Specialist Software

Exploiting Synergies between Teaching Mathematics and Programming to Second Year Engineering Students

Jens-Dominik Müller ▪ School of Aeronautical Engineering ▪ Queen's University Belfast

Abstract

Students in Aeronautical Engineering are taught MATLAB in the first year to provide them with programming skills and for use in later courses. In the second year MATLAB is used to enhance the teaching of linear algebra and to apply the mathematical techniques to engineering problems.

Level of Material: Second Year

The Execution

The course links the learning of second year Mathematics, mainly linear algebra, and computing. The course is one module of 72 contact hours over the entire academic year. 36 hours of mathematics and 6 hours of computing are taught in the first semester, leaving 18 hours of mathematics and 12 of computing for the second. The bulk of the mathematics is taught in the first semester in order to provide the tools for a computing assignment in the second semester drawn from an aeronautical engineering problem in structures or aerodynamics.

The first semester mathematics teaching consists of, on average, two hours of lectures and one hour of in-class exercises. Four exercise sheets with solutions are handed to the students during the first semester. The computing in the first semester is kept very simple and parallel to the mathematics. Two problem sheets are set, asking the students to program some basic linear algebra techniques they are familiar with from their hand-calculations, e.g. calculate a matrix-matrix product, calculate a determinant by expansion, perform Gaussian elimination without and with pivoting. The students are familiar with the techniques and programming them enhances the comprehension of the mathematics. Being familiar with the techniques helps to understand how one structures an algorithm.

Since the course is very new, the second semester projects are not defined yet. Their level will be made dependent on the level of competence of the class and the individual candidates. While a minimal level project will be defined that average students can work on following clear instructions, the better students will be encouraged to define individual and more challenging projects. As far as possible, the projects will be drawn from the aerodynamics or structures classes of the second year.

Pre-requisite Knowledge

The mathematics course uses some of the first year knowledge, but is mostly independent. The computing relies on a half-module computing course (3h/week/semester) that introduces the students to programming techniques in general and MATLAB syntax in particular. MATLAB is used in all years and students directly entering the second year would be encouraged to get familiar with the language. Experience with the course so far shows that the students have extremely varied levels of competence in programming. Difficulties for some students need to be addressed by the research student in the lab and by tutorial sessions with small groups of students and the lecturer.

How Are Students With Different Mathematical Backgrounds Supported?

So far no direct entry students without a background in MATLAB needed to be supported, although this could be achieved with individual tutorial help. Different levels of competence are addressed by small group teaching in tutorial sessions for students with particular difficulties and by offering customised levels of difficulty in the projects.

What Support Was Needed?

The MATLAB teaching in the first semester is critical to achieving a programming competence that allows the tackling of interesting and applied projects. The level of achievement in the first year MATLAB course needs to be critically assessed and the level of the second level course adjusted accordingly. As it turns out, students achieved very different levels of competence which in turn required more individual attention in the lab session than was anticipated and more than the single research student could provide.

The Barriers

With the advent of powerful black box packages for structural analysis or CFD, students seem to tend to have an attitude that all the maths they need will be provided by these packages. Students clearly see the benefits of using the computer and IT skills have improved over the past years, but they seem less inclined to do programming. The weaker students who are struggling with the basics of linear algebra see their difficulties compounded by having to program it.

The Enablers

At this young stage of the course it is rather difficult to evaluate what is helping the students to succeed. Given the wide range of competence, key items are individual or small group tutorials to address specific difficulties. Small group tutorials “off the computer” are needed in addition to “online” help of a demonstrator for problems in the computer labs.

Evidence of Success

Success of the concept will be assessed partly from the student feedback and partly from the results in the math exams and the computing projects. Student feedback is sought verbally in pastoral tutorial sessions and from a standard university questionnaire at the end of the course.

For the next academic year a peer evaluation system is planned.

How Can Other Academics Reproduce This?

Basic proficiency in a mathematical programming language needs to be taught prior to using it in an applied context, but any package e.g. Mathcad should be suitable for the design of a similar course. Staffing with demonstrators needs to be sufficient such that students can maintain motivation rather than being frustrated by programming issues.

The level of the programming exercises needs to be carefully graded. Also students will achieve very different levels of proficiency and the setting of projects needs to be flexible to accommodate the varying abilities.

Quality Assurance

All QUB courses undergo a module review procedure at the end of the semester. Pathways and subjects are reviewed every few years. Students are called in for personal tutorials which assure the pastoral care and identify problems with a course early during the semester.

Other Recommendations

A “multi-disciplinary” course such as this will always require more attention in its development and maintenance compared to a pure maths or pure computing course. It requires enthusiasm from the lecturer, as well as good support from colleagues in providing examples, integrating other subjects or supervising projects.