MATLAB IN THE ENGINEERING CURRICULUM

Alan Davies and Stephen Kane
Department of Physics, Astronomy and Mathematics,
University of Hertfordshire, Hatfield, AL10 9AB, UK.
{a.j.davies, s.j.kane}@herts.ac.uk

Abstract
In the University of Hertfordshire we use *MATLAB* in the mathematics curriculum for teaching engineering students. A University-funded teaching and learning initiative has allowed us to develop a set of web-based workshops. These materials have been written for first and second year engineering students. However, they have also been used successfully with mathematics, physics and astronomy students. We describe the resources and show how the workshops are embedded in our teaching.

Introduction
Over the past ten years or so, the mathematical ability of the majority of entrants to engineering degree programmes at the University of Hertfordshire has, in common with most other UK higher education institutions, decreased significantly. The reasons are varied and are not straightforward and do not concern us here, we are more concerned with how we cope with this reduction in ability and at the same time keep the more able students interested. There is a significant amount of literature describing the use of computer algebra packages in the undergraduate curriculum. Some of the early contributions include Davies (1990), who describes the use of *REDUCE* for civil engineers, Matthews *et al.* (1996) who use *Maple* as a programming language, Edwards and Planitz (1990) who describe a *MATLAB* application, Robin (1990) who applies *Derive* to the analytic solution of ordinary differential equations and Malek-Madani (1996) who uses *Mathematica* for the solution of problems in engineering mathematics.
Since these early days there have been many publications describing the use of a variety of computer algebra packages but they are all very similar to the publications just mentioned. While the material hasn't changed to any great extent, other than our seeing more sophisticated applications packages, the mode of delivery has changed significantly. In the late 1980's students used software on mainframe computers running in compile-link-run process. Stand-alone PC's together with interactive software allowed a big jump forward in the early 1990's and networked PC's became available in the mid 1990's. The early twenty-first century saw the widespread use of the internet and web-based materials now dominate the scene. See, for example, the proceedings of the series of international conferences ICTCM (Goodell 1996-2002).

We choose to use *MATLAB* for two reasons:
(i) It is an industry standard so we are giving our students valuable experience in the use of a software applications package that they may well use in their later employment.
(ii) The many toolbox add-ons, as well as the simulation package *Simulink*, are available to them in other areas of their engineering education.

So, what is *MATLAB* and what can it do?

- *MATLAB* is an interactive system for doing numerical computations.
- It has a very powerful symbolic computation facility.
- A numerical analyst called Cleve Moler wrote the first version of *MATLAB* in the 1970s. It has since evolved into a successful commercial software package.
- *MATLAB* relieves you of a lot of the mundane tasks associated with solving problems numerically. This allows you to spend more time thinking, and encourages you to experiment.
- *MATLAB* makes use of highly respected algorithms and hence you can be confident about your results.
- Powerful operations can be performed using just one or two commands.
- You can build up your own set of functions for a particular application.
- Excellent graphics facilities are available and the pictures can be inserted into *Word* documents with very little difficulty.

All students study mathematics for three semesters and *MATLAB* is embedded in these three modules. The delivery is via the University's managed learning environment *StudyNet*.

MATLAB materials

A University-funded learning and teaching initiative has allowed us to develop a web-based set of information files and associated workshops. Students can log-on to *StudyNet* either from within the University or from outside and use the information files in the same way as they would use a text-book and the workshops have been developed so that they can practice the topics in the associated information file. The information files and workbooks are listed in the appendix. In each of the modules students have coursework exercises which require them to use both hand calculation and *MATLAB*. A typical coursework exercise is given in the appendix.
The information files clearly form an important reference resource for students but we believe that it is the work that they do in the workshops which gives them the experience and confidence with those topics. Each workshop is arranged in two sections, A and B. Section A contains enough material for a forty-five minute laboratory session and covers that basic material found in the associated information file. Explicit references to the relevant material are given in each exercise. Satisfactory completion of section A should ensure that students have sufficient experience of that material to move on. The material in section B gives the opportunity to reinforce the ideas of section A and encourage students to work outside the timetabled laboratory sessions.
We encourage students to set up their monitor screen with three separate windows.

One is for the relevant information file, which is reduced and accessed from the lower toolbar. The other two are left on screen one for the workshop and one for the *MATLAB* command window as shown in Figure 1.

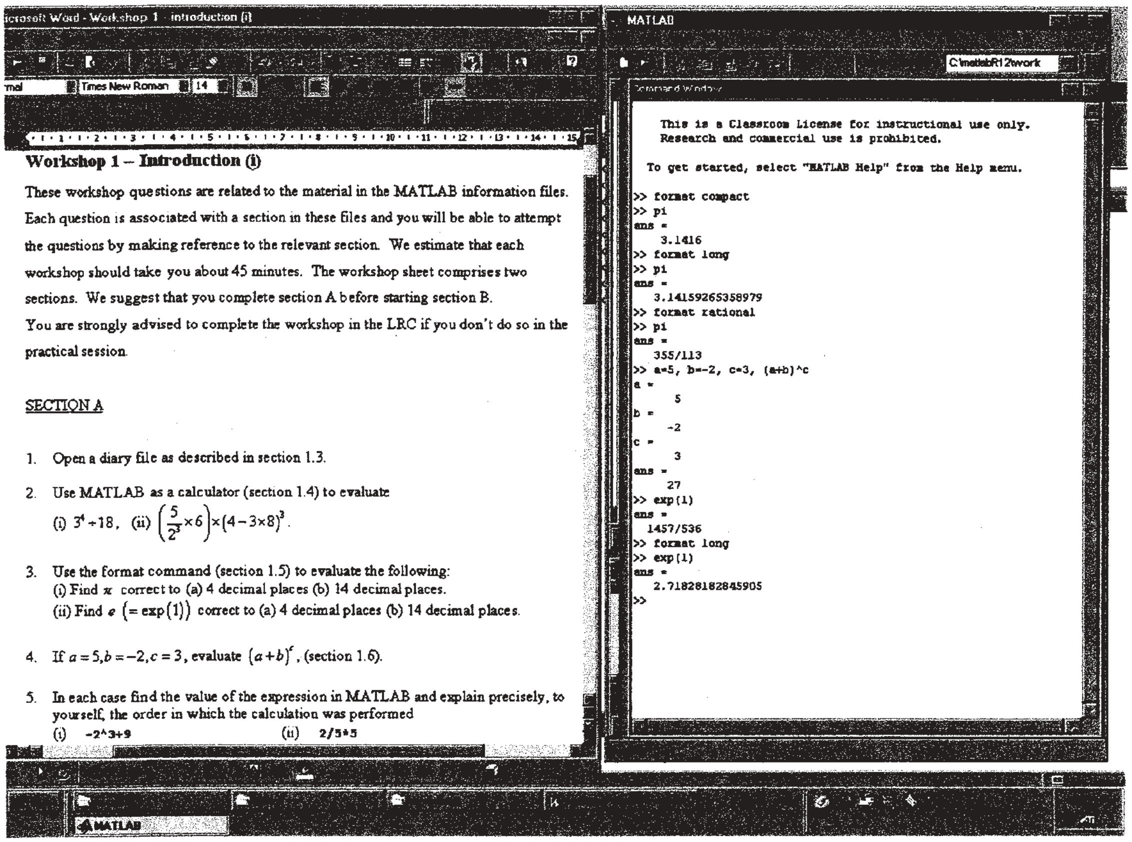

Figure 1, Screen layout for use of the information files, workshops and *MATLAB*

It is interesting to note that while students' mathematical ability is less than it used to be they almost all have very good IT skills and it takes only about fifteen minutes for them to learn how to set up their screen and get into *MATLAB* within the *StudyNet* environment. Subsequently they are able to get going in just a couple of minutes.

How MATLAB is used

Our engineering students have a four-hour per week teaching schedule which comprises three lectures and an alternating tutorial/computer practical sessions. The lectures and tutorials are run in a traditional manner and the practicals are used to back-up the students' mathematics learning using *MATLAB*. Coursework assignments are set so that both hand calculation and *MATLAB* solutions of the problem are required. A typical assignment is shown in the appendix where we see that students solve a second order differential equation in four ways: (i) by hand, (ii) using *MATLAB's* analytical differential equation facility, (iii) using *MATLAB's* numerical differential equation facility and (iv) using Simulink.

We have found that students engage in the process with significantly more enthusiasm than previously. Just as important, we have found that the numbers of students passing the modules have increased significantly. Now part of the reason for this may well be that our assessment strategy has changed but we also believe that the learning activities associated with using *MATLAB* have contributed to students' improved understanding.
All the materials are available from the authors.
The vast majority of publications associated with the use of symbolic computation packages have considered only how the software may be used. Pedagogical aspects are usually not considered. Davies and Fitzharris (1995) describe some of the important issues. Our use of *MATLAB* has certainly addressed some of the problems that we face in our own teaching and we hope that others will wish to follow suit.

References

Davies AJ (1990) Some reflections on the use of a computer algebra package with Civil Engineering students, *Mathematics in a Changing Culture*, 1A1.1-1A1.4, Glasgow College.

Davies AJ and Fitzharris AM (1995) *DERIVE* in undergraduate courses for mathematicians and engineers, *International. DERIVE Journal*, **2**, 24-32.

Edwards D and Planitz M (1990) MATLAB – more than a matrix laboratory, *Mathematics in a Changing Culture*, 1A11.1-1A11.4, Glasgow College.

Goodell G (1996-2002) *Seven Proceedings of the International Conferences on Technology in Collegiate Mathematics*, Addison-Wesley.

Malek-Madani (1996) Engineering mathematics on *Mathematica*, *Proc. ICTCM 7*, Addison Wesley.

Matthews D, Narayan, Schwingendorf K (1996) Maple V as a mathematical programming language in calculus,

Robin WA (1990) On the exact solution of linear ordinary differential equations and its implementation on a computer algebra system, *Mathematics in a Changing Culture*, 2B11.1-2B11.15, Glasgow College.

Appendix

<u>MATLAB information files</u>

MATLAB 1 Introduction
MATLAB 2 Plotting graphs
MATLAB 3 m-files
MATLAB 4 Vectors
MATLAB 5 Matrices
MATLAB 6 Calculus
MATLAB 7 Programming
MATLAB 8 Further MATLAB
MATLAB 9 Simulink
MATLAB 10 Fourier series
MATLAB Appendix

<u>Workshops</u>

Workshop 1 – Introduction (i)
Workshop 2 – Introduction (ii)
Workshop 3 – Plotting graphs

Workshop 4 – Differentiation
Workshop 5 – Integration and limits
Workshop 6 – Plotting surfaces
Workshop 7 – Differential Equations
Workshop 8 – Simulink

Typical first year coursework exercise

This piece of work is associated with the solution of the second order differential equation

$\ddot{x}+4\dot{x}+5x=2\cos(3t)$

subject to the initial conditions

$x(0)=1,\ \dot{x}(0)=\frac{1}{5}$

1. By finding the complementary function and the particular integral obtain the general solution. Hence find the specific solution that satisfies the initial conditions. What is the amplitude of the steady-state term?

[20 marks]

2. Use MATLAB's differential equation solver facility to obtain the specific solution and compare it with the result obtained in part 1.
Use MATLB's plotting facility to plot the solution for $0 \le t \le 6\pi$.
Now use MATLAB's numerical differential equation solver, `ode45`, to obtain the specific solution. By plotting the numerical solution compare it with the analytic solution for $0 \le t \le 6\pi$. Compare the amplitude of the steady-state in the plot with that in part 1.
Hint: You will need to write a short m-file to store the equation written as a pair of first order equations.

[10 marks]

3. Build a SIMULINK model for the problem and use it to obtain a plot of the solution for $0 \le t \le 6\pi$.
Compare the plot with that in part 2.

[10 marks]

Your solution must include the following:
For part 1. the derivation of the complementary function and the particular integral together with the evaluation of the specific solution.
For part 2. the MATLAB commands, including the m-file, and a copy of the plot.
For part 3. The Simulink model and the output plot together with details of the information supplied in the block parameter boxes.