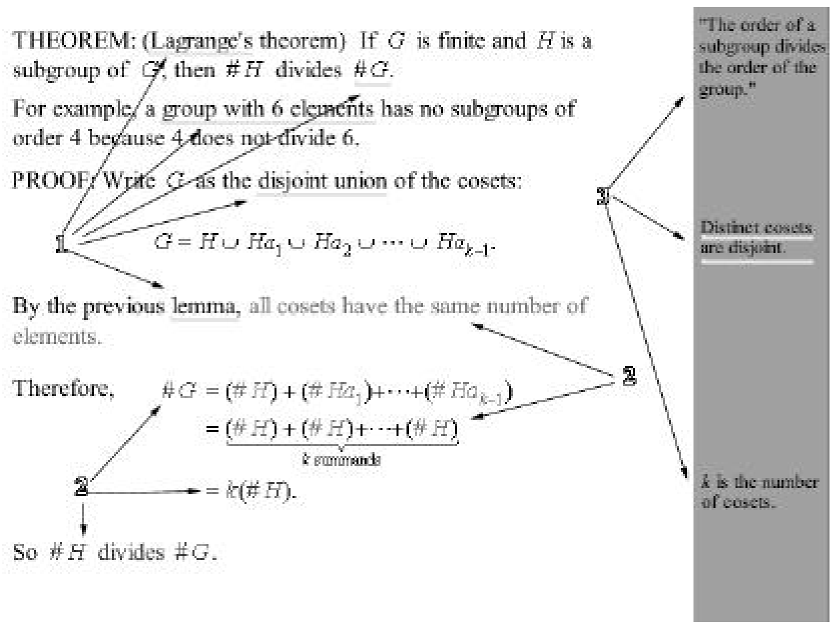

Figure 1: A sample EPGY course lecture.

Hypertext links, see "1" in Figure 1, afford students the non-linear browsing which is often necessary when reading multilayered, proof-based mathematics. Through these links, students have access to previously discussed material, such as definitions (e.g. subgroup, coset, order), notations (e.g. #*H*), theorems and lemmas (e.g. the previous lemma being used in the proof), as well as miscellaneous information (while dominant links are indicated, any term or expression can have a link associated with it as its "home page"). Some links that are presented are exploratory links that may access interesting, but somewhat tangential, topics which students are encouraged to investigate. Each link (and in particular the depth of any given link path) accessed by each student is recorded as part of the assessment of that student's progress. Additionally students are able to create links that point between their notebooks and the course lecture screens. This allows students to quickly return to specific points in the course.

Color coding, see "2" in Figure 1, is used to connect ideas, statements, formulas, etc. with other such objects so that students can more easily and quickly make the necessary deductions when following a proof. Often, when a conclusion of some argument or sub-argument appears, it is colored along with a central premise which leads to that conclusion. Color also is frequently used to tie together a previous result with a particular application of, or reference to, this result.

Margin notes, see "3" in Figure 1, are asides, remarks, comments, etc. which may be relevant to the learning and understanding of a proof, but are not part of the proof itself. In some sense they act as the difference between what a lecturer would *say* when presenting a proof and what he or she might *write* as a proof. Ideally, margin notes should become less common in the more advanced courses, as those students learn to fill such gaps themselves.

Animation, though not shown here, is used principally to maintain continuity throughout proofs and discussions. For example, in proofs requiring more than one page-screen, animation is used to move the most important and relevant statements onto the next screen in such a way that further reference to them can be made smoothly, without disrupting the general flow of the argument. In a way that static textbook examples cannot capture, animations are also used during computations to emphasize the dynamic components of the process one goes through in doing the computation.

Assessment, Feedback and Symbolic Computation

The ability to provide students with immediate feedback to their work is one of the great strengths that computer-based courses have. Immediate feedback is particularly important in the distance learning context where students face additional difficulties in submitting and retrieving written solutions to problems. Providing immediate feedback requires the ability to assess student work. Ideally this should include assessment both at the level of being able to answer standard questions, as well as understanding why a solution is correct. The types of questions students are asked in the course are of the sort that instructors traditionally ask after lectures or on examinations. They consist predominately of questions requiring closed-form mathematical expressions as solutions, though we have been experimenting with interactive proofs and will discuss these below. In free answer questions several issues must be taken into consideration.

One important issue is ease of input. If students have to type complex mathematical expressions in an input language, the odds that an incorrect response is caused by an error in typing will make meaningful evaluation impossible. Care must be taken to provide students with a convenient means of input that does not require a great effort to learn, together with the ability to see their input formatted, so that they can verify that what the computer has understood is in fact what they wished to express. The EPGY structural input system addresses both of these concerns.

Another issue is flexibility in answer form. Students should not have to constrain their answers to fit a particular form, outside of those constraints which an instructor would reasonably place upon them in a traditional class. The correct approach is to process the answers symbolically, taking into consideration their mathematical meaning, and considering possible correct answers in terms of equivalence classes. This minimizes the need to require that students conform to an arbitrary input standard, allowing the computer to understand natural variations of correct answers, thereby accommodating different approaches to a problem which can result in equivalent correct answers with different forms.

A simple example from the first year of algebra shows the range that a student's answer can take. Suppose a student is asked to factor the expression $12t^2 + t - 35$. One will likely want to accept as correct any of the following answer variants: $(3t - 5)(4t + 7)$, $(4t + 7)(3t - 5)$, or $(5 - 3t)(7 + 4t)$, not to mention several others with essentially the same form. On the other hand, the response $t(12t + 1) - 35$ should be rejected. Whether or not the student's answer is correct can be determined by passing the student's input and the author-coded answer plus specification of equivalence class to a symbolic computation program for evaluation and comparison. Exploiting the fact that the answers are mathematical expressions increases the flexibility for student input and simplifies author coding.

An additional benefit of this approach is the ability to automatically diagnose common student errors. There are a number of almost correct and incorrect answers that deserve special treatment. Errors in choice of variable, e.g.$(3x - 5)(4x + 7)$, errors caused by transposition of a factor or of the minus sign, e.g.$(4t - 7)(3t + 5)$, or $(3t + 5)(4t - 7)$, should be detected so that the mistake made by the student can be explained.

A more difficult problem than determining if a student's answer is correct is that of evaluating the student's entire solution. The link between understanding and the evaluation of work at this level is the sentiment back of the perennial dictum of "show your work." One step EPGY has taken towards being able to perform this sort of evaluation has been to make use of an interactive derivation system. The derivation system is an environment in which students can formally manipulate mathematical expressions by applying inference rules. A derivation system differs from a raw symbolic computation environment, such as Maple or Mathematica, by having the logical structure necessary to represent mathematical inference and logical dependency. This enables the derivation system to detect when students make fallacious inferences while working a problem. In the environment the student supplies the rule and the derivation performs the appropriate calculation. The results of the calculation are preserved for the student to further manipulate. A derivation of a problem is the set of steps from the statement of the problem to the solution. By requiring students to explicitly justify their inferences, it becomes possible to examine the process that a student goes through to produce an answer and not just the answer itself.

Symbolic Mathematics and Interactive Theorem Proving

The derivation system described above lacks the logical apparatus necessary to prove general mathematical theorems. We are just now looking for ways to integrate past work in interactive-theorem proving with the derivation system to create a more general theorem-proving environment that can be used to teach proof writing interactively on the computer in university-level mathematics courses beyond calculus. In this system students' mathematical and logical reasoning will be verified automatically. This should be a valuable tool for restoring deductive reasoning as an essential component in the advanced mathematics curriculum.

Our goal is to have a theorem-proving environment that is versatile and easy to use; it will be a hybrid system that uses automated reasoning to verify logical inference, and computer algebra to verify mathematical reasoning such as algebraic manipulation and numerical computation. Our aim is to allow students to freely apply valid logical intuitions and their prerequisite knowledge of mathematical facts in order to write proofs in our proving environment that are as close as possible to those one would be expected to produce in traditional versions of the same courses.

Students will use palettes to select proof strategies, inference commands, axioms, definitions, theorems, and mathematical symbols. With a structural input system like the one presently contained in our derivation system, students will be able to construct expressions quickly and efficiently. The user interface will consist of:

- A *workspace* for entering mathematical statements;
- a *proof history window* with a scrollable view of the entire proof;
- an *inference rule menu* for verifying steps in the proof;
- a *justifications menu* for selecting from a database of axioms, definitions, and previous theorems, as well as previous proof lines and a student-created database of lemmas;
- a *goal window* for viewing the current proof goal and various sub-goals that allow students to organize their proofs and apply good forward and backward proof strategies; and
- a *proof structure menu* for selecting from a list of proof structures such as proof by contradiction and proof by induction.

The incorporation of the theorem-proving environment into our Linear Algebra, Multivariable Calculus, Linear Algebra, and Ordinary Differential Equations courses, will occur incrementally over the next two years. We will carry out extensive testing with our students to refine both the theorem-proving environment and the changes in the course curriculum necessitated by the use of such a tool. Once interactive theorem-proving is running smoothly in these courses, we will work toward including it in other advanced courses such as Number Theory, Abstract Algebra, and Real and Complex Analysis, as well as in the courses below calculus, namely high school algebra and pre-calculus. It is our belief that an easy-to-use proof-writing tool can be applied to teach deductive reasoning in mathematics at an early age, and our hope is that this will allow deductive reasoning to once again be recognized as an important part of the advanced mathematics curriculum.

Improving Remote Interaction with Human Instructors

Even though the goal of EPGY is to automate as much of the instructional process as possible, interaction with human instructors remains an important component of the EPGY course model. Traditionally this interaction has been provided by asynchronous means such as telephone and e-mail. While effective, they constitute only a first step in providing robust interaction between instructors and students. There are several points to be made on this subject.

The first point is ease of communication. Any time a student wishes to send e-mail from within the course he or she may do so by simply selecting an option from a menu at the topic of the screen. The program will automatically append to this message the student's name and exact location in the course. This makes it possible for students to say things like "I do not understand this exercise" without having to figure out how to describe the exercise in question.

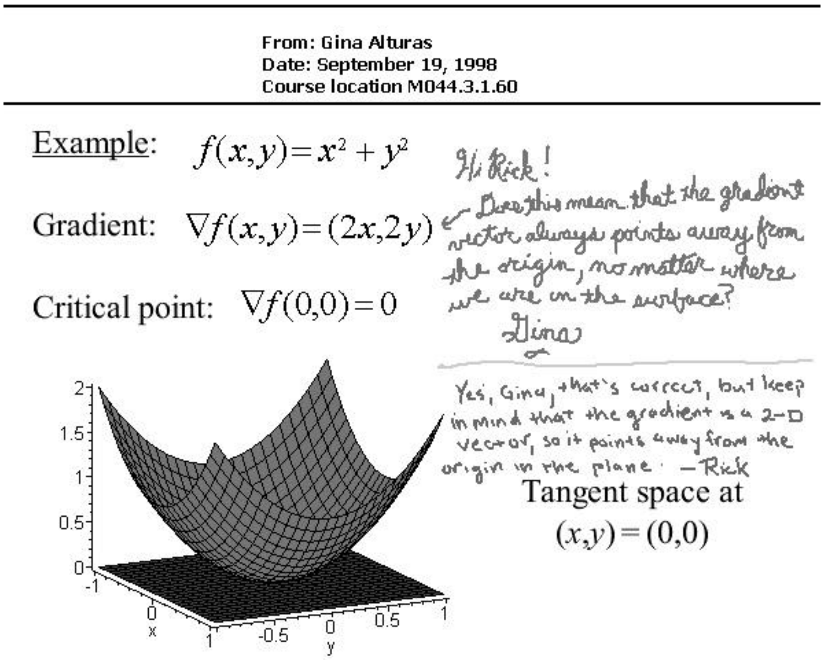

Figure 2: A sample e-mail message from an EPGY course.

An important component of e-mail communication in these courses is the ability to send graphics and sound in addition to text. The illustration above shows a message sent by a student to an instructor in the multivariable calculus course. The student has taken the screen image from the lecture she was in and has annotated it using a graphics tablet. The instructor has made his own annotation to her message as part of his reply. The student or the instructor could have included digitized sound in the message as well. Allowing handwriting and speech in messages makes asynchronous mathematical communication much more natural and it also frees students from having to learn outmoded linear notation.

The final point we wish to make here is the need to move beyond viewing asynchronous communication as the ideal mode of student/teacher interaction in the distance learning context. For the last several years we have experimented with using a variety of shared whiteboard conferencing environments in conjunction with internet telephony to create a cost-effective virtual classroom. In this virtual classroom one has the essential elements of any mathematics classroom: one has a common space on which to write (in this case the computer screen rather than the chalkboard) and one can talk whenever one is given permission to do so. The virtual classroom allows for the sort of immediate teaching experience that is common in traditional office hours or discussion sections, but which is usually thought of as unobtainable in the distance learning context. We expect this feature, as it becomes thoroughly integrated into the our courses, to have a profound impact on both future education and future course development at EPGY.

References

Ravaglia, R., Alper, T. M., Rozenfeld, M., & Suppes, P. (1998). Successful applications of symbolic computation. In *Computer-human interaction in symbolic computation*, ed. N. Kajler, New York, NY: Springer-Verlag, 61-87.

Ravaglia, R., Suppes, P., Stillinger, C., & Alper, T. M. (1995). Computer-based mathematics and physics for gifted students. *Gifted Child Quarterly*, *39*: 7--13.