Using CAS in a Graduate Numerical Methods Course

Shirley B. Pomeranz
The University of Tulsa

1 Introduction

This paper describes the introduction of a computer algebra system (CAS) (e.g., *Mathematical* or *Maple)* as a tool in a course which has traditionally used FORTRAN or C as the programming tool of choice. The claim here is not that one type of programming language-CAS (interpretive language) versus FORTRAN or C (compiled languages) -is generically better, but that for teaching purposes, each offers different advantages. Some of the benefits of a CAS approach will be described [1].

2 Background

The course, Numerical Methods for Engineers and Scientists, offered by the Department of Mathematical and Computer Sciences at The University of Tulsa, is taken by graduate students in chemical, mechanical, and petroleum engineering, geosciences, and mathematical and computer sciences. Occasionally there is a graduate student from some other discipline, e.g., business administration. The topics covered, primarily numerical methods for partial differential equations (pales), include finite difference methods, method of characteristics, and the finite element method.

3 Traditional Course

There is usually a course text focusing on the descriptions and comparisons of methods, i.e., on the theoretical/analytical aspects. For example, these topics include consistency, stability, and convergence analyses for finite difference time-marching methods for parabolic and hyperbolic problems, efficient linear solvers for elliptic problems, and an introduction to error analysis for the finite element method.

purposes. Being able to conveniently graphically interpret a solution is crucial for students' understanding of the correctness of a solution [6]. If desired, *Muthematica,* for example, can be used as a front-end to run code that is already written in FORTRAN or C, via its *MathLink* utility. Notwithstanding that for CPU-intensive computations CAS are not efficient, CAS are good for writing and developing prototype programs.

The symbolic capabilities of CAS came in very handy for such usually tedious hand computations as truncation error analyses for finite difference methods. *Mathematica's* symbolic treatment of Taylor expansions can remove much of the tedious calculation and frustrating careless algebra that usually accompanies these calculations [7], [8]. Students can more clearly see the reasons for the importance of such concepts as truncation error and order of accuracy when they do not get lost in the repetitive algebra details. Symbolic manipulations and helpful graphics for stability analyses can also be utilized.

5 CAS Projects

Students organized some of their homework into *Mathematica* notebooks and these were submitted electronically to the instructor [9], [10]. Models of interactive *Mathematical* notebooks had been given previously in the computer lab portion of the course.

Specific *Mathematical* assignments included an implementation of the Crank-Nicolson method for the solution of a one-dimensional diffusion equation initial-boundary-value problem (ibvp). The results included numerical and graphical output. There was a *Mathematica* implementation of an explicit finite difference solution to a wave equation ibvp. A transport equation ibvp was discretized via the Lax-Wendroff and upwind finite difference methods. *Mathematica's* animation of the graphics clearly showed the dissipative and dispersive effects of the numerical methods. A simple two-dimensional elliptic finite element program using piecewise-linear shape functions was also assigned. Again, students interpreted their results graphically to produce 3-D surface graphics of the solution (temperature), contour plots, and plots of two-dimensional vector fields (heat flux). Contour plots of temperature and vector-field plots of heat flux were superimposed to gain more insight into the relationships between various quantities. Graphics showing jump discontinuities in the (numerical) solution gradient components at interelement boundaries in the finite element work led naturally into a discussion of error estimation.

6 Summary

Finally, especially considering the diverse student population served by this course, each student had become familiar with a generally very useful tool for applications beyond this course. From the point of view of a member of the Mathematics faculty, a CAS such as *Mathematica is* more open-ended in its applications and may turn out to be a more generally useful tool for many of the students in this course than the traditionally used FORTRAN.

Actual programming examples and problems may or may not be included in the text. The text by Granville Sewell, **The Numerical Solution of Ordinary and Partial Differential Equations [2],** includes FORTRAN subroutines and programs and refers to PDE/PROTRAN (IMSL'S partial differential equation package). The FORTRAN programs can be used by students as templates for their own projects. On the other hand, **Numerical Solution of Partial Differential Equations** by K.W. Morton and D.F. Mayers [3], has no specific programming language or software coordinated with the text. This is intentional on the part of the authors. Quoting from their preface, "...However, numerical analysis has very practical objectives, so there are many numerical illustrations of the methods given in the text; and further numerical experiments can readily be generated for students by following these examples. Computing facilities and practices develop so rapidly that we believe this open-ended approach is preferable to giving explicit practical exercises." Similarly, the text by Celia and Gray [4] and texts by many other authors do not refer to any specific language for use in implementing the numerical methods studied.

4 CAS Version of the Course

A CAS, specifically, *Mathematical, was* chosen as the programming tool for this course instead of the traditional choice, FORTRAN. Mathematical was used with the text by Morton and Mayers [3]. In addition, the text, Mathematical **for Scientists and Engineers,** by Thomas B. Bahder [5] was selected as a tutorial and reference for *Mathematical.* It was emphasized to the students that in actual applications a compiled language such as FORTRAN or C would be used for the CPU-intensive parts of the problem. CAS, which are interpretive languages, are not efficient-they are too slow—for these computationally intensive parts of the problem. However, as with many aspects of numerical methods themselves, in the teaching of such methods, there are trade-offs. There are advantages to the use of CAS as a tool in the teaching of numerical methods for pales.

For the first several weeks of class, the course met in a computer lab and was taught as a course with computers in the classroom. This was done to permit those students who were less familiar with CAS to get up to speed. In the end, there were a couple of students whose expertise with CAS far exceeded that of the rest of the class. These students were very willing to serve as lab mentors for the rest of the class. With respect to CAS homework, the students were assigned to work together in small groups. This was intentional, considering that the ability to work well in groups is vital in the workplace (and that the students would probably be working together anyhow). Some of the CAS computer work served as previews of topics that were soon to be studied in a more theoretical context. Thus, the students got to see/do some examples of finite difference and finite element work (at a very introductory level) before they saw the analytical presentation of these topics. This helped the latter presentations make much more sense.

The rationale for offering a CAS-oriented course is based on the following benefits. CAS are easier to program than, say, FORTRAN (in spite of the syntax rules that accompany either *Mathematical* or *Maple.* Their interactive nature, allowing the execution of code one line at a time, permits students to discover many types of errors and fix them at this phase. The graphics capabilities of CAS are an aid in the debugging process and, more importantly, in interpreting and understanding the numerical solution to a problem. This may be the greatest strength in the use of CAS (versus FORTRAN or C) for teaching

The intended emphasis in this paper has been that of a trade-off. What has been presented is another possible approach to the teaching of an introductory graduate numerical methods course. Using CAS does not necessarily result in a globally better approach to teaching such a course, but it does offer advantages in a number of areas over a more traditional approach.

References

[1] Robert Lopez and Mark Yoder, NSF Workshop: Revitalizing the Engineering, Mathematics and Science Curriculum via Symbolic Algebra, Rose-Hulman Institute of Technology, Terre Haute, IN., July 10-15, 1995.

[2] Granville Sewell, **The Numerical Solution of Ordinary and Partial Differential Equations,** Academic Press, Inc., N. Y., 1988.

[3] K.W. Morton and D.F. Mayers, Numerical Solution of Partial Differential Equations, Cambridge University Press, Cambridge, England, 1994.

[4] Michael A. Celia and William G. Gray, Numerical Methods for Differential Equations, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1992.

[5] Thomas B. Bahder, Mathematical for Scientists and Engineers, Addison-Wesley Publishing Company, Inc., N.Y., 1995.

[6] David Epstein and Silvio Levy, Experimentation and proof in mathematics, Notices *of the AMS, v.* 42,670-674 (June 1995).

[7] Stan Wagon, Taylor polynomials efficiently, *Mathematical in Education* and *Research,* vol. 4, 54-57 (1995).

[8] E. Kreyszig and E.J. Normington, Mathematical Computer Manual for Advanced Engineering Mathematics, John Wiley and Sons, Inc., N. Y., 233-234, 1995.

[9] Haim H. Bau and Karen Lebiedzinski, Teaching Thermodynamics with Mathematical, *Computers in Education Journal, v.* III, 47-55 (Oct.-Dee. 1993).

[10] Malcolm Getz, Mathematics across the curriculum, *Vanderbilt Magazine,* Vanderbilt University, Nashville, TN., 21-22 (Fall 1994).

SHIRLEY B. POMERANZ

Dr. Shirley Pomeranz is an Associate Professor of Mathematics in the Department of Mathematical and Computer Sciences at The University of Tulsa. Her research interests include numerical methods and, in particular, the finite element method and finite element error estimation. Her e-mail address is: pomeranz@euler.mcs.utulsa.edu.