# Teaching Dynamical Systems with a Computer Algebra System

Jaime E. Villate

Universidade do Porto, Faculdade de Engenharia and INESC Porto
Rua Dr. Roberto Frias, Porto 4200-465, Portugal,
villate@fe.up.pt,
WWW home page: `http://villate.org/`

**Abstract.** We report on our experience in teaching a dynamical systems course to second-year engineering students. The course has been taught with a hands-on approach, using a Computer Algebra System: Maxima. We have developed some additional programs to help students explore the dynamics of the systems, in a more intuitive way and without loosing too much time with computer programming. The result has been very encouraging; students are better motivated using this approach, as compared to the more traditional methods that we used in the past.

## 1 Pedagogical considerations

Differential equations play a very important role in Engineering and Science. Many problems lead to one or several differential equations that must be solved. In the last decades the way differential equations are studied has undergone dramatic changes. Much attention is now given to geometrical aspects, and several concepts from classical dynamics have been extended to equations found in various other fields. That new area of research has become known as Dynamical Systems.

The rapid development of the theory of dynamical systems has been powered by the emergence of the computer. Before the computer age, non-linear differential equations had to be approximated to linear ones in order to use analytical solution techniques. The computing power provided by today's computers can now be used to find numerical solutions to non-linear equations.

This new approach to the field has disseminated to the educational sector, and many educators [1] are advocating a teaching methodology that relies more on graphical analysis and computer calculations, which makes it possible to introduce modern concepts such as chaos and fractals from an early stage. Furthermore, the graphical approach used by this new way of teaching differential equations makes the subject more appealing and easier to understand.

A modern course on dynamical systems must necessarily include many results from numerical calculations and computer graphics. Undergraduate students may get too distracted from the goals of the course if they have to make their own computer programs. On the other hand, if they are given results obtained by somebody else they will not gain the experience necessary to comprehend the concepts involved.

Computer Algebra System (CAS) can be used to overcome the difficulties mentioned above. A CAS allows students to obtain their own results, without getting lost in programming details or complicated mathematical calculations.

The course we are teaching on dynamical systems is attended by second-year undergraduate students majoring in Computing and Information Engineering. There is an average enrolment of around 120 students. The master classes are given using an projector connected to a computer where the teacher can work on a CAS session.

The students attend weekly two-hour sessions in groups of around 20 students, at computer rooms equipped with one computer for two students. They can work either in GNU/Linux or the Windows operating system. In each of those sessions students will log into a Moodle Course Management System where they can find a list of problems that they are expected to solve during the session.

They will have to start a CAS session to solve the proposed problems. By the end of each session, each student will submit in the Moodle server a batch file with the commands used to solve the problems proposed. The lecture notes provided to the students have become a book that can be downloaded from http://fisica.fe.up.pt/pub/maxima/sistdinam.pdf

The evaluation of the course is made through multiple-choice quizes, which are then graded in the Moodle server, and a term paper on a project assigned to each student. Those term projects involve some bibliographic research and analysis of a dynamical system.

## 2 Dynamical Systems and Maxima

Maxima [3] is a CAS that includes several tools for the analysis of dynamical systems [2], such as solving differential equations and finding eigenvalues and eigenvectors of a matrix. There are several libraries written in various different programming languages, which can be used for the same purpose. However, using those libraries implies a good knowledge of the programming language used.

The subject of dynamical systems is too broad to be covered by a book on computer programming. Teaching manuals for dynamical systems usually either show the results, leaving the programming to the student, or provide a complete program that the student can use to obtain the results. In both cases the student will take the results for granted and will only get a very superficial comprehension of the subject, unless he/she has the time and skills to do his/her own programs.

Consider for instance a simple numerical method to find the solution of a differential equation. The student will find an implementation of that method in almost any programming language chosen, but in each case the student will have to write a new program for every new equation to be solved. A CAS might include a program that implements a numerical method for differential equations and which takes as input the differential equations to be solved. The same program can be used to solve several different problems, without having to do any programming. The student can move on to a graphical analysis of the solution, without spending too much time with programming.

Other advantages of Maxima are its distribution with a Free Software license and the fact that it was derived from one of the oldest Computer Algebra Systems, Macsyma, developed in the late 1960's at the M.I.T. CAS software can lead to wrong results in certain cases; its results can be trusted if it has been tested extensively in various fields. Thus, having been used by many people during several years is a strong point in favor of a CAS.

There are several graphical interfaces for Maxima. We have used Xmaxima, which includes a browser for Web pages with active links. Figure 1 shows a Web page being displayed by Xmaxima.

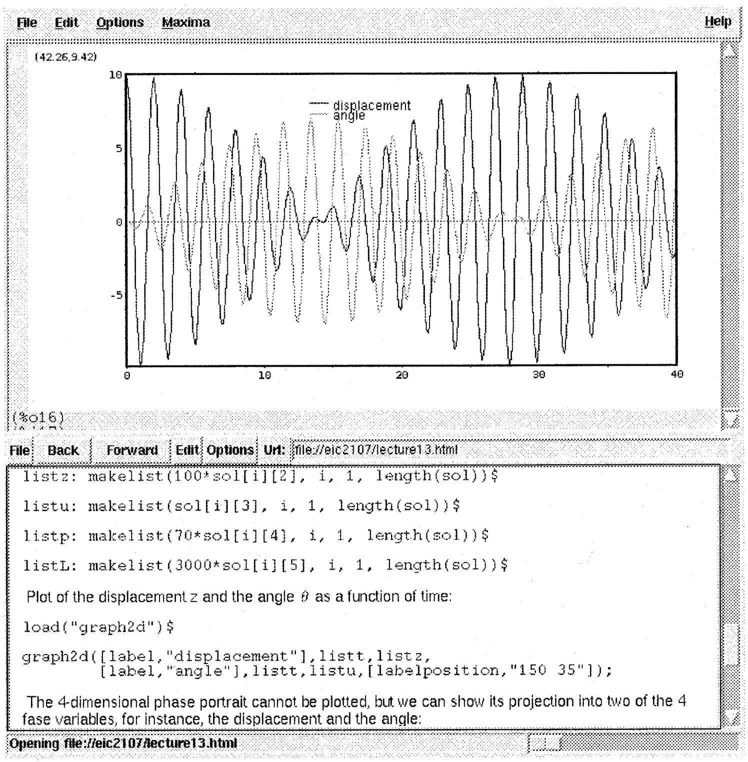


**Fig. 1.** An Xmaxima session, showing the browser on the bottom and the Maxima interface window on top.

As any other CAS, Maxima has a syntax that at first can be unfamiliar to students. They also have to become familiar with the greater variety of objects that can be manipulated in a CAS: rational numbers, algebraic expressions, mathematical equations, etc. Once they overcome that initial stage, they get very motivated by the many features available: factorization of algebraic expressions, solving equations, differentiating and integrating functions, and various other utilities.

## 3 Technical considerations

Being free software, we have been able to study Maxima's source code and to introduce a few changes to adapt it to the needs of our course. One of the most important things we missed in Maxima was a function to plot direction fields. A design of a direction field is very useful in order to understand the possible solutions to a system of differential equations.

The original maintainer of Maxima, Professor William Schelter, had already written a set of scripts — NetMath — that included a function to plot direction fields. After he passed away in 2001, that function remained within the source code of Xmaxima, but it could not be called directly from Maxima. We have recovered that function, creating an interface from Maxima. We have also made some improvements to the graphical interface Xmaxima.

The program to draw direction fields is called `plotdf` and it is included in Maxima since version 5.9.2. An example to plot the direction field of a pendulum with air resistance follows:

```
plotdf([y, -g*sin(x)/l - b*y/m/l], [sliders,"m=0.1:1"],
       [parameters, "g=9.8, l=0.5, m=0.3, b=0.05"]);
```

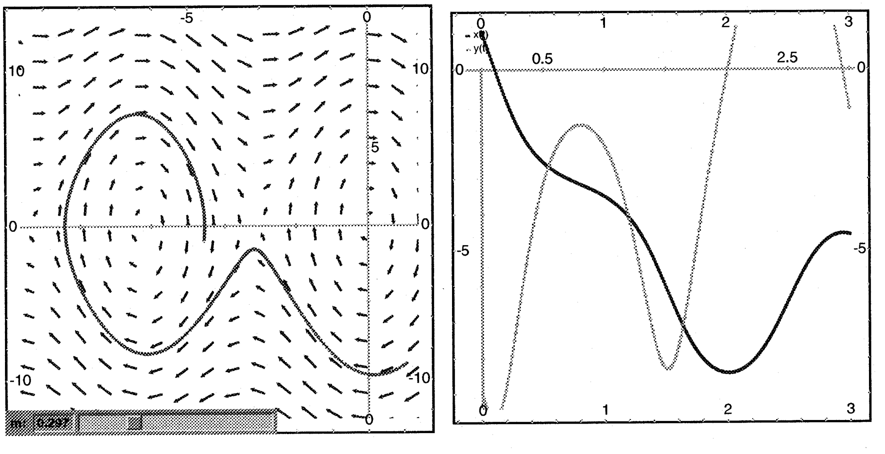

**Fig. 2.** An example of the results of the program to draw direction fields.

The result is shown in Figure 2. A slider button was drawn, which allows the parameter $m$ to be changed from 0.1 to 1, and the direction field will be updated in real time. The user can also click on a point of the phase space, to obtain the orbit that goes through that point. There is a menu with several options; one of those options can be used to make a second plot, showing the evolution of the phase variables as a function of time (right-hand side of Figure 2)

We have also developed other additional functions for Maxima, to obtain the numerical solution of a system of equations and to draw various graphical representations of a discrete dynamical system.

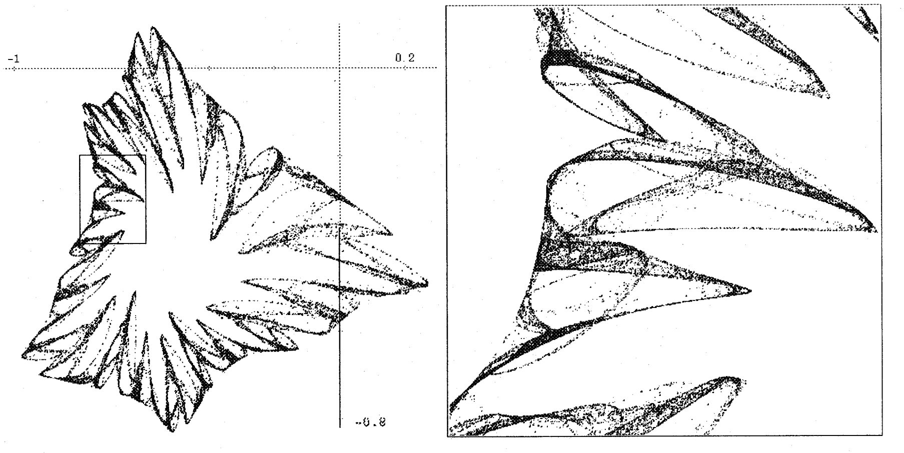


**Fig. 3.** The orbit of a two-dimensional discrete system giving rise to a fractal, and an enlargement of a portion of the fractal.

Figure 3 shows the graph obtained for a second-order discrete dynamical system, using the program `evolution2d`. That program allows the user to enlarge a portion of the graph.

Other functions that we have developed for Maxima plot the orbits of a stochastic dynamical system, using the so-called "chaos game" or using Iterated Function Systems (IFS). We also have functions to draw the Julia and Mandelbrot sets (Figure 4). All of these programs are distributed with a GPL free software license, and they can be obtained from: http://fisica.fe.up.pt/maxima/

Programming additional modules for Maxima can be made using its own programming language or in Lisp, which is the language in which Maxima is written. A good reference for programming in Lisp for Maxima is the book by Norvig [4].

## 4 Results and Conclusions

We have been teaching the course on dynamical systems in our Engineering School since the Fall of 2003. The course replaced a course on Mechanics for Engineering. The course on Mechanics, as well as most courses on Physics and

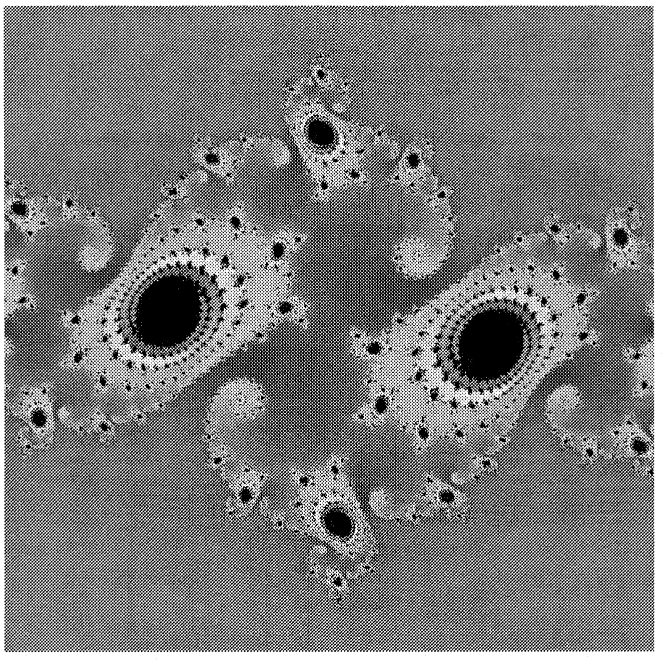

**Fig. 4.** A graphical representation of a Julia set.

Mathematics taught at our School, used to have very low approval rates. During the years that we were involved with the teaching of the Mechanics course, the approval rate was close to 50%.

The year when we introduced the new course on Dynamical Systems, the approval rate increased dramatically to over 90%. That improvement is due in part to the use of a more active teaching methodology and to the change in evaluation method, using a term paper instead of a final exam. However, it is very clear that the students motivation has increased and they are much more interested in the modern topics included with the new course.

The feedback from the students has been very positive and the results from yearly student surveys conducted at our School show that they are very satisfied with this new course.

Computing tools have proved to be a very valuable aid for teaching the subject of dynamical systems. Using CAS tools some tasks are much easier for the students, but that does not mean they have to work less: by lowering the initial barriers, students move deeper into more advanced topics which are also intellectually very demanding. What happens is that the reasoning expected from the students has shifted from mastering algorithms and analytical methods, into getting a better understanding on how a system evolves, using a geometrical analysis.

## References

1. Kallaher, M. J., editor: Revolutions in Differential Equations. Exploring ODEs with Modern Technology, The Mathematical Association of America, ISBN 0-88385-160-1, U.S.A. (1990)
2. Redfern, D., Chandler, E. & Fell, R. N.: Macsyma ODE Lab Book. Jones and Bartlett Publishers, U.S.A. (1997)
3. Homepage of the Maxima Project: http://maxima.sourceforge.net
4. Norvig, P.: Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp. Morgan Kaufmann publishers Inc., U.S.A. (1992)