World Transactions on Engineering and Technology Education
Vol.5, No.2, 2006

A computational mathematics education for students of mechanical engineering

Mikael Enelund & Stig Larsson

Chalmers University of Technology
Göteborg, Sweden

ABSTRACT: In this article, the authors present a new model for computationally-oriented mathematics education. This education combines traditional symbolic mathematics with computational mathematics and programming in the *Matlab* environment. Engineering applications are explored in computational exercises that are taught jointly with the courses in mechanics and thermodynamics at Chalmers University of Technology in Göteborg, Sweden.

INTRODUCTION

The mechanical engineering programme at Chalmers University of Technology, Göteborg, Sweden, has taken part in the development of the CDIO model of engineering education since 2000 [1]. For example, in the courses in mechanics and strength of materials, a common methodology of mathematical modelling and abstract thinking is emphasised. The important goals are to be able to set up mathematical models; formulate the models in the form of equations; simulate the phenomena by solving the equations on the computer; and analyse the simulations in order to assess the correctness of the models and solutions, as well as to improve the learning of basic phenomena and concepts. The interaction between courses is also emphasised.

At the same time, new mathematics courses for engineering education have been developed at Chalmers and implemented in the chemical engineering and bioengineering programmes since 1999. These courses emphasise mathematical modelling, simulation, the use of modern computational tools, and interaction with courses in chemistry and chemical engineering. This is achieved by taking a computational (constructive) approach to the teaching of mathematics.

A basic idea of the reformed mathematics courses is the full integration of the computational (numerical) aspects of mathematics (including programming in the *Matlab* environment) and the analytical (symbolical) aspects. The traditional separation of these aspects, where analytical mathematics is usually presented in the first year and the computational mathematics in a later course, is not adequate.

The authors believe that the computational aspects of mathematics should be presented from the start. This permits the discussion of, for example, nonlinear algebraic and differential equations, and hence also the introduction of realistic applications from applied subjects like mechanics and thermodynamics. This creates motivation to study analytic mathematics and raises the level of both the mathematics and the applied courses. Students write their own equation solvers in *Matlab*, based on the algorithms presented in the lectures that focus on related analytic concepts, such as convergence, limit, linearisation and derivatives.

In this article, the authors present their ongoing work on developing such mathematics courses for the mechanical engineering programme at Chalmers. If they are accepted, the new courses will be launched in the 2007/2008 academic year.

More specifically, the work involves the following:

- Developing course materials for computational mathematics to supplement the traditional textbooks that are used;
- Developing computer-oriented projects and exercises that will be used simultaneously in the mathematics courses and the courses in mechanics and thermodynamics;
- Developing a new introductory course in *Matlab* programming.

Mathematics is a fundamental subject in engineering education and the authors' goal is not a *mathematics for mechanics* toolbox-based course. However, both subjects can benefit from interactions and the exchange of examples.

COURSES

The mathematics courses in the first year of the mechanical engineering programme are listed below, together with the accompanying engineering courses. The year is divided into four periods (quarters of eight weeks).

Period 1:

- Programming in *Matlab* (4.5 ECTS);
- Introduction to mathematics (7.5 ECTS);
- Introduction to mechanical engineering.

Period 2:

- Analysis and linear algebra A (7.5 ECTS);
- Thermodynamics (7.5 ECTS);
- Introduction to mechanical engineering (7.5 ECTS) (ctd).

Period 3:

- Analysis and linear algebra B (7.5 ECTS);
- Mechanics and solid mechanics I (7.5 ECTS).

Period 4:

- Analysis and linear algebra C (7.5 ECTS);
- Mechanics and solid mechanics II (7.5 ECTS).

The third year also contains two mathematics courses, specifically: *Mathematical Statistics* and *Transforms and Differential Equations*. These are not considered at the moment and are, therefore, not included in the work reported here.

The following is a short overview of the contents of the mathematics courses and their connections with the accompanying engineering courses:

- Period 1: *Programming in Matlab*: introduction to *Matlab*, general programming concepts and techniques; *Introduction to Mathematics*: number systems, elementary functions, derivative, integral, graphs, geometry in space, vector, elimination method; *Common Project*: function gallery;
- Period 2: *Analysis and Linear Algebra A*: Nonlinear algebraic equations, derivative, integral, ordinary differential equations; matrix algebra, linear systems of equations, linear independence; *Common Projects with Thermodynamics*: steady state heat equation and equilibrium equations;
- Period 3: *Analysis and Linear Algebra B*: Eigen value problem for matrices and differential equations, linearisation and stability for systems of differential equations; *Mechanics*: statics. principal stresses, differential equations for bars and axles;
- Period 4: *Analysis and Linear Algebra C*: Analysis in several variables; introduction to partial differential equations; boundary value problems; finite element method; *Mechanics*: Elasticity, plane problems, beams and plates, stability, fracture mechanics.

TEACHING

The teaching is organised as follows:

- Lectures;
- Exercises, both analytical and computational;
- Computational assignments common to both the mathematics course and accompanying engineering course.

From the viewpoint of the mechanics and thermodynamics courses, the purposes of the common computational assignments are as follows:

- To allow more complex applications compared with the traditional analytical methods or handbook methods;
- To illustrate phenomena like deflection curves, stability, stress concentrations, stress distribution and fracture to make parameter studies;
- To give an introduction to working with realistic and complex engineering problems.

From the mathematical viewpoint, the purposes are as follows:

- To give a better understanding of mathematical concepts;
- To encourage the study of mathematics.

An overall purpose is to strengthen the ability to apply a modern method of working based on modelling, simulation and analysis; an approach that is generally applicable in all engineering subjects, not just mechanics. Several courses in the second and third years involve computational simulation, eg *Mechanics, Mechatronics, Machine Design, Machine Design Project, Manufacturing, Finite Element Methods (elective), Control Theory* and *Fluid Dynamics*. The teachers of these courses will be informed about the changes of the mathematics education so that they take proper advantage of the new skills and knowledge fostered. This will be followed up in course evaluations and the yearly programme evaluation.

MATLAB

Matlab (Matrix Laboratory) is a software package for numerical matrix computations. It can be used at many levels, from a simple calculator to a rather advanced interactive programming environment. It contains advanced tools for graphics and creating graphical user interfaces, as well as *toolboxes* for many problem areas of science and engineering. *Matlab* has been chosen as the programming environment. Software for symbolic calculation, such as *Mathematica* or *Maple*, have not been used. This may be viewed as an inappropriate promotion of commercial software to students, in particular, because the license cost has increased dramatically recently. However, the lack of competitors with the same flexibility, as well as the long tradition among researchers and lecturers, has motivates the selection of *Matlab* software.

Matlab is utilised in three ways: to illustrate mathematical concepts using ready-made interactive programs; to let students write their own software for implementing the numerical algorithms that are used in the courses; and to present results in the form of graphs, plots and simulations.

The motivation for the first two is obvious. The second one may require some comments. Even though *Matlab* contains tools for most of the computations that need to be carried out, students can write their own programs from the simple bisection algorithm to the rather advanced finite element method. It is believed that there are pedagogical advantages with this. Clearly, it gives students skills in programming and implementing numerical algorithms. But it also forces an understanding of the algorithms and mathematics behind them; *Matlab* is more unforgiving against logical errors and sloppy typing than any teacher. This constructive approach to mathematics based on numerical algorithms gives the computer exercises a natural and strong connection with other forms of teaching. Finally, it is hoped that students will gain self-confidence from using their own software, based on mathematics that they understand, to model advanced engineering systems instead of running *black box* simulations with ready-made programs.

GENERAL VERSUS SPECIAL EQUATIONS

The use of numerical algorithms makes it possible to discuss equations in their most general forms, and use them in mathematical models in engineering and science. In contrast, a discussion of the solvability of general ordinary differential equations is outside the scope of a traditional first year mathematics course and numerical methods are often treated in a later course. Since there is little that can be said or done about the general case, all emphasis is placed on special cases that can be solved symbolically. A reversal of this order of priority is attempted.

A *general algebraic equation* is of the form $f(x)=0$. A special case is $x^2+ax+b=0$, which is solved symbolically by the formula:

$$x = -a/2 \pm (a^2/4-b)^{1/2} \qquad (1)$$

it should be noted that this is not an *exact solution* because it is no more accurate than the accuracy in computing the square root, which is carried out by solving $f(x) = x^2 - 2 = 0$ numerically. However, it provides a formula that can be manipulated by analytical techniques.

A *general ordinary differential equation* is of the form $u'(t) = f(t,u(t))$. A special case is $u' = au$, which is solved symbolically by the formula:

$$u(t) = A\exp(at) \qquad (2)$$

Again, this is not an *exact solution*; it expresses the solution in terms of a well-known special function that can only be evaluated numerically.

The possibility to solve general equations numerically does not diminish the importance of special solutions, such as $\cos(\omega t)$ or $\exp(at)$. On the contrary, they help in interpreting numerical results and understanding complex behaviour in terms of simple and well-known cases. But there is no reason to pursue symbolical computation to its extreme; it is the simple cases that are useful.

A CONSTRUCTIVE APPROACH

The most important feature of the reformed mathematics courses is to target a *constructive* approach based on numerical algorithms. For example, after studying the bisection algorithm, it is noted that it proves the intermediate value theorem, as explained below. It is believed that this constructive/computational approach has the following advantages:

- It makes the mathematics more understandable;
- It makes it possible to discuss general equations, not just simplified special cases;
- It makes it possible to conduct applications early in the curriculum;
- It makes it possible to reach advanced applications at the end of the curriculum;
- It allows open-ended project work.

This approach is based on the textbook (ref. [2]) and has been implemented since 1999 in the chemical engineering and bioengineering programmes at Chalmers University of Technology [3]. However, the textbook has proved to be rather difficult for students and the use of traditional textbooks complemented by lecture notes is planned.

This approach is illustrated by describing the bisection algorithm and intermediate value theorem. The real number, R, is identified with the set of all decimal expansions. If there is a decimal expansion, then an approximating sequence x_n can be formed by truncating it after n decimals. On the other hand, a convergent approximating sequence provides a decimal expansion (this is made rigorous by means of the mathematical concept of the Cauchy sequence). Thus, a real number cannot (in general) be specified exactly; however, it can be evaluated up to *any desired accuracy*.

The use of numerical algorithms to construct new objects is fundamental to the programme. Students first encounter such a constructive argument in the classical construction of $\sqrt{2}$ by solving the algebraic equation $f(x) = x^2 - 2 = 0$ by means of the bisection algorithm. Each student is instructed to write a *Matlab* program implementing this algorithm. The result of the computation with the program is presented in a table with x_n denoting the n^{th} entry of the table. Students then observe that the decimals are fixed going down the table, which indicates that they form a decimal expansion. Indeed, it is proved by constructing $|x_n - x_m| \le K\,2^{-n}$ if $m \ge n$, which shows that the algorithm always generates a decimal expansion. It is concluded that a new real number (decimal expansion) has been constructed: $x = 1.41421356\ldots$

Students are then instructed to prove that x solves the equation in the sense that the residual tends to zero, $f(x_n) \to 0$. In other words, $f(x)=0$. Finally, it is noted that f is strictly monotone for $x>0$, so that the equation $f(x)=0$ cannot have two positive solutions; the solution is unique. Therefore, any other construction would give the same result. This new number is so important that it is given a name: $\sqrt{2}$.

The following steps in the constructive argument should be noted:

- An algorithm that generates a decimal expansion;
- Proof that the limit solves the equation;
- Proof that the solution is unique.

The bisection algorithm proves *the intermediate value theorem*: A continuous function f: $[a,b] \to R$ attains all values between $f(a)$ and $f(b)$. Proof: If y is an intermediate value, then solve the equation $f(x) - y = 0$ by means of the bisection algorithm.

An analogous treatment is also made to *systems of algebraic equations* (algorithm: Newton's method), *systems of ordinary differential equations* (algorithm: Euler's method) and *partial differential equations* (algorithm: the finite element method).

EXAMPLES OF APPLICATIONS

Two examples of joint computer assignments from mathematics and mechanics are presented here. The first one is placed in Period 3 while the second is placed in Period 4. The assignments will be tutored by lecturers and assistants from both mathematics and mechanics.

The first assignment is a static analysis of a plane truss by the displacement-based matrix method. The truss is shown in Figure 1. The task is to determine a value of the area parameter A so that the magnitude of the normal stress is less than half the yield stress everywhere in the truss. Further, using this information, the normal stresses in all members and the joint deformations are to be calculated. The deformed truss should be displayed and a figure showing the relative stress in each

member should be produced. The student needs to handle matrices with dimensions up to 20×20 and a *Matlab* code must be written. Examples of the aims are: to provide an in-depth understanding of fundamental principles used for static analysis and corresponding computational procedures; to serve as an introduction to the application of the finite element technique in structural mechanics; to train students in (*Matlab*) programming from problem definition to working code; to utilise graphical tools for presentation of results and for better understanding; to understand the mathematical treatment of large linear systems of equations.

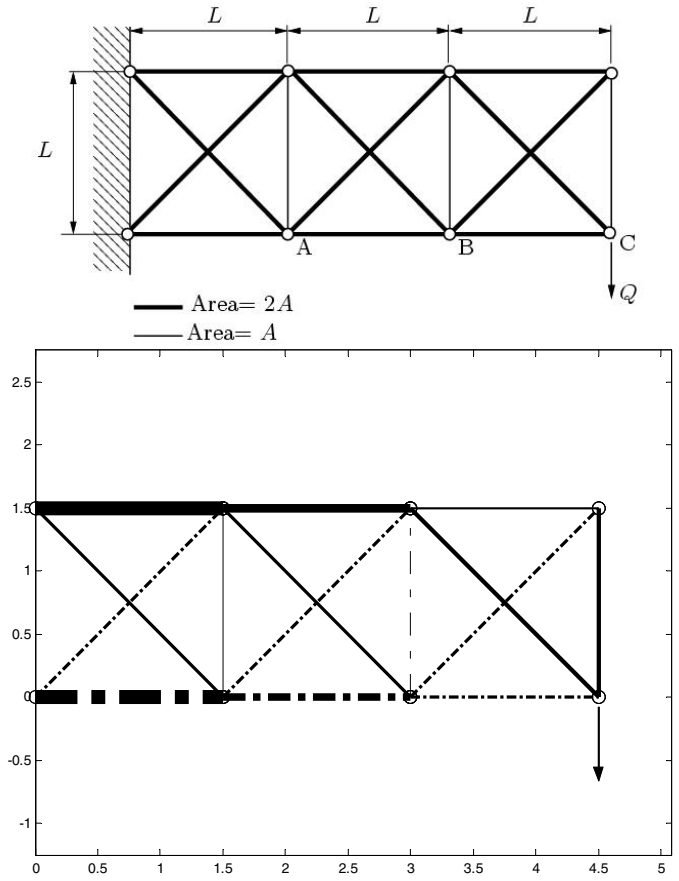

Figure 1: (top): plane truss to be analysed; (bottom): the line width indicates the relative stress in the members (bars), a continuous line indicates tension while a dash-dotted line indicates pressure.

The second application considers a two-dimensional stress analysis of a thin plate with three holes subjected to uniform stress at the vertical boundaries (see Figure 2). Plane stress conditions are assumed. The analysis is carried out using the finite element method and the PDE-toolbox in *Matlab*. The task is to calculate the stress concentration factor K_t, which is defined as $K_t = \sigma_{max} / \sigma_{nom}$. By varying the distance b, students should be able to decide whether the stress concentrations near the holes are correlated or not. Further, the calculated K_t should be compared with tabulated values from handbooks. To reduce the number of elements, symmetries should be used and only a quarter of the plate needs to be considered. This means that special attention needs to be placed on the choice of boundary conditions.

There are several aims with this assignment, for example: by visualising the stress distribution, students can develop intuition about stress distributions and how the stress is increased due to abrupt changes in geometry; motivate the need to study the governing equations of elasticity; it serves as an introduction to the finite element method; it provides an introduction to error estimation and adaptive mesh refinement in the finite element method.

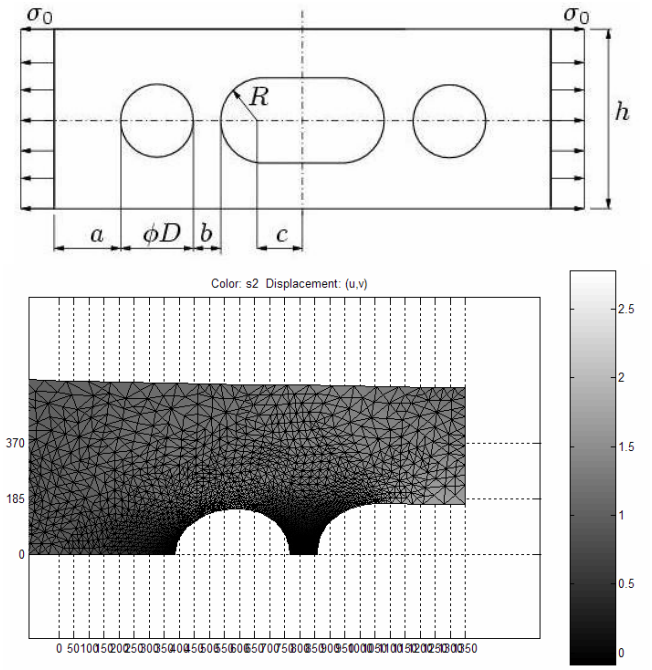

Figure 2: (top): plate with three holes; (bottom): the shading represents the stress distribution in a quarter of the plate. The largest principal stress is presented. The applied stress is σ_0=1. Note the higher stress near the holes and the mesh refinement around the holes, as well as the deformed geometry.

DISCUSSION

From the small scale already implemented in the mechanics courses, as well as the authors' experience from the chemical engineering programme, it is clear that students appreciate the ability to work with realistic models, and the possibility to gain insight and understanding of the behaviour of the systems studied [4]. For instance, students have used these tools in their 2nd year design project and the quality of the analysis has been significantly raised. Further, it is strongly believed that the proposed education also has the potential to increase interest in the underlying mathematics. Clearly, the proposed approach strengthens the connection between the applications and mathematics. This is very important for engineering education, bearing in mind that mathematics is the fundamental tool for most engineering students. This kind of mathematics makes it possible to study the complete problem: from modelling and solution to a simulation of the system and comparison with physical reality. This is one of the cornerstones of the CDIO curriculum. However, experience also shows that it is important to emphasise symbolic hand calculations and basic programming concepts in the teaching, so that these are not lost in the excitement over the possibility of conducting simulations.

REFERENCES

1. The CDIO Initiative (2006), www.cdio.org
2. Eriksson, K., Estep, D. and Johnson, C., *Applied Mathematics – Body and Soul*. London: Springer (2003).
3. Chalmers, Courses at Computational Mathematics (2002), http://www.math.chalmers.se/cm/education/courses
4. Öhrström, L., Svensson, G., Larsson, S., Christie, M. and Niklasson, C., The pedagogical implications of using Matlab in integrated chemistry and mathematics courses. *Inter. J. of Engng. Educ.*, 21, 683-691 (2005).