An innovative learning model for computation in first year mathematics

Birgit Loch
Department of Mathematics and Computing, USQ

Elliot Tonkes
CS Energy, Brisbane

Antony Stace
Department of Mathematics, UQ
Project aim

- Build computational components into the first year curriculum
- Integrate the components into the course (not simply a course add-on)
- Must use Matlab
Context of the project

- Implemented into two first year courses with huge enrolments
  - 800 students each per year
  - 50 lab tutorials per year
  - Familiar issues of student diversity:
    - Diverse backgrounds
    - Diverse intended courses of study
  - Approximately one module a week
Courses cover a traditional curriculum
- Calculus of one variable
- Calculus of several variables
- Ordinary differential equations
- Linear algebra
Interpretation of the project

- What does “computational components” mean?
  - perform administrative tasks such as distribution of information or online quizzes
  - convey classical ideas in a more interactive way than a text
  - provide an introduction to numerical mathematics
  - provide an introduction to scientific computation
  - provide an introduction to programming techniques
Historical Experiences

- Both Maple and Matlab have been used
  - Student attitudes have been negative towards computational components: survey results and poor attendance
- Issues identified
  - Students have an initial hurdle in learning the syntax of the software
  - Students have difficulty seeing the direct relevance to the rest of the course
Solutions discovered

- No magic bullet
- Several innovations over several years
- Combination of techniques provided a positive experience for students
Key Components of the Learning Model

- Demonstrations
- Interactive workbook/webpage
- Prepared GUIs
- Prewritten programs
- Linking with lecture material
Demonstrations

- Strongly demanded by students in previous semesters
- Two weeks – on-screen demonstrations to explain how to negotiate Matlab
- Provides confidence
- Overcomes initial syntax problems
- Encourages attendance
The workbook/webpage

- Has now become a standard teaching methodology (large first and second year service courses)
- Web logs showed the hints and solutions were used a great deal
A Module Example

- Students receive the printed module when entering the class. They are encouraged to write notes on it as they go through the module.
- We now go through a module, hints and solutions
For this module, you should enter the lab, and sit down at a machine. You should fill in the blank spaces in your workbook. You will also have some Matlab questions on your weekly tutorial sheet, and you should use Lab time to complete these. If you miss the lab, you can work on it in your own time, using the hints, tips, tricks and solutions page. You should also look at the assignment, which will have some Matlab components.
Example: Construct two vectors with 101 elements in each:

\[ x = [0, 0.1, 0.2, \ldots, 10] \] and \[ y = [\sin(0), \sin(0.1), \sin(0.2), \ldots, \sin(10)] \].

Plot \( y \) against \( x \) using these vectors.

We will use a loop to do this. The code is written below, and some explanation is at the Hints and Solutions pages.

```matlab
% Matlab Program to make vectors x and y, then plot one
% against the other
for i=1:101
    x(i) = (i-1)*0.1;
    y(i) = sin(x(i));
end
plot(x,y)
```

Exercise: Create a matrix which displays

\[
\begin{bmatrix}
0 & \sin(0) \\
1 & \sin(1) \\
2 & \sin(2) \\
\vdots & \vdots \\
100 & \sin(100)
\end{bmatrix}
\]
**H5:** Make sure you follow the code carefully and understand it. This small program contains a loop, vectors x and y, and calculations on the elements in the vectors. Finally, the two vectors are plotted to give the figure shown here.

**S6:** This matrix is a 101-by-2 matrix (101 down, 2 across). Matlab does not display the brackets, so you should get a useful way of obtaining a table of x against \( f(x) \). [Click here](#) to download some code to perform this.
Sample implementation:

Matrices as transformations

- Inbuilt Matlab demonstration GUI
  “makevase”
Sample implementation:
Matrices as transformations

- Modified to mapping a vector under matrix transformations:
Sample implementation:

Matrices as transformations

- **Student investigations:**
  - Derive inverses
  - Motivate eigenvectors/values
  - Interpret matrix actions
Sample implementation:

Taylor Polynomials

- Difficult concept to convey in lectures:
  - Summing functions together
  - Radius of convergence
Sample implementation:

**Monte Carlo Integration**

- Trick is not to calculate the entire answer
- Fill in the blanks
- Students must work out the rest
Sample implementation:

Sequences and Series

- An example of introducing programming techniques (loops)
- Convergence a difficult concept to convey in traditional lectures
## Student Feedback

<table>
<thead>
<tr>
<th>Question</th>
<th>Pre 2002</th>
<th>2002</th>
<th>2003</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of responses</strong></td>
<td>398</td>
<td>254</td>
<td>67</td>
</tr>
<tr>
<td><strong>1</strong> There was enough introductory material to help me learn the computer packages</td>
<td>3.3</td>
<td>3.5</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>2</strong> The computer assignments helped me understand the course</td>
<td>3.7</td>
<td>3.7</td>
<td>2.1</td>
</tr>
<tr>
<td><strong>3</strong> There was enough help available with computer problems</td>
<td>2.9</td>
<td>3.4</td>
<td>2.2</td>
</tr>
<tr>
<td><strong>4</strong> The computer assignments were the most interesting part of the course</td>
<td>4.2</td>
<td>4.1</td>
<td>2.4</td>
</tr>
<tr>
<td><strong>5</strong> I prefer to work alone rather than working in a pair</td>
<td>NA</td>
<td>NA</td>
<td>2.74</td>
</tr>
<tr>
<td><strong>6</strong> The interfaces are easy to use</td>
<td>NA</td>
<td>NA</td>
<td>2.32</td>
</tr>
<tr>
<td><strong>7</strong> The Matlab code is easy to understand and adapt</td>
<td>NA</td>
<td>NA</td>
<td>2.80</td>
</tr>
</tbody>
</table>
Tutor Feedback

- Students asked more difficult questions and displayed a deeper understanding of the mathematics, rather than just the Matlab syntax.
- Tutors were not run off their feet by repeatedly answering the same question since the hints and solutions provided that assistance.
- Student retention was improved (with students even coming to multiple classes).
Other Observations

- Lecturer must show relevance of Matlab during lectures. Even little remarks like “this graph was done in Matlab”
- The modules eliminated many of the same basic questions, tutors spend their time answering more difficult questions
- Students need to see link between lecture material and Matlab classes, and also relevance of Matlab for future subjects and jobs
Conclusions & Further Work

- Developed a learning model that works well in practice, from the lecturer’s, student’s and tutor’s perspectives.
- Important to stress importance of Matlab, show relevance to the use of Matlab to future studies and careers.
- The learning model can be further improved with more integration of lecture and laboratory material.
- Compared to previous teaching model, students have a better Matlab ability at the end of course.