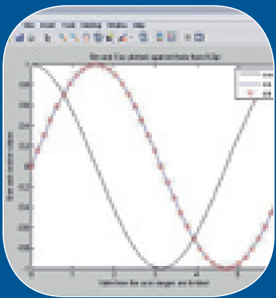
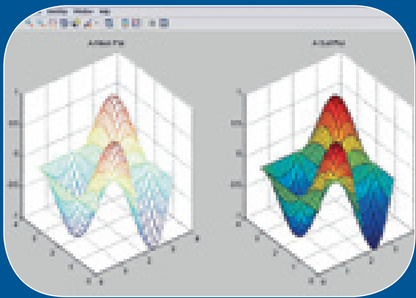
Automated Assessment using MATLAB

Professor Harry Tolley, Professor Jon Huntley
and Dr Adam Crawford





engCETL

Loughborough University

Introduction

The focus of this case study is on the development of an automated system for the assessment of students' MATLAB scripts through a process of collaboration between engCETL and Professor Jon Huntley, of Loughborough University's Wolfson School of Mechanical and Manufacturing Engineering.

Context

The *Engineering Computation* module is taught to all undergraduates studying Mechanical Engineering at Loughborough as well as to some (typically 5-15) physics students. The aims of the module are to improve the students' computer literacy and develop their ability to solve 'real' engineering problems using numerical methods. The module is taught by means of 16 lectures supplemented by 20 hours of computer-based tutorials. Assessment involves both examinations and coursework, the latter consisting of a series of assignments that require the students (working individually) to develop programs to solve engineering problems written in MATLAB – the numerical analysis software chosen for the purpose.

Problems

In 1998, Professor Huntley approached the then Faculty of Engineering Teaching and Learning Support Centre (EngTLSC) with the following problem related to his teaching of *Engineering Computation: "In previous years this module has taken over 240+ contact hours and last year was the second most expensive second year course to teach in Mechanical Engineering"*. The main reasons why the module was so labour-intensive and costly were:

1. each tutorial was taught four times over, and by two members of staff, due to the numbers of students on the course and the size of the computer rooms;
2. as part of their coursework assessment for this module, all of the students (typically between 100 and 160 each year) had to produce a series of *"short (but difficult)"* pieces of coursework in the form of Fortran 77 code. The marking of each set of these programs was extremely time consuming – not just as a consequence of the size of the student cohort, but because each individual piece of software had to be marked for its content, accuracy and syntax. This resulted in a large recurrent marking load for the person responsible for the delivery of the module.

Project aims

Professor Huntley worked with a member of EngTLSC (the Computer Aided Learning Officer) to write and submit a proposal for a project aimed at solving the problem, which would be both technically and pedagogically sound. The decision had been taken at this point to switch the programming language from Fortran to MATLAB in view of its 'user friendliness' and graphical display capabilities. The aims of the resultant project were to:

- Provide tutorial support and improved feedback for students on the Engineering Computation Module.
- Produce a MATLAB toolkit that would provide a set of generic tools for the analysis and marking of programming structures within MATLAB.
- Make available improved levels of feedback for future course development.
- Enhance the quality of the students' learning experiences.

It was anticipated that the automated marking of the students' MATLAB functions[1] would reduce staff marking time, whilst enabling individual feedback to be given to students in a much shorter time than was previously possible. These aims, set out at the inception, have guided the development activities undertaken at successive stages in the project.

Key dates

1998 Initial project proposal
1999 Pilot study for one MATLAB tutorial
2000 Updated project proposal: increased to four MATLAB tutorials
2001 Enabled online submission of students' MATLAB tutorial functions
2002 Pre-submission test file for students
2004 Added plagiarism detection and setting of different tutorials annually

Project actions

The first stage of the project began with a pilot exercise to produce automated marking and feedback for one of the tutorials on the module. This required the development of MATLAB code, using the technical expertise available in the Centre, to read the emailed student submissions, and to mark the resulting outputs based on different input parameters. The automated feedback was then collated and made available by email to students, and also to Professor Huntley. The latter quickly scanned the feedback files, to identify common mistakes and problems that had not been detected by the marking software. This information was then used to send additional feedback via email to all students. The outcomes of the piloting exercise were reviewed, first with Professor Huntley and then with students. The latter indicated that they found learning MATLAB difficult, and that the system for submitting their functions was too complicated. The lessons learned from the pilot study re-affirmed the importance of supporting the students' learning experiences by the provision of formative feedback from the assessment of their MATLAB functions and the need to improve the process by which those functions were submitted. Nevertheless, the outcomes were sufficiently positive to support the extension of the toolkit to three additional MATLAB tutorials.

In 2000, a further proposal by Professor Huntley led to the second stage of the project, in which the tools were developed in a generic fashion so as to provide more flexibility in applying them and to make them easier to use. Two EngTLSC staff (the Learning Technology and Distance Learning Co-ordinators) worked with Computing Services to develop a system whereby the students' MATLAB functions were submitted online to a file store. They could then be marked automatically as a group, whilst still providing individual feedback as developed in the pilot study. Individual feedback was then communicated to the students via a webpage and to Professor Huntley on the submissions of the whole group by means of a separate webpage.

From 2001, the module that introduced Year 1 students to computer programming was withdrawn. As a result, Engineering Computation (Year 2) was now the first experience Mechanical Engineering students had of computer programming in their course. It was anticipated that these students would require improved tutorial support if they were to cope with this change in the curriculum. The third stage of the project (again developed in conjunction with the staff in the Centre) was designed to address this need.

[1] Strictly speaking the submissions are MATLAB 'function M-files'. The term 'MATLAB functions' is therefore used to avoid confusion with MATLAB 'script M-files'.

Common mistakes made by students in previous years were identified, and checks were incorporated into the code to identify these automatically – typical examples being order of magnitude errors and transposed matrices. The other developments undertaken at this time included the addition of basic-level online tutorials to the existing MATLAB tutorials, in order to provide learning support for students with no previous experience in computer programming. A facility was also created for students to select and check their work before final submission. This provided them with instant feedback and encouraged them to further their own learning by finding and correcting their own mistakes.

Since the completion of the main development work, small refinements and adjustments have been made in an on-going response to communication between Professor Huntley and the Centre, with a view to further improving the students' learning experiences on the module and to ease the burden of marking their coursework. Notable improvements include the development by the Centre of automated plagiarism detection and the creation by Professor Huntley of variations in the MATLAB tutorials so that they can be changed annually, on a three-year cycle, to prevent students copying from previous years.

Project outcomes and impact

When interviewed, Professor Huntley, (academic responsible for the Engineering Computation module) noted that the students find MATLAB programming difficult and that they *"seem to grudgingly accept that they benefit by being forced to write their own computer programs from scratch"*, adding that *"the feedback they receive is welcomed"*. The standards achieved by the students on the module have been maintained despite the withdrawal of the Year 1 "Introduction to Computing" module and the persistence of the so-called 'Maths problem' (i.e. the difficulty of recruiting students with good grades in A-level Mathematics). A member of the Centre staff most closely associated with the project noted that although *"the students receive more detailed feedback on their coursework submissions"* there has *"not been an increase in student marks or exam scores"*. However, in the circumstances, maintaining standards in this aspect of the curriculum could well be one of the many positive achievements of this project.

Since the full implementation of the developments initiated by the project (i.e. since 2000) the students have received individual feedback on each of their coursework submissions supplemented by group feedback provided by Professor Huntley. The online submission and automated marking system, which has made this possible has also has also led to a significant reduction in the time required to: collect and collate students' coursework; mark individual MATLAB files; analyse submissions from across the group in order to provide collated feedback for the module leader together with direct links to files submitted by the students; as well as to collate and distribute marks and feedback to the students.

In 2007 engCETL received a proposal to develop a similar system for the teaching of MATLAB to Year 1 students in Aeronautical and Automotive Engineering. At the same time the university made the transition to a new virtual learning environment (VLE), which required the redevelopment of the MATLAB code so that it would work with the new online submission system. The project proposal was approved and further development work was undertaken and implemented through the CETL, including new automotive-related tutorials and online submission to the new VLE. The engCETL Panel has also recently approved a project to transfer this approach to Electronic and Electrical Engineering and to develop tutorials specific to this subject area. Generic developments are also planned to update the original system for on-going use by Mechanical Engineering.

Discussion

The practical problems, which this engCETL project has addressed through the development of an automated system for marking students' MATLAB functions, highlight two closely related pedagogic issues: the assessment of students' learning outcomes (especially with regard to the provision of formative feedback) and the importance of supporting student learning when they encounter difficulties with what might be termed 'troublesome knowledge'.

The MATLAB project, as described above, highlights a number of issues relating to the assessment of students' learning outcomes including the multiple purposes which assessment can serve. First, the project addressed the need to assess the students' MATLAB functions for the purposes of providing each student with a valid and reliable mark for the coursework assessment component of the module. Second, it dealt with the assessment of the students' coursework with a view to providing them with feedback, which they could then use formatively to enhance the quality of their future learning. Gipps (1994) referred to assessment for these two distinct purposes as being *"assessment of learning"* and *"assessment for learning"* respectively. The third purpose served by assessment in this instance was to obtain feedback data for use in curriculum evaluation and course improvement, i.e. by assessing the outcomes of the students' learning experiences (in this case their completed MATLAB functions) evidence was obtained, collated and analysed, with a view to informing decisions and actions with regard to the future design and delivery of the module. Each stage of the project therefore was akin to an action research cycle, with the feedback from the experience of one developmental cycle being fed into the following one.

The literature on generic formative assessment (see e.g. Knight, 2001) suggests that in order for it to be effective in enhancing students' learning the feedback needs to be immediate – i.e. they need to receive their feedback shortly after the work has been completed and submitted. In other words whilst it is still fresh in their minds and before they have become pre-occupied with something else. This is especially important in the case of the *Engineering Computation* module in which the students have to complete a series of tasks and submit the resultant MATLAB functions for assessment at regular intervals – satisfactory completion of the course requiring them to overcome a series of assessment hurdles of this kind. Ideally, in circumstances such as these, they receive feedback on each piece of work prior to attempting the next one – the feedback in effect being 'feed forward' in that it helps them to cope better with the next item of coursework. The feedback should also be personal i.e. it should address the specific issues arising from the assessment of each student's work. In other words, it must not be so generalised that the students cannot make the necessary connections between the feedback they have received and the work they submitted. Finally, the feedback must be constructive i.e. it must do more than simply identify errors and weaknesses, which the student can interpret as being negative and unhelpful. To be useful in furthering learning, it must offer advice and guidance on how the student could have improved her/his performance.

Research would suggest therefore, that in order to enhance the learning experiences of students, those responsible for teaching them need to: provide them with effective feedback, actively involve them in their own learning, quickly adjust their teaching to take account of the outcomes of assessments, recognise the influence that assessment has on the self-esteem and motivation of students, and finally, help the students to assess themselves and to understand what they need to do to improve their own future performance. This is markedly different from situations in which the tutor simply marks the students' work and then provides feedback in the form of grades/marks. To be effective, it entails the adoption of changed pedagogic practices in which assessment is deeply embedded in the interactions that take place between teacher and learners (Shepard, 2000) – as is the case of the MATLAB module under consideration in this case study.

The nature of the coursework tasks in the MATLAB module, the frequency with which they were set and submitted for assessment and the large numbers of students involved meant that before an automated marking system was developed it was a time consuming and labour intensive activity, and there was an inevitable delay between the completion of a coursework task by the students and reception of their feedback. The nature of the module – to teach computer software programming – lent itself to the solution that was adopted. However, the problem of assessing coursework (such as lab reports) submitted by students in large groups is widespread in Higher Education in subjects such as science and engineering. Alternative strategies have been developed elsewhere in the HE sector to address this problem; for a review see Rust (2001).

The original proposal for this development project referred to the pieces of MATLAB code that the students had to write and submit for their coursework assessment as being: *"short (but difficult)"*. This suggests that experience had shown that for whatever reason the students found them to be what is referred to in recent literature on teaching and learning in Higher Education as *"troublesome knowledge"*, which is best explained in terms 'threshold concept theory' (Meyer and Land, 2003 and 2005; Land, Meyer and Smith, 2008). According to these writers, 'thresholds' can be thought of as portals or gateways through which students must pass in order to achieve the learning that they are finding to be difficult – like writing MATLAB functions for the first time.

According to the theory these conceptual gateways encountered by learners are: transformative (i.e. they lead to a significant shift in the learner's perceptions); integrative' (i.e. they reveal previously hidden inter-connections in the student's knowledge and understanding); irreversible (i.e. once learned they are unlikely to be forgotten); and, like writing MATLAB scripts and functions, 'troublesome' (Perkins, 1999). Because students find them difficult to cross, thresholds are the places in the course where the students get 'stuck'.

In providing a framework for the analysis of how individuals are transformed as new learning is experienced, threshold concept theory draws on the social constructivist theories of learning associated with Vygotsky (1973), particularly with regard to learning in the so-called *'Zone of Proximal Development*', and provides a set of ideas that are transferable across subject disciplines. In practical terms it helps teachers in Higher Education to locate troublesome aspects of the knowledge in their disciplines and to assist them in modifying or re-designing curricula to better support their students in negotiating such learning transitions more successfully – as occurred in the case of the MATLAB module through the improvement of the tutorial guidance available to students and the automated provision of individual feedback. It is principally therefore, an analytical framework, which can be used to understand how students learn, where the barriers to their learning can be located and how pedagogy can be adjusted or modified to better support the students' learning experiences. For research on threshold concepts in computer science see Zander et al (2008) and Shinners-Kennedy (2008) and in engineering see Baillie and Johnson (2008) and Carstensen and Bernhard (2008).

Conclusions

The key to the success of the project at each of the above stages was the multi-disciplinary team approach, with the engineering academic and learning technology and educational development staff from the Centre working together to deliver the project's intended outcomes, and in so doing solve the problems it was asked to address. The automated marking of the students' MATLAB functions has not only effected huge savings of staff time and effort, but has enabled feedback derived from the assessment of those functions to be used to enhance student learning and for the purposes of course improvement.

References

Baillie, C. and Johnson, A. (2008). A Threshold Model for Attitudes in First Year Engineering Students. Ch 10 in Land, R., Meyer, J.H.F. and Smith (Eds.) *Threshold Concepts within the Disciplines*. Rotterdam: Sense Publishers.

Carstensen, A-K. and Bernhard, J. (2008). Threshold Concepts and keys to the portal of Understanding Some Examples from Electrical Engineering. Ch 11 in Land, R., Meyer, J.H.F. and Smith (Eds.) *Threshold Concepts within the Disciplines*. Rotterdam: Sense Publishers.

Gipps, C. (1994). *Beyond testing: towards a theory of educational assessment*. London: Falmer Press.

Knight, P. (2001). *Assessment: A Briefing on Key Concepts*. Assessment Series Number 7. York: Learning Teaching Support Network (LTSN) Generic Centre.

Land, R., Meyer, J.H.F. and Smith (2008). (Eds.) *Threshold Concepts within the Disciplines*. Rotterdam: Sense Publishers.

Meyer, J.H.F. and Land, R. (2003). Threshold concepts and troublesome knowledge: Linkages to ways of thinking and practicing within the disciplines. *Improving student learning – ten Years on*, OCSLD, Oxford, 412-424.

Meyer, J.H.F. and Land, R. (2005). Threshold concepts and troublesome knowledge (2): Epistemological considerations and a conceptual framework for teaching and learning. *Higher Education*, 49(3), 373-388.

Perkins, D. (1999). The many faces of constructivism. *Educational Leadership*, 57(3), 6-11.

Rust, C. (2001). *Assessment: A Briefing on Assessment of Large Groups*. Assessment Series Number 12. York: Learning Teaching Support Network (LTSN) Generic Centre.

Shepard, L. (2000). *The role of classroom assessment in teaching and learning*, CSE Technical Report 517, Los Angeles, CA, National Center for Research on Evaluation, Standards, and Student Testing.

Shinners-Kennedy, D. (2008). The Everydayness of Threshold Concepts: 'State' as an Example from Computer Science. Ch 9 in Land, R., Meyer, J.H.F. and Smith (Eds.) *Threshold Concepts within the Disciplines*. Rotterdam: Sense Publishers.

Vigotsky, L. (1973). *Thought and Language*. Cambridge, MA: MIT Press.

Zander, C., Boustedt, J., Eckerdal, A., McCartney R., Mostrom, J.E., Ratcliff, M. and Sanders, K. (2008) Threshold Concepts in Computer Science: A Multi-National Empirical Investigation. Ch 8 in Land, R., Meyer, J.H.F. and Smith (Eds.) *Threshold Concepts within the Disciplines*. Rotterdam: Sense Publishers.

Contact us:

Engineering Centre for Excellence in Teaching and Learning (engCETL)

Keith Green Building Faculty of Engineering
Loughborough University Leicestershire LE11 3TU UK

Tel 01509 227191
Fax 01509 227181
Email engcetl@lboro.ac.uk
Web www.engcetl.ac.uk

ISBN 978 1 907382 12 3