Queen Mary ADEPT Fellowship Scheme

The Account of Professional Practice
Fellow Application

Name	Tomasz Popiel
School/Institute	School of Mathematical Sciences
Date	24/04/17

Evidencing A1: Design and plan learning activities and/or programmes of study

I am a lecturer in QMUL's School of Mathematical Sciences. I organise and lecture the module MTH5110: Introduction to Numerical Computing, which concerns numerical methods for solving various mathematical problems, with a heavy emphasis on the implementation of these methods in the mathematical software package *Maple*. Previously, I lectured a range of mathematics modules across all undergraduate levels at The University of Western Australia.

As evidence for A1, I will discuss how I design and plan the learning activities for MTH5110. This is a challenging module to plan for, being a mathematics module with a particularly heavy computer-programming component. Although it is not unusual for mathematics modules to include 'algorithms' for solving various mathematical problems, the level to which programming in a specific language is emphasised in MTH5110 is perhaps atypical. This is important in terms of the School of Mathematical Sciences' fifth *Objective of Taught Mathematics*, namely that "all graduates will possess basic computational skills" (V4), but poses two challenges in terms of module design/planning, which I address as follows.

The first challenge is to motivate and justify the relevant *mathematical* concepts in sufficient detail so that the students understand how and why the corresponding algorithms work, but not in so much detail as to obscure the intuition behind them or detract from the task of actually implementing them. Therefore, although the algorithms should ideally be justified via rigorous mathematical arguments from the subject of numerical analysis, I have chosen the lecture material in such a way as to treat only the essential mathematical details (K1, K2). Moreover, the mathematical arguments that I *do* include focus on how the solutions to the various problems are actually *constructed*, so that abstract mathematical ideas are primarily used to to justify the correctness of the algorithms, rather than themselves becoming the focus of the lecture material. The second challenge is to keep the computer programs themselves sufficiently straightforward, bearing in mind that the students are, after all, studying mathematics, and do not necessarily have a great deal of programming experience or expertise (V1). To this end, my lecture material introduces only the 'simplest possible' versions of the relevant algorithms. The students are then encouraged to refine these algorithms (e.g. to improve functionality or efficiency) via exercises and/or coursework questions (K2, K3).

These approaches have been informed by subject-specific literature on teaching numerical analysis, e.g. Trefethen, 1992 and Wang, 2004 (V3), and by discussions with colleagues who have previously lectured either MTH5110 or its prerequisite MTH4105 (V4). In particular, these colleagues suggested that the level of mathematical rigour with which MTH5110 had been taught in 2015/16 was too high (based on student feedback and performance), and that it would be better to adopt a more 'balanced' approach in the sense described above.

Judging by feedback collected from approximately 40 students in week 4 of term (February 2017) and peer observation of one of my lectures (K6), I believe that I have designed lecture material appropriate to the above considerations. In particular, several students remarked positively on the pace and difficulty of the lecture material, saying e.g. that the "structure and teaching is great and easy to follow", that the "pace and explanation in lectures is great", and that they are "completely happy with the module". This seemed to be confirmed in the formal module evaluation, where e.g. "the module is well organised and runs smoothly" scored 4.59/5.

Having previously taught mostly pure mathematics modules that did not involve significant amounts of computer programming, I have learned how to design, organise and lecture a programming-intensive mathematics module. In particular, I have learned that students appreciate an approach that focuses on concepts, introduces algorithms with a moderate amount of mathematical detail, and invites them to investigate further via exercises/coursework (K3, V1).

References

L. N. Trefethen, "The definition of numerical analysis", *SIAM News* **25** (1992).

W.-W. Wang, "Student-centred teaching of Numerical Analysis", in: M. Peat and M. King, *The China Papers: Tertiary Science and Mathematics Teaching for the 21st Century*, UniServe Science, Sydney, 2004, pp. 60–63.

Evidencing A2: Teach and/or support learning

Here I discuss my teaching of MTH5110. A typical mathematical problem in this module is to approximate the integral of a function, interpreted as the area under its graph. In lectures, I begin by motivating an intuitively simple solution to this problem, e.g. to approximate the area under the graph by adding up the areas of small rectangles. The effectiveness of this solution can be justified with full mathematical rigour, but I choose instead to convey the essential details by working through an *example* rigorously. This gives the students a feeling for why the method works, without burdening them with too many abstract details (K2). I then introduce corresponding Maple code, carefully explaining each step with reference to the mathematical justification, and then present the code and examples on a computer. Coursework questions reinforce the lecture material, and also prompt students to think about some more complicated issues, in terms of both mathematics and programming (K3). This approach allows students to learn largely via practical implementation and examples, thereby addressing difficulties that students can have when learning programming-based subjects via excessive amounts of theory (Tan *et al*., 2009).

Learning is supported via QMPlus (K4), where I upload typeset lecture notes, accompanying hand-written notes (from the lectures themselves), and supplementary Maple files. Students are strongly encouraged to take hand-written notes during lectures, as this promotes better processing of information compared with using a computer (Mueller and Oppenheimer, 2014). The usefulness of lecture recordings in mathematics has been recognised for some years now (Mullamphy *et al*., 2010), so all lectures are recorded using QReview. In particular, lecture recordings provide students with the flexibility to watch lectures outside of formal class times (V1, V2), and to revisit lectures when revising for exams (K3). Based on discussions with my students, they seem to find them extremely valuable, and I will continue to provide this option in future. Additional reading material is suggested, but *not* necessary for the module; in particular, students are not required to purchase any textbooks, which could be prohibitively expensive depending on their socio-economic backgrounds (V1, V2). This is in line with QMUL's *value* of "supporting the best and brightest of students … regardless of social or economic background". Likewise, the college allows students to download a free copy of Maple.

Based on early feedback, my students were happy with the lecture notes and supplementary material, but requested more learning support via more time being spent presenting code on the computer in lectures, and more 'exam-style' examples. I immediately accommodated both of these requests (K5, K6, V2), and this seemed to be acknowledged in the module evaluation, where "I had access to good learning resources for the module" scored 4.41/5. One significant difficulty that I encountered in this regard was that the lecture rooms that were booked for the module (before I was assigned as organiser) were not computer labs, so I was unable to structure my lectures in such a way that students could follow along on their own computers when I implemented code. Such a student-centred approach is how the level-4 prerequisite MTH4105 is taught, and, based on discussions with the organiser of MTH4105 and other colleagues, I would like in future to make similar arrangements, insofar as the material for MTH5110 allows (K6, V3, V4). Specifically, I think it would be useful to spend roughly two lectures per week on theory in a 'traditional' lecture setting, with the remaining lecture spent in a computer lab implementing code together with the students. Of course, tutorials, which *are* held in a computer lab, go some way to addressing this issue, but I feel it would be more useful to have at least some lectures run as computer lab classes (c.f. Wang, 2004, cited above).

References

P. A. Mueller and D. M. Oppenheimer, "The pen is mightier than the keyboard", *Psychological Science* **25** (2014) 1159–1168.

D. F. Mullamphy, P. J. Higgins, S. R. Belward and L. M. Ward, "To screencast or not to screencast", *ANZIAM Journal* **51** (EMAC2009) (2010) C446–C460.

P.-H. Tan, C.-Y. Ting and S.-W. Ling, "Learning difficulties in programming courses: undergraduate's perspective and perception", *ICCTD '09 Proceedings of the 2009 International Conference on Computer Technology and Development – Volume 01*, IEEE Computer Society, Washington, 2009, pp. 42–46.

Evidencing A3: Assess and give feedback to learners

Assessment for MTH5110 comprises weekly coursework, marked formatively; a mid-term test, marked formatively; and a final exam, marked summatively and worth 100% of the final grade. This initially seemed challenging for me, as I have previously used a 70% exam to 30% in-term assessment split when lecturing in Australia. However, I view it as an opportunity to allow the students to take responsibility for their own learning (c.f. Nicol and Macfarlane-Dick, 2006), encouraging them to consistently engage with coursework while providing them with support via QMPlus, tutorials, and comprehensive individual feedback on their coursework solutions (K3).

Coursework includes both routine calculations and more difficult problems. The more difficult problems typically come in two flavours: students are either asked to take the basic version of an algorithm from lectures and refine it with more advanced functionality, or to think about some mathematical subtlety of an algorithm (K3). This engages students to develop their knowledge of both programming and the underlying mathematics, giving them the ability and confidence to tackle unseen problems, both in the final exam and in their future studies or employment (V2, V4). The tutorials themselves begin with a 'mini-lecture' where I provide some suggestions for how to approach the assignment problems, after which students are free to work on their own or in groups and to ask me for assistance. Solutions are submitted via a Maple file uploaded to QMPlus, and detailed formative feedback is provided for all questions. Maple has various useful features for this purpose, e.g. one can annotate sections of code, making it easy to pinpoint students' errors and suggest improvements (K4). In turn, I solicit feedback from students about the perceived usefulness of the comments provided to them, and adjust my marking accordingly. Generally speaking, students seem to appreciate personalised feedback that addresses specific gaps in their understanding, so I have endeavoured to become better at providing such feedback and will continue to do so (K5, K6). This seems to have been reflected in the module evaluation, where "I have been given adequate feedback during this module" scored 4.35/5. Overall, having previously used mostly summative formal feedback when lecturing in Australia, I have learned more about the value of formative feedback for improving students' understanding of lecture material and motivating them to take responsibility for their own learning.

The test and exam are held in a computer lab and administered via QMPlus (K4). Both are open book, which allows me to test the students' problem-solving skills as opposed to their memory of what are sometimes very subtle mathematical and/or programming details. Here I take a very broad definition of "problem-solving skills", in that it is arguably very useful to be able to quickly find resources to solve problems in an efficient way, whether those resources be my lecture materials, solutions to past exams, or e.g. Google. Indeed, this is a recognised benefit of open book exams, c.f. Gupta, 2007, who also suggests that "open-book exams could lower the anxiety level of students who find reassurance in the available books and other reference materials" (V1). The final exam is yet to take place as of this writing, but I feel that the formative in-term assessments have been generally effective. In particular, a reasonable proportion of students consistently submitted coursework, and those who were most consistent managed to continuously improve their skills. However, in future I would like to have the mid-term test worth a small percentage of the final grade. Based on test performances of some students who do not consistently engage with coursework, I feel this would encourage students to prepare for the test more intensively, thereby consolidating the first half-term's worth of material at an earlier stage in preparation for the final exam (K5, K6).

References

M. S. Gupta, "Open-book examinations for assessing higher cognitive abilities", *IEEE Microwave Magazine* **8** (2007) 46–50.

D. J. Nicol and D. MacFarlane-Dick, "Formative assessment and self-regulated learning: a model and seven principles of good feedback practice", *Studies in Higher Education* **31** (2006) 199–218.

Evidencing A4: Develop effective learning environments and approaches to student support and guidance

I engage with my MTH5110 students in several learning environments, including lectures, tutorials, QMPlus, email, and office hours.

Lectures provide an opportunity to explain the mathematical reasoning behind the Maple computer programs studied in the module. Although the module has a substantial programming component, I feel that 'traditional' lectures also have an important role to play, as a lot of the material is very to easy to explain intuitively using pictures or examples, but considerably more difficult to explain in precise mathematical notation and/or Maple code (e.g. the problem of approximating the integral of a function described in Section A2) (K1). Pictorial explanations in particular seem to be most effectively conveyed in a lecture setting (c.f. Alcock and Inglis, 2010), because students can see a picture being drawn and explained simultaneously, whereas when reading lecture notes they must look at the corresponding picture and read a potentially long explanation before the concept in question becomes clear (K1, K2, K3). This is achieved using the lecture room's document camera, and recorded via QReview, where students can choose to watch the feed from this camera directly (K4). Tutorials, on the other hand, provide an environment for the students to consolidate and refine their knowledge, while being free to work either alone, in groups, or with my help (V1). Tutorials are held in a computer lab, where students can implement the algorithms learned in lectures.

The virtual learning environment QMPlus plays a crucial role in MTH5110, perhaps even more so than most other mathematics modules, as the entire module, including the final exam, is administered via QMPlus. I upload two sets of lecture notes (c.f. Section A2): a typeset version, and a hand-written version that I produce in lectures on the document camera. Although I had initially intended to only upload the typeset notes, several students requested via early feedback that I also upload the handwritten notes, explaining that they find it easier to learn from handwritten notes (K5, K6, V1). All assessments (coursework, mid-term test, and final exam) are administered as QMPlus "assignments", which allows me to upload the students' Maple files (annotated with my feedback) directly back to QMPlus for them to download and review (K4). Early feedback suggested that most students were happy with the organisation of the module's QMPlus page, with about 75% of them saying that they could "always" find what they need there (K5, K6). This seemed to be confirmed in the module evaluation, where "the use of QMPlus has made an appropriate contribution to this module" scored 4.53/5.

QMPlus also provides an "announcement forum", which I have frequently used to communicate with students *en masse*. I also feel that it is important to answer students' emails in a timely manner, and with as detailed/helpful a response as possible, especially since — based on past experience — some students are more comfortable asking questions via email rather than in person (V1). Indeed, the importance of effective email communication with students is also recognised in the literature (Sheer and Fung, 2007). On the other hand, office hours provide the opportunity for students who prefer to do so to discuss questions related to the module with me in a one-to-one setting. Likewise, I make it clear to my students that they may contact me to arrange alternative meetings (V2). More generally, I strive to be *accessible* and *flexible* in my interactions with students, in the sense of the "teacher behaviour checklist" of Keeley *et al.*, 2006 (V3).

After working as a research associate for the bulk of the past three years, organising MTH5110 has provided me with current experience in working in various teaching environments, which I shall draw on to effectively teach other modules in future.

References

L. Alcock and M. Inglis, "Visual considerations in the presentation of mathematical proofs", *International Journal of Media, Technology and Lifelong Learning* **6** (2010) 43–59.

J. Keeley, D. Smith and B. Buskit, "The teacher behaviors checklist: factor analysis of its utility in evaluating teaching", *Teaching of Psychology* **33** (2006) 84–91.

V. C. Sheer and T. K. Fung, "Can email communication enhance professor–student relationship and student evaluation of professor?: Some empirical evidence", *Journal of Educational Computing Research* **37** (2007) 289–306.

Evidencing A5: Engage in continuing professional development in subjects/disciplines and their pedagogy, incorporating research, scholarship and the evaluation of professional practices

Lecturing MTH5110 has been challenging because I had never taught a module on the subject before, and because I had never taught a module with such a heavy programming component. To help overcome these challenges, I have engaged in various professional development initiatives within the School of Mathematical Sciences, both before my teaching began (to prepare myself as well as possible) and during term (to evaluate and improve my teaching).

The usefulness of formal peer observation is widely recognised (e.g. National Research Council, 2003). Before my teaching began, I formally observed a lecture for the Maple-based prerequisite module MTH4105. This class was run in a student-centred fashion, in the sense that it began with a 'mini-lecture', presented via Maple, and then students worked through a corresponding Maple file at their own pace. I have had to structure my lectures somewhat differently to this, because my lecture rooms are not computer labs, and because much of my material is arguably still best presented in a traditional lecture setting (K1, K2). Nevertheless, observing the MTH4105 lecture gave me a good idea of what my MTH5110 students would be used to in terms of class structure and level of difficulty, and showed me how to effectively present lecture material using mathematical software, which I had not done before (K3, V3,V4).

During term, a colleague observed one of my lectures, and I participated in an "education seminar" series, the so-called Bag-Lunch Education Seminar. My peer-observed lecture happened to involve some revision material from linear algebra, in preparation for writing certain matrix-based algorithms. As the material was revision based, I feel that I perhaps did not engage the students with questions as much as I normally would, and my observer noticed this too. After the lecture, we discussed strategies for engaging students even when lecture material is necessarily 'routine' or repetitive. Based on this discussion, I believe it will be useful, when faced with such a situation in future, to use the routine nature of the material to my and the students' advantage, by having the students take a more active role in the lecture, based on their familiarity with the material (K3, K6). Overall, however, the students seemed happy with the level of engagement in lectures, based on several module evaluation comments.

The Bag Lunch Education Seminar covers many varied topics concerning teaching, and seems to be an invaluable initiative of a kind that I had not previously had the opportunity to take part in. One particularly useful point of discussion concerned early feedback, the usefulness of which has long been recognised (Overall and Marsh, 1979). Various suggestions were made for using early feedback to its full advantage, including the importance of stressing to one's students that their feedback is immediately actionable, as opposed to only improving future iterations of a module. I made this clear to my students, and received several extremely useful suggestions that I implemented immediately (see Section A2) (K5, K6). Based on advice in the aforementioned seminar, I also wrote a forum post on the module's QMPlus page within a week or so of receiving feedback, addressing almost every suggestion that I had received, and explaining how it had or would be acted upon (or why this was infeasible). I believe that this showed the students that their opinions were treated seriously and acted upon in a timely manner (V1, V3).

I plan to continue my professional development through the aforementioned avenues, as well as through various avenues in the wider UK university community. In the immediate future, I plan to attend the LMS Education Day in May 2017, which will focus on how lecturers can help to reverse the trend of teacher shortages in mathematics (V4).

References

National Research Council, *Evaluating and Improving Undergraduate Teaching in Science, Technology, Engineering, and Mathematics*, The National Academies Press, Washington DC, 2003.

J. U. Overall and H. W. Marsh, "Midterm feedback from students: Its relationship to instructional improvement and students' cognitive and affective outcomes", *Journal of Educational Psychology* **76** (1979) 856–865.

Appendix – the dimensions of the UKPSF

Areas of Activity	Core Knowledge	Professional Values
(A1) Design and plan learning activities and/or programmes of study	(K1) The subject material	(V1) Respect individual learners and diverse learning communities
(A2) Teach and/or support learning	(K2) Appropriate methods for teaching, learning and assessment in the subject area and at the level of the academic programme	(V2) Promote participation in higher education and equality of opportunity for learners
(A3) Assess and give feedback to learners	(K3) How students learn, both generally and within their subject/disciplinary area(s)	(V3) Use evidence-informed approaches and the outcomes from research, scholarship and continuing professional development
(A4) Develop effective learning environments and approaches to student support and guidance	(K4) The use and value of appropriate learning technologies	(V4) Acknowledge the wider context in which higher education operates, recognising the implications for professional practice
(A5) Engage in continuing professional development in subjects/disciplines and their pedagogy, incorporating research, scholarship and the evaluation of professional practices	(K5) Methods for evaluating the effectiveness of teaching	
	(K6) The implications of quality assurance and quality enhancement for academic and professional practice with a particular focus on teaching	