Computer Algebra for Mathematics Majors

Michael Monagan,
Department of Mathematics, Simon Fraser University
mmonagan@cecm.sfu.ca

May 29, 2008

Extended Abstract

Every mathematics department has had to decide how it will integrate the use of mathematical software into its undergraduate mathematics curriculum. Some departments have chosen to supplement the syllabus of some of their lower division courses, usually the calculus, linear algebra, and differential equations courses, with a "mathematical software component". This may be done by including a weekly computer lab. Others have substantially revised their core calculus courses to follow the "Calculus Reform" where the use of the computer becomes an integral part of the course and assignment problems.

At Simon Fraser University, we have, instead, created a new regular course "MACM 202: Modeling and Computation" which is devoted to instruction in the use of mathematical software (we use Maple). Weekly instruction includes three lectures and one lab. The course is a second year course that is required for our mathematics majors. The prerequisites are first year calculus and first year computer programming. Because it's a second year course, and because of the prerequisites, we can apply the mathematics already learned to solve more interesting problems and show the students where and how the software is useful and indispensable.

Naturally, we want to include a systematic coverage of tools for solving equations, calculating derivatives and integrals, manipulating and simplifying mathematical formulae, and graphing functions, curves, and surfaces, etc.

But, we also want to get the students to practice their newly acquired programming skills. Otherwise they'll forget how to write a loop and then they'll not be able to do anything! The way we've chosen to do this is to assign three types of questions. Firstly, exercises for learning to use the commands in the mathematical software and practicing the writing of loops. Secondly, because there is some new mathematics introduced in the course, there are some problems to be done by hand. And thirdly, to pick a more challenging problem to work on for each assignment. The six problems we work on are

1. Computing the period-doubling bifurcation points in the logistic map $f(x) = ax(1-x)$ and approximating Feigenbaum's constant.

2. Constructing a two dimensional cellular automaton for simulating Conway's game of life or modeling a forest fire.

3. Constructing fractal images in 2 dimensions and 3 dimensions using recursive and iterative constructions.

4. Solving a first order ODE using Euler's method, Heun's method, and a second order Taylor series method – and comparing the convergence of the three methods.

5. Visualizing (in 3D) partial derivatives, the tangent plane and higher order series approximations of a function of two variables $f(x, y)$ at a point $x = a, y = b$.

6. Symbolic stability analysis of the fixed points of a system of two first order non-linear DEs in terms of the parameters.

Each of these project problems requires some background mathematics which is presented by the instructor, some programming, some visualization, and some application of numerical and/or symbolic methods. For example, to compute the bifurcation points of the period-doubling cycle of the logistic map, we can express them as a solution of a system of two non-linear (polynomial) equations of the form $\{g(a, x) = x, g'(a, x) = -1\}$ in two unknowns a and x. We can then solve the system symbolically or numerically using the appropriate solver in the computer algebra system. If we try to solve the system symbolically, using elimination, this leads naturally to resultants and a discussion of methods that the solver uses. If we try to solve the system numerically, this leads naturally to the method of bisection, Newton's method, and automatic differentiation. We can also solve this system graphically by intersecting the surfaces $z = g(a, x) - x$ and $z = g'(a, x) + 1$.

What I would like to present in the talk is the following.

- Describe the course and how we assess the students.
- List the advantages of having a new course wholly devoted to the use of mathematical software over modifying existing courses to include a mathematical software component.
- Describe more completely the content of the course, in particular, show solutions to the six problems and illustrating, using problems 1,4 and 5 above, where computer algebra systems, with their ability to manipulate mathematical formulae symbolically have an advantage over numerical systems like Matlab.
- Discuss that we need to tell to teach the students that there's a limit to what we can do using exact algebraic computation – and that for many problems, a numerical solution will be the best we can do.