WhistleSoft, Inc

LA-SUB--98-56

Final Report to LANL-TCO Office

A Computer-Based "Laboratory" Course in Mathematical Methods for Science and Engineering: The *Legendre Polynomials* Module

Richard R. Silbar, *WhistleSoft, Inc., Los Alamos NM*
September 28, 1998

Abstract of Project

WhistleSoft, Inc., proposed to convert a successful pedagogical experiment into multimedia software, making it accessible to a much broader audience. A colleague, Richard J. Jacob, has been teaching a workshop course in mathematical methods at Arizona State University (ASU) for lower undergraduate science majors. Students work at their own pace through paper-based tutorials containing many exercises, either with pencil and paper or with computer tools such as spreadsheets. These tutorial modules cry out for conversion into an interactive computer-based tutorial course that is suitable both for the classroom and for self-paced, independent learning. WhistleSoft has made a prototype of one such module, *Legendre Polynomials*, under Subcontract (No. F97440018-35) with the Los Alamos Laboratory's Technology Commercialization Office for demonstration and marketing purposes.

Technical Milestones (from Proposal)

- **Milestone 1**: Adapt our Style Guide template as needed for converting the paper workbook modules already in hand into software modules using Authorware.
- **Milestone 2**: Research ways to ensure that the student actually *does* the exercises, which form the interactive basis of the learning experience.
- **Milestone 3**: Convert *one* of the presently available workbook modules into an Authorware piece. This will involve some additional research into how to expand these modules with more graphics and computer animations.

- **Deliverables**:

1. At about 4 weeks, report on progress in meeting the first two milestones. We expect that Milestone 1 will be completed at this time, but Milestone 2 will still be in an intermediate state.
2. At about 7 weeks, report on the expected completion of Milestone 2 and description of the initial prototype for the module that constitutes Milestone 3.
3. At about 10 weeks, submission of a complete final report for the project, including description of the final version of the Legendre Polynomials software module.

DISCLAIMER

The contract with LANL was signed by Richard R. Silbar (WhistleSoft, Inc.) and James R. Jefferis (LANL) on July 15, 1998. The work was to be finished and the final report delivered before the end of the LANL fiscal year, September 30, 1998.

Summary of Work Accomplished

To re-iterate how this project was conceived, one of us, Richard Jacob (RJJ), has been developing the curriculum for and teaching a workshop course in mathematical methods at Arizona State University (ASU) for lower undergraduate science majors. The ASU physics majors and others start his two-semester sequence as second semester sophomores, after completing the basic three-semester sequence in calculus-based physics. The PHYS 201 and 302 sequence is *not* a lecture course. Students learn mathematical methods and how they are applied in physics by working through paper-based tutorials containing many exercises at their own pace. They do this either with pencil and paper or with computer tools such as spreadsheets and software like Mathcad.

These tutorial modules, about 550 pages in all, are in a form suitable for conversion into an interactive computer-based tutorial course that is suitable both for the classroom and for self-paced, independent learning. When the two authors met last January at the Winter Meeting of the American Association of Physics Teachers and saw what each other was doing, it was natural to consider starting a collaboration to see how easily this could be done. We had in mind, of course, that by making it accessible to a much larger audience, this new curriculum could lead to a commercially publishable software product that has a reasonably large market.

In June 1998 the Los Alamos National Laboratory's Technology Commercialization Office issued a call for proposals from Northern New Mexico businesses for small, short-time-scale projects which could enhance the local economy. WhistleSoft, Inc., a company that spun off of LANL five years ago and which creates multimedia products for science and engineering, took this opportunity to propose converting, in the ten weeks available to the project if funded, **one** of RJJ's paper tutorials into a multimedia computer-based training application. The purpose of this exercise was to establish that there were indeed no major problems in doing so and to produce a prototype that could be tested in the fall PHYS 302 course. The prototype could also be used for finding suitable funding and a publisher for the whole mathematical methods course.

The LANL-TCO accepted this proposal and this report describes the result of our work, completed on deadline. The software module we produced, *Legendre Polynomials*, based on eleven typewritten pages in the paper worksheet, turned out to be rather larger than we originally expected. In fact, it is the biggest of the five completed physics software tutorials that WhistleSoft has produced. One measure of this is that the number of "icons" in *Legendre Polynomials*, about 13,000, is almost a factor of three larger than in any of our other pieces.

How was it possible to do this in so short a time? First, the content already existed in a well prepared form. We could often author directly from the paper *Legendre Polynomials* tutorial without having to go through the intermediate stages of developing story boards. In those cases where we did lay out storyboards, a rough sketch was often all that was necessary.

Second, WhistleSoft had developed earlier, under a DOE SBIR grant for a different project, a Style Guide and user interface for efficient student-centric learning of scientific and mathematical material. Our Style Guide template represents a major extension of the functionality that already comes with Macromedia's Authorware, which is WhistleSoft's main authoring application. In the Style Guide template we have taken care to separate, as much as possible, the underlying programming structure from the content pages. The Style Guide sets up the menus, buttons, and navigational options. It illustrates (and sets up models for) invoking a pop-up window, executing a hyperlink to a new page, and programming a page so that the student sees the materials in a series of small steps (progressive disclosure). We anticipated that almost all of the WhistleSoft's present Style Guide template could be re-used in authoring *Legendre Polynomials*. In fact, that turned out to be the case; **Milestone 1** was easily achieved, since very little had to be done.

It turned out that there was only one area in which the Style Guide functionality had to be extended, and this was in conjunction with meeting **Milestone 2**. The three calculation icons we added to the Style Guide library were all related to the research we did to ensure that the student actually *does* the exercises in the tutorial. As is well known, it is the interactive act of doing things, which forms the basis of a true learning experience. In dealing with this problem we came up with two approaches, which then later became re-useable models.

One, which is a model we call "Wofie" (for "Write or Fill In Equation"), asks the student to work out the algebra on paper and pencil, and then input the answer in the Authorware piece by clicking to place equation elements in a solution box. Figure 1 shows a screen which uses this model.

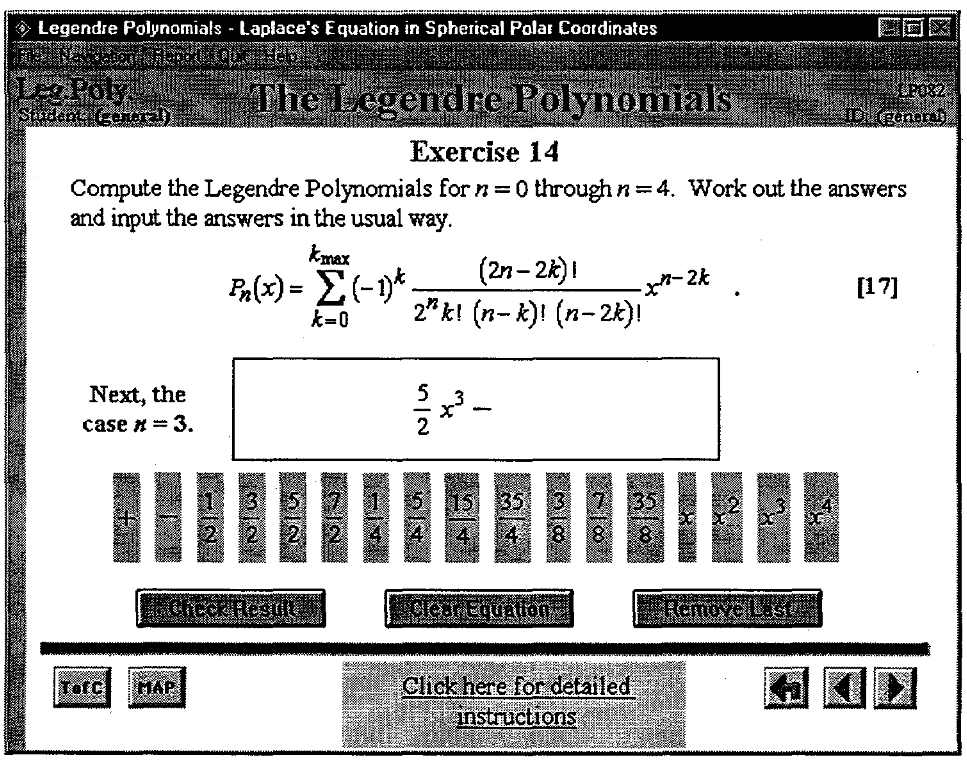

Figure 1. Screen shot from *Legendre Polynomials* illustrating use of the Wofie Model.

The figure actually shows the screen at the point where the student has successively clicked on three of the equation elements in the menu beneath the solution box. Each click, say, on the "x^3", moves a copy of that element up to the next available position in the solution box. At this point in

Fig.1 the student must still finish up data entry by clicking on the "3/2" and the "x" elements. When done, he or she then clicks on the "Check Result" button and gets a "Ta-Dah!" sound along with a visual "Correct" text. If the answer input is wrong, the response is a "Chord" sound and a "Try Again" text.

We have chosen to arrange things so that the student *must* get the correct answer before he or she gets credit for having visited this (and other) pages. This information is gathered by the software and is available to the student from the "Report Card" menu item available in the "Report" pulldown menu at the top of the display window.

The other approach to ensure active student participation uses a call-and-return procedure which takes the student out of Authorware into MathCad. (Mathcad is an inexpensive mathematics application from MathSoft, Inc., that is widely available to many students; it is certainly available to those taking the ASU mathematical methods course.) On arriving at the Mathcad worksheet, the student then follows the directions given there. He or she might be asked to work out a number or an algebraic result. Or, just to play with the formulas already given in the worksheet to see, for example, the functional forms or graphs for a particular Legendre polynomial. On exiting Mathcad, the student returns to Authorware. If it was the case that the student was asked to get a correct answer, and did, the Mathcad worksheet will write a codeword in a hidden file that Authorware reads. This allows the Authorware piece to validate that the student did the work properly and thus gets credit for it. This "JumpOutReturn" from Authorware to Mathcad and back turned out to be a simpler procedure than we first expected. It did not entail having to write a DLL to facilitate the communication between the two applications. It also has the advantage of being a cross-platform solution, since MathCad exists for the Macintosh platform as well as Windows 95 computers.

The efforts to meet Milestone 2, described in the above paragraphs, essentially extended throughout the project's ten weeks. It was premature, in our second interim report of September 3, to declare the model-making effort over. Even late in the game of finishing up the *Legendre Polynomials* module, we came across situations while authoring where we felt it was useful to make yet another modification in the Wofie model. I can imagine that, when we return to this project under other funding, we will continue to make changes and improvements in this model.

Let me turn now to what we accomplished in meeting **Milestone 3**. There is probably little to say here other than it was relatively straightforward authoring. *Legendre Polynomials* consists of five main sections. The Table of Contents Page looks like:

0. Introduction
1. Laplace's Equation in Spherical Coordinates, Legendre's Equation
2. Legendre Functions
3. Summary of Properties
4. The Spherical Shell, an Electrostatics Problem

The bulk of the material is in Section 2, which breaks down as follows:

2.1 Series Solution
2.2 Legendre Polynomials

with the latter sub-section being further broken down into

2.2.1 Legendre Functions of the Second Kind
2.2.2 Generating Function
2.2.3 Recursion Relations
2.2.4 Rodrigues Formula
2.2.5 Orthogonality and Normalization

We had some initial concerns about how four people programming in one piece would proceed without getting in each other's way. (The Authorware programming environment is not well set up for version control and multiple authors.) We eventually came to the following working solution: Bob Williams "held the baton" for most of the time for consolidating other people's work. At the time we moved into production mode, Kris Kern authored Section 2.2.3, Bill Mead handled Section 2.1, and Bob did Section 2.2.2. The remaining sections were authored by me.

Not all of the Content Pages in the *Legendre Polynomials* tutorial end up looking like that shown in Fig. 1. There are pages containing historical vignettes (of Laplace and Legendre, in this case). Some pages just present problems (there are four) that could be required, turned-in homework or, alternatively, worked through in the classroom under the guidance of an instructor. Other pages fill in background material, often using common multimedia techniques, such as moving text or multiple-choice items the student is asked to choose from. Also, when a given equation, such as [17] in Fig. 1, reappears later, it is hyperlinked back to the page where it first presented. Other navigational aids available to the user are the Table of Contents (mentioned above), a Concept Map, and an Index, all of whose items are hyperlinked to their respective pages.

Certain Content Pages have a self-test Question Button, which appears next to the Map Button at the lower left in the navigational controls area under the green separation bar (see Fig. 1). This button leads the student to a page with one or several questions, each of which in turn lead to an page presenting multiple choice answers. Clicking a blue choice brings up a response indicating correctness. Wrong-choice responses often give an explanation for why that is wrong. The Report Card records the number of correct choices the student made.

The features mentioned in the preceding two paragraphs all came easily because of the use of our Style Guide template. One novel aspect of the project which involved some thought and working through was how we handled complicated mathematical material. (This piece is far more mathematical than the ones we have done in the *Accelerators and Beams* project.) We did most of this using an equation editing program called MathType (from Design Science). What is interesting is that we learned it was possible to embed a given equation in the Authorware piece ***as*** a MathType equation. That is, when in authoring mode in Authorware, we could double-click on a selected equation, and it would very quickly bring that equation up in the MathType application for editing, if so desired. Then, after editing, one could select and copy the revised equation to the pasteboard. On returning to Authorware, the new equation could then be inserted in place of the old by a "paste special" operation. This procedure turned out to be much more efficient in inputting the mathematics than we originally expected.

At the end of this project, the *Legendre Polynomials* tutorial is complete, covering everything that is in RJJ's paper worksheet but the solution hints (many of which are incorporated in the authored pages) and the exercises on looking up formulas in Abramowitz and Stegun's *Handbook* *of Mathematical Functions*. (We would have violated that reference book's copyright were we to have done that section.) In its final form, the tutorial consists of 86 Content Pages, 7 Question Pages and 15 Answer Pages. The executable version for Windows 95/98/NT platforms is less than 4 MB. We have checked that a similar executable version for the Macintosh platform can also be compiled with very little further effort.

Testing of this module in the ASU class will take place in late October, outside the scope of the LANL-TCO project. We still have to clarify how we are going to deliver the software to those students. We will probably make CD-ROMs for this purpose. Also, we need to decide how we will evaluate the student reactions to the program.

To conclude this report, I want to say that this has been quite a fun project, but by no means the "slam dunk" that we originally expected it to be. I feel the *Legendre Polynomials* tutorial we created here will be a big help in our finding funding for developing the entire mathematical methods course.