

# STUDENT CENTRED LEARNING IN A MATRIX ALGEBRA COURSE

**John CURRAN**

University of Otago  
PO Box 56, Dunedin  
New Zealand

e-mail: [jcurran@maths.otago.ac.nz](mailto:jcurran@maths.otago.ac.nz)

## ABSTRACT

A new approach was tried in presenting a matrix algebra course to students with differing abilities and diverse needs. The course had previously been presented using fairly traditional methods where the emphasis was on the transmission of knowledge from the lecturer to the students (using partially completed notes which students filled out during lectures). While exam results were reasonable, based on well practised examples, the course was fairly narrow and prescribed.

A change to a more student centred approach was effected using the following mix.

(1) The lectures paralleled a text book (Linear Algebra with Applications by David Lay), which closely followed the Linear Algebra Curriculum Study Group recommendations for an appropriate core syllabus responsive to client needs, and using a matrix-oriented problem solving approach. The resulting lectures were reasonably informal, encouraged students to read the text for themselves, attempted interaction and incorporated some technology.

(2) There was a weekly computer laboratory session using Matlab, where students were encouraged to work cooperatively in pairs on problems from the databank of Matlab exercises available with the text, and on other projects that emphasised understanding and demanded written interpretation.

(3) There was a weekly individual exercise, which provided a variety of question types from routine computations and standard algorithms to short proofs that required an understanding of key concepts. The paper considers such questions as how successful the course was from the viewpoint of lecturer and student, how Matlab was used not only for calculating more realistic examples but to aid understanding, and how well the group approach worked.

## Background

Admission to university in New Zealand is fairly open. School leavers require a modest performance in their final national school exams, while those aged 20 or over are granted automatic entry. As in other countries, there has been a large increase in the numbers of students attending university in recent years, with Otago's role increasing by some 50% in the past 10 - 15 years. This means the mathematical background and ability of entry students is very mixed. As well, students now pay sizeable fees (from some 2000 Euro for science to more than 5000 Euro for health science), so although there is a Government loan scheme, there is considerable pressure to have reasonable pass rates in courses.

The matrix algebra course is a 200 level one semester paper usually required by mathematics majors. Its prerequisite is two semesters of 100 level mathematics, which combine algebra and calculus and are fairly traditionally taught large classes using little or no technology. Topics taught in the algebra section include introductory material on 3-D vector algebra, matrices, determinants, 2-D linear transformations and eigenvectors.

The University enrolls students in arts, science, business, health sciences and other disciplines (but has no engineering school), and the matrix algebra course caters to students from all of these areas, although predominantly science. For about half the students it is their only mathematics paper. But there are also some honours students from the likes of mathematics, statistics, physics and computer science.

The previous course used "outline notes", that is a bound set of partially completed notes in which students copied down examples, diagrams and arguments during lectures. Some problems using Matlab were set on the exercise sheets (from Kolman 1997). On the face of it students learned well, but I became concerned that the course was too prescribed and the students too "spoon-fed". When all mathematics papers were changed in 2000, I decided to try a more "student centred approach", that is making the students (where possible in collaboration with others) more responsible for their learning, and changing the lecturing role to more that of a facilitator, as advocated in Berry et al 1999. At the same time I introduced technology as an integral part of the course.

## The Course

In designing any mathematics course at Otago, there is a certain tension between providing a challenging course but being aware of competing courses which often make fewer time demands. Even the other 200 level mathematics courses have only a one hour (come if you need help type) tutorial. After due consideration the following mix was chosen.

(1) The lectures (32) paralleled a text book (Lay 2000), the author being a member of the Linear Algebra Curriculum Study Group, who have put out a recommended curriculum for a first course in linear algebra (Carlson et al 1993) which this text follows. This syllabus, widely debated and generally acclaimed (see Dubinsky 1997 for a contrary view), worked well with the diverse students in my course.

Because of the sort of conceptual difficulties discussed in Dorier & Sierpiska 2001, the chapter on general vector spaces was avoided in favour of the more concrete approach of treating all vectors as  $n$ -tuples in  $\mathbb{R}^n$ . The emphasis placed by Lay on considering the columns of a matrix aided understanding. For example the definition of  $Ax$  as a linear combination of the columns of  $A$

with weights from the vector  $\mathbf{x}$  eased the initial problems students have in reformulating between vector equations and matrix equations. The early introduction of key concepts such as linear independence and spanning (together with the computer work) also seemed to give students a much better and more confident grasp of these often troublesome ideas. The geometrical focus helped students grapple with concepts, from the more abstract notion of a subspace (which still takes time) to the more concrete ideas involving least squares.

The students received in advance, chapter by chapter, a one page summary of each section we covered in the book, giving key definitions, results and Matlab commands needed. So in theory, if not in practice, students could read ahead and prepare for lectures. The lectures themselves were reasonably informal, highlighting the principal issues in each section and leaving the students to work through much of the detail, although I still probably explained more than I should. The first year I gave no notes at all but just talked, and tried with varied success to get class participation (somewhat difficult with 90 students). The second time I gave some informal notes (due to feedback from the first year, possibly influenced by comparison with the companion calculus paper where all notes were given by hand), such as quick summaries, problems to watch for, or perhaps considering part of a proof.

Lay's text comes with a database of all the book's exercises in whatever CAS is used (we used Matlab). By typing the simple command `cisj`, Matlab prompts which question you want from chapter *i*, section *j*. This made it easy to consider any question from the text in lectures.

(2) Students attended a weekly two hour Matlab session, using problems from the text to practice standard algorithms and computations (before doing their individual exercises) and mini-projects from various sources, such as the CD or Matlab manual (Day 2000) accompanying Lay, or suggested in the MAA notes (Carlson et al 1997).

Matlab was selected because the command structure is straightforward (no programming was required). In the first lab before lectures began, students were given a handout and asked to get to grips with entering matrices and doing standard matrix operations. From then on students were asked to work in groups (most chose pairs because of the lab layout), jointly writing up their work and sharing the mark, which counted towards their internal assessment. Because of their algebra knowledge from 100 level mathematics, students started the course applying new technology to old mathematics (following the rule of thumb suggested by Berry et al 1999) and coped easily, perhaps occasionally needing a reminder about syntax (which was usually supplied by another student!).

There is evidence that using technology can develop understanding (see Mayes 1996). The following sample of examples we considered convinced me of this.

**(a) Realistic examples of linear equations.**

After looking at smaller examples, linear traffic flow problems or temperature grids involving perhaps 20 variables can be easily tackled. Homogeneous systems such as balancing the chemical reaction



also give concrete examples of what a vector  $n$ -tuple might represent.

**(b) Visualizing linear transformations.**

The M-file `planelt` (which comes in a package with Kolman 1997) nicely demonstrates the effects of 2-D transformations. I ask students to take a shape and observe (sketch) how it changes under a succession of transformations and then find a single matrix with the same net effect (and check that it works – they quickly learn to compose matrices in the right order!) A similar M-file `drawpoly` (from Day 2000) can be used to illustrate affine transformations in the

plane in the section on computer graphics, using homogeneous coordinates and  $3 \times 3$  matrices. For example to rotate a figure about an arbitrary point.

**(c) Standard algorithms.**

The computer allows students to concentrate on the algorithm rather than the arithmetic, and is useful for practising routine calculations such as LU factorizations or diagonalizing matrices. Further, they can easily check their result works which may iron out any misconceptions, for example with regard to the order in which the eigenvalues are placed in the diagonal matrix.

**(d) Computer insights.**

Counting the number of operations (flops) used in a task is instructive, for example in calculating  $(A*A)*x$  or  $A*(A*x)$ , which brings the associative rule to life. Considering rounding errors also gives insights. For example students found by chance a simple integral matrix for which  $\det(A) = 0$  but  $\text{rref}(A) = I$  (row equivalent to  $I$ ). Then using the `rrefmovie` command they could observe how a pivot could be small but not small enough to be set to 0, unless the default value for the tolerance (numbers smaller than this value are set to 0) was increased. The effect of partial pivoting can also be explored. For example, students could use this technique to produce the same LU factorization calculated by Matlab (which differs from the usual hand calculation because of partial pivoting).

**(e) Experiment and prove situations.**

The computer can be quickly used to show patterns leading to conjecture and (possibly) proof. For example, the behaviour of triangular matrices (used in LU decompositions and other applications) is explored in many texts. Students get bored pretty quickly using random matrices to observe that like triangular matrices are closed under products (although a picture proof is instructive), but guessing the form of the inverse of a  $3 \times 3$  (or  $4 \times 4$ ) triangular matrix with integral entries (using rational format) usually requires a good number of repetitions to spot the pattern and the consequent conjecture can then be proved.

**(f) Iterative processes.**

Finding the steady state vector of a stochastic matrix or iterative solutions to linear systems using the Gauss or Jacobi method are ideal for CAS.

**(g) Eigenvectors.**

Lay introduces eigenvectors by considering a dynamical system involving the three life stages (junior, subadult, adult) of an owl population. Matlab enables these populations to be quickly modelled and graphed simultaneously. Students can then get a real feel for how the populations behave as the survival parameter (junior to subadult) varies. Later (after eigenvector bases have been explored) they can explore how the eigenvalues of the associated matrices change with this parameter. Matlab can also be used to plot the iterates of a point under the action of a matrix (so for example the trajectory of a dynamical system) and observe how these vary according to the eigenvalues.

**(3)** There was a weekly exercise that students worked on individually, with a variety of questions (mostly from Lay) such as routine calculations, standard algorithms or short proofs. The well designed questions from Lay require little computation, largely avoided being repetitive (often asking the same underlying question in different ways), and tested knowledge in quick but searching ways.

## Conclusions

Delegating more responsibility to students for their learning, provided more time in lectures to

stress the main results (using lots of transparencies), to give informal and intuitive meanings to concepts and to discuss pedagogical issues. However there was still a conflict between covering the material (for those who might use the linear algebra) and spending more time understanding it (for those for which the course was a vehicle for learning some mathematics). Using Matlab helped break up the lectures and generate discussion, but further class activities would help.

A fairly detailed student survey of the course was conducted and some of the responses are recorded in table 1. As can be seen, generally the students found the pace appropriate and the book easy to follow. Surprisingly (see table), they also preferred the lecture approach taken to the more secure outline notes which they had used in 100 level courses and which they rated favourably there.

	Strongly agree	agree	neutral	disagree	strongly disagree
Pace of material appropriate	30%	45%	22%	3%	0%
Book easy to follow	28%	55%	11%	6%	0%
New approach better than outline notes	19%	48%	26%	6%	0%
Matlab enhanced understanding	13%	57%	9%	21%	0%
Group work enhanced understanding	30%	33%	23%	10%	3%

**Table 1**

From my perspective the group activity in tutorials worked really well and students generally agreed with this. They also found Matlab helped their understanding although not all agreed (see table). There was a much better participation rate in tutorials and a more vibrant (nosier!) atmosphere. Various problems sometimes associated with group activities, such as inactive members or subdividing material, were largely avoided since the students worked mainly in pairs sharing one computer (largely dictated by the lab layout). Usually the longer the pairs worked together the better the collaboration, but I didn't force this and for various reasons there were realignments or the occasional person who wished to work alone. In most partnerships, even when one was mathematically weaker than the other, both were able to contribute in various ways (perhaps one might be more computer savvy or a better recorder than the other). There was the odd instance of a very lopsided liaison in which the weaker student was considerably helped by their partner.

I tried to encourage the groups to record a clear description of the object and outcome of each exercise as well as the mathematics involved, but success here was mixed (sometimes lots of numbers were recorded but not the big picture). The marking scheme tried to reward good explanations, but because not all parts were able to be marked the scores awarded were not always very discriminating and were generally quite high, but this had a good attitudinal spin off. I would probably design a lab sheet in future to encourage better explanations. The University now regularly conducts surveys of student opinion in all courses and mathematics performs well in areas such as "developing problem solving skills" but poorly in "written communication skills" and "ability to work as a team member". The computer labs should help address these concerns.

The individual exercise was more challenging and discriminating. Answers to odd numbered questions are given in Lay, so students could try an adjoining exercise, which although different might have some similarities. Of course more routine hand calculations were often checked by students using Matlab. Student's lack of experience in tackling proof type questions was obvious and even with lots of help (in office hours) success was mixed.

In general I was happy with the course, and felt it was a step in the right direction away from a passive lecturing style to a more active student involvement.

#### REFERENCES

- Berry, J., McIntyre, P., Nyman, M., 1999, "Student centred learning in undergraduate mathematics" in Proceedings of the Delta '99 Symposium on Undergraduate Mathematics.
- Carlson, D., Johnson, C.R., Lay, D.C., Porter, A.D., 1993, "The Linear Algebra Curriculum Study Group Recommendations for a First Course in Linear Algebra", College Math. J. 24, 41-46.
- Carlson, D., Johnson, C.R., Lay, D.C., Porter, A.D., Watkins, A., Watkins, W., 1997, *Resources for teaching linear algebra*, Mathematical Association of America notes vol 42.
- Day, J.M., 2000, *Instructors Matlab manual*, Addison-Wesley.
- Dorier J.L., Sierpiska, A., 2001, "Research into the Teaching and Learning of Linear Algebra" in *The Teaching and Learning of Mathematics at University Level*. ICMI Study series vol 7, Kluwer Academic.
- Dubinsky, E., 1997, "Some thoughts on a First Course in Linear Algebra at the College Level" in MAA notes vol 42.
- Kolman, B., 1997, *Introductory linear algebra* (6<sup>th</sup> edition), Prentice-Hall.
- Lay, David C., 2000, *Linear algebra and its applications* (2<sup>nd</sup> edition), Addison-Wesley.
- Mayes, R.L., 1996, "Current State of Research in CAS in Mathematics Education" in Berry, J., Monaghan, J., Kronfellner, M., Kutzler, B., *The State of Computer Algebra in Mathematics Education*, Chartwell-Bratt, 171-180.