Multiple Approaches

Process Systems Engineering – A Course in Computing and Numerical Methods for Second Year Chemical Engineers

Jack Ponton and Bill Morton ▪ Chemical Engineering ▪ University of Edinburgh

Abstract

We describe a course aimed at providing chemical engineering students with an understanding of the fundamental classes of equations which occur in chemical engineering, the mathematical basis of their numerical solution methods and the basic methods of implementing these in a high level computing language. The course thus integrates elements of both conceptual and practical mathematics and computing.

Level of Material: Second Year

The Execution

All non-trivial problems in chemical engineering practice are nowadays solved using numerical methods on computers. In 1993, we realised that numerical mathematics as taught by our mathematics department was having a negative impact and so we devised this module, taught by chemical engineers, addressing typical chemical engineering problems and requiring the students to write computer programs to solve them.

The problems were chosen to cover the range of types of equation: algebraic, o.d.e. (p.d.e. are left for an 'advanced' option), linear and non-linear and also simple linear and non-linear optimisation problems. From the engineering standpoint they also cover our major application areas including separation processes, reactors, material balances, and data fitting.

The emphasis is very much on practicalities. How do numerical methods work? What can go wrong? How do I write a program for this method? How can I adapt someone else's program? Only what we see as the very minimum necessary amount of theory is developed and issues such as numerical stability are mainly addressed by experiment.

In the full time module we include the teaching of Fortran 90. It is seen as somewhat controversial by some to continue to teach a formal programming language to all engineering students, but we believe that the development of an algorithm, which the writing of any program requires, is an important exercise in rigorous thinking which contributes to students' problem solving skills.

By 1998 we had most of the teaching material for the course on a web server and we decided to use this as the basis for a distance learning module to be offered, along with our web based control course, by Strathclyde University as part of their distance learning MSc in process technology. The major change, other than a different set of hand-in exercises, was to remove the formal programming element and provide the students with a set of web-based modelling tools which would generate models to run in a Lotus or Excel compatible spreadsheet.

Pre-requisite Knowledge

A minimum skill in algebraic manipulation, i.e. "changing the subject of a formula" along with the understanding of a derivative as a rate of change, are really the only absolute prerequisites. For the full time course all students have completed a Scottish First year or A level and are, at least in theory, familiar with the ideas of sets of equations, differentiation, integration and the analytical solution of linear first order o.d.e.s, and this knowledge is assumed in some problems.

How Are Students With Different Mathematical Backgrounds Supported?

For the full time course students have a homogeneous background. For the distance learning course the only assumed mathematical skills are those noted above.

What Support Was Needed?

The course depended critically on the presence of two staff members whose interests and research work lie in the field of mathematical and computing methods, and who believed in the course philosophy of teaching these in a practical and "hands on" manner. A computer lab which could accommodate the whole class was extremely important. Robust software had to be identified and/or developed.

The Barriers

Students do arrive with the idea that numerical mathematics is "difficult" and their earlier exposure to some concepts of numerical methods in A level and/or first year maths courses appears to have reinforced this misconception.

Although most students (and all of the distance learning course participants) are not very good at the mechanics of algebraic manipulation, it can be quite hard to persuade them to avoid it and allow the computer to carry out numerical calculations instead.

Most students hate being made to write programs since these must be absolutely correct; they find it hard to accept that an almost-working program is valueless.

The Enablers

The full time course has had mixed fortunes since its inception nearly ten years ago, but now receives good feedback from students and has a near 100% pass rate. We believe that the key to its present success lies in the workshop sessions; we have always believed that numerical mathematics must be learned by solving problems for oneself, and experience has borne this out. The workshops benefit from a well equipped lab with fast and reliable computers, something that we lacked in the early days.

The organisation of the workshops has also evolved from the early days. Originally students were required to hand in programs and write-ups after the labs. It soon became clear that cheating and copying were becoming endemic. We now require nearly all problems to be done in the lab under the eye of lecturers and tutors. Collaboration, as opposed to copying, is encouraged, and we see evidence of students helping each other in a positive way and even of groups forming spontaneously to tackle a problem jointly.

Evidence of Success

Full time course: In subsequent years students are required to use the techniques they have been taught.

Distance learning module: Feedback from participants has been very positive. Although many students initially expressed concern about their ability to return to a mathematically oriented subject after several years out of university, we have in fact achieved a 100% pass rate for the four cohorts who have taken the module.

How Can Other Academics Reproduce This?

Although tailored to chemical engineering, the underlying idea, that numerate engineering is ultimately about solving equations on computers, can be extended to other disciplines. Much of the background material, and certainly the web-based tools for model construction, are quite generic, and we have either placed these in the public domain or are prepared to licence them without charge.

The key to adopting our approach however, is the presence of an enthusiastic product champion.

Quality Assurance

The full time course is subject to internal and external QA procedures though staff-student liaison committees, student questionnaires, annual course review committee, etc. The degree programmes of which it forms part were given a "commendable" rating (the highest generally awarded) in the 2001 QAA review and our other, but rather more extensive, web based course in process control, on the experience of which we developed the present distance learning module, was awarded the exceptional "exemplary" rating.

The formal exam for the distance learning module is scrutinised by an external examiner. This year he commented that he thought the questions "..were rather more challenging than many others in the course"; despite this we had a 100% pass rate.

Other Recommendations

We add the following suggestions for anyone developing material for this type of module:

- Choose a set of engineering examples to illustrate the mathematical points.
- Make sure that the engineering content of these is very well known to the students; if necessary take time to describe this.
- If possible, extend one engineering example to cover more than one mathematical issue.
- Minimise the amount of theory described.
- For more complex examples, provide sample programs which students can modify.
- Make sure that the mechanics of computer access, printing etc., are as transparent as possible.

References

A public domain version of the teaching material (without hand-in exercises) is available linked from: http://ecosse.org/courses/

Some of the tools are linked from: http://ecosse.org/general/