A Reformed Mathematics Education at Chalmers

Stig Larsson
Department of Mathematics
Chalmers University of Technology
SE–412 96 Göteborg, Sweden
URL: http://www.math.chalmers.se/~stig
E-mail: stig@math.chalmers.se

Kvalitetskonferensen, Norrköping, September 25–27, 2001

Abstract

Engineering education faces the challenge to reform to effectively use the new tools offered by computers. We present a program for a reformed engineering mathematics education. The full program covering the basic mathematics courses of 25–30 credit points has been implemented at Chalmers since 1999 in the "Bio Engineering" (Kb) and "Chemical Engineering with Engineering Physics" (Kf) programs with an enrollment of 90 students and it has been extended to include also the "Chemical Engineering" (K) program reaching a total of 160 students in 2001.

The mathematics reform program presents a synthesis of mathematical analysis, linear algebra and computation with focus on constructive/quantitative aspects and on applications in science and engineering. The computational elements of the program open the possibility to use teaching methods with active student participation including open-ended projects connected to applications.

1 Introduction

Computer-based simulation and virtual reality offer new revolutionary tools to science and technology. New products and systems may be developed and tested through computer simulation on time scales and at costs, which are orders of magnitude smaller than those of traditional techniques based on extensive laboratory testing, hand calculations, and trial and error. Computational modeling is finding new applications in biology, medicine, environmental sciences, economy, and finance.

At the heart of the new simulation techniques lie the new fields of Computational Mathematical Modeling CMM, including Computational Mechanics, Physics, Fluid Dynamics, Electromagnetics, Chemistry, Chemical Engineering, and Biology, all based on solving systems of differential equations using computers, combined with geometric modeling CAD.

Engineering education now faces the challenge to reform to effectively use the new tools of CMM and CAD. The new technology crosses borders between traditional engineering disciplines and schools, and puts strong pressure on teachers to modernize the engineering education in both content and form, from basic to graduate level.

A full program for a new reformed engineering mathematics education has been developed by K. Eriksson, C. Johnson, Chalmers University of Technology, and D. Estep, Colorado State University. The program includes the books *Computational Differential Equations*, Cambridge University Press 1996, and *Applied Mathematics — Body and Soul*, to appear at Springer-Verlag, and various pieces of supporting software. The full program covering the basic mathematics courses of 25–30 credit points has been implemented at Chalmers since 1999 for a group of 90 students in the "Bio Engineering" (Kb) and "Chemical Engineering with Engineering Physics"

(Kf) programs and has been extended to include also the "Chemical Engineering" (K) program with a total of 160 students 2001. The implementation of the reform program was initiated by the former Chalmers vice president of education S. Irandoust, as a part of a general program for education reform, and is carried out in a cooperative effort of the mathematicians M. Asadzadeh, K. Eriksson, C. Johnson, S. Larsson, M. Larson, K. Samuelsson, and N. Svanstedt at Chalmers University of Technology.

The mathematics reform program presents a synthesis of mathematical analysis, linear algebra and computation with focus on constructive and quantitative aspects and on applications in science and engineering. The computational elements of the program open the possibility to use teaching methods with active student participation including open-ended projects connected to applications.

2 The reform program

2.1 Courses

Our program incorporates all the mathematics courses (except mathematical statistics) in the first and second year of the three engineering programs K, Kf, and Kb: analysis in one and several variables, linear algebra, Fourier analysis, programming, numerical analysis, and differential equations. The program is now organized into the following courses:

- year 1
  - Analysis and Linear Algebra A, 4 points
  - Analysis and Linear Algebra B, 7 points
  - Analysis and Linear Algebra C, 4 points
- year 2
  - Differential Equations and Scientific Computing A, 4 points
  - Differential Equations and Scientific Computing B, 4 points
  - Fourier Analysis and Complex Analysis, 4 points

with a total of 27 credit points. (The above description is somewhat simplified, there are minor differences between the three programs.)

2.2 Teaching

The teaching is essentially organized as follows:

- lecture, 4 hours per week, one group of 160 students;
- computer studio, 4 hours per week, four groups of 40 students;
- tutorial, 4 hours per week, in groups of 8–10 students.

The two computer studios in the School of Chemical Engineering, that were opened in the autumn 2000, are class rooms for 48 students with 24 computers. The studios have given a boost to our reform program: the student quickly gets familiar to the Matlab computing environment and gets used to combining analytical and computational mathematical techniques from the beginning. We use older students to give tutorials and to assist in the computer studios.

2.3 Examination

We have used both written and oral exams as well as written and oral project presentations. We try to develop new forms of examination based on (i) hand-in assignments concerning basic skills and (ii) application-oriented project work. Each student has to demonstrate satisfactory knowledge of all the basic elements through the hand-in work collected in an individual folder. Typically, the basic techniques cover the first part of the course and a short oral examination checks that each student masters the material in his/her folder. The results of the project work is presented at the end of the course. This is supposed to give a clear separation of basic skills and applications. This form of examination is still under development.

3 A constructive approach

The most important feature of the reformed mathematics courses is that we try to take a *constructive* approach based on numerical algorithms. For example, after we have studied the bisection algorithm, we note that it proves the intermediate value theorem, as I will explain in Section 6.1 below. We believe 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 do 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.

4 Matlab

Matlab (Matrix Laboratory) is a software 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 for creating graphical user interfaces, as well as "toolboxes" for many problem areas of science and engineering. We have chosen to use Matlab as our programming environment. We do not use any software for symbolic calculation such as Mathematica or Maple.

We use Matlab in two ways: (i) for illustrating mathematical concepts using ready made interactive programs; (ii) for letting students write their own software for implementing the numerical algorithms that are used in the courses.

The motivation for (i) is obvious. The other one may require some comments. Even though Matlab contains tools for most of the computations that we need to do, we let students write their own programs, from the simple bisection algorithm to the rather advanced finite element method. We believe that there are pedagogical advantages with this. Clearly it gives the students skills in programming and implementation of numerical algorithms. But it also forces understanding of the algorithms and the mathematics behind them; Matlab is more unforgiving against logical errors and sloppy typing than any teacher. Our constructive approach to mathematics based on numerical algorithms gives the computer exercises a natural and strong connection with the other forms of teaching. Finally, we hope that the 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.

That computation is a totally integrated part of the courses should be quite clear to the student after spending four hours per week in the computer studio!

5 General versus special equation

The use of numerical algorithms makes it possible for us to discuss equations in their most general form and to 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 put on special cases that can be solved symbolically. We try to reverse this order of priority.

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

$$x = -\tfrac{a}{2} \pm \sqrt{(\tfrac{a}{2})^2 - b}.$$

Note that I did not say "is solved exactly" because this is no more accurate than the accuracy in computing the square root, which done by solving $f(x) = x^2 - 2$ numerically.

A general ordinary differential equation is of the form $u'(t) = f(t, u(t))$. A special case is $u'' + \omega^2 u = 0$, which is solved symbolically by the formula

$$u(t) = A\cos(\omega t) + B\sin(\omega t).$$

Again this is not an "exact solution"; it expresses the solution in terms of well known special functions, which however can only be computed 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 us interpret the numerical results and to understand complex behavior in terms of simple well known cases. But there is no reason to pursue symbolical computation to its extreme; it is the simple cases that are useful.

6 Description of the courses

In this section I briefly describe the contents of some of the courses in our program, emphasizing those aspects that make them different from traditional courses in mathematics for engineers. Each course has a theme: a class of equations that is studied throughout the course.

6.1 Analysis and Linear Algebra A

Theme: system of algebraic equations $f(x) = 0$.

The course begins by introducing the integers and the rational numbers $\mathbf{Q}$ and by exploring their algebraic properties. Then we introduce polynomial functions $f : \mathbf{Q} \to \mathbf{Q}$ and the *Lipschitz condition*:

$$|f(x) - f(y)| \le L_f |x - y|. \tag{1}$$

We use Lipschitz continuity in all proofs even though most results are valid for continuous functions. The advantage of the Lipschitz condition is that it makes it possible to avoid the difficult ϵ-δ argument; (1) is an explicit relation between ϵ and δ, telling us directly how small $|f(x) - f(y)|$ is, if we know how small $|x - y|$ is.

A sequence x_n of rational numbers is called a *Cauchy sequence* if

$$|x_n - x_m| \to 0 \quad \text{as } n, m \to \infty. \tag{2}$$

We define the *real numbers* $\mathbf{R}$ as the set of all *decimal expansions*. Note that there is a correspondence between decimal expansions and Cauchy sequences of rational numbers. On the one hand, writing the terms of a Cauchy sequence x_n in decimal form and using (2), we easily see that the decimals are fixed as n is increased. On the other hand, if we have decimal expansion, then we can form a Cauchy sequence x_n by truncating it after n decimals. Note also, that a real number cannot (in general) be specified exactly; but we can specify it up to any desired accuracy.

If f is Lipschitz continuous and x_n is a Cauchy sequence, then it follows from (1) and (2) that $f(x_n)$ is a Cauchy sequence. In this way we extend f from $f : \mathbf{Q} \to \mathbf{Q}$ to $f : \mathbf{R} \to \mathbf{R}$.

The use of numerical algorithms to construct new objects is fundamental to our program. The 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 the bisection algorithm. The results of two computations with the program are shown in the table. The results indicate that the decimals are fixed as we go down the table and that therefore x_n, y_n are Cauchy sequences.

x_n	y_n
1.50000000000000	1.50000000000000
0.75000000000000	1.25000000000000
1.12500000000000	1.37500000000000
1.31250000000000	1.43750000000000
1.40625000000000	1.40625000000000
1.45312500000000	1.42187500000000
1.42968750000000	1.41406250000000
1.41796875000000	1.41796875000000
1.41503906250000	1.41503906250000
1.41357421875000	1.41455078125000
1.41394042968750	1.41418457031250
1.41416931152344	1.41419982910156
1.41419219970703	1.41420745849609
1.41420364379883	1.41421127319336
1.41420936584473	1.41421318054199

Indeed, students will realize that, by construction, $|x_n - x_m| \leq K2^{-n}$ if $m \geq n$, so that the algorithm always generates a Cauchy sequence. We therefore have constructed two real numbers (= decimal expansions)

$$\bar{x} = 1.4142... \quad \text{and} \quad \bar{y} = 1.41421...$$

The students are then instructed to use the Lipschitz condition to prove that $\bar{x}$, $\bar{y}$ solve the equation in the sense that the residuals tend to zero, i.e.,

$$f(x_n) \to 0, \quad f(y_n) \to 0.$$

In other words, $f(\bar{x}) = f(\bar{y}) = 0$. Finally, we note that f is monotone for $x > 0$, so that the equation $f(x) = 0$ cannot have two positive solutions. Therefore, $\bar{x} = \bar{y} = 1.4142...$. This new number is so important that we give it a name: $\sqrt{2}$.

Note the following steps in the constructive argument:

- an algorithm that generates a Cauchy sequence;
- a proof that the limit solves the equation;
- uniqueness of solutions.

A solution is constructed in the first two steps. The last step means that all constructions lead to the same result.

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

Our study of algebraic equations then proceeds to the fixed point algorithm $x_{n+1} = g(x_n)$ for equations written in the form $x = g(x)$, which is immediately generalized to systems of equations. Each student writes a Matlab program and proves the "fixed point theorem": the system of equations $x = g(x)$ has a unique solution if the Lipschitz constant $L_g < 1$. Proof: show that x_n is a Cauchy sequence.

Trying to optimize the fixed point algorithm, we soon discover Newton's algorithm, which is based on the *derivative* defined by means of *linearization*:

$$f(x) = f(\bar{x}) + f'(\bar{x})(x - \bar{x}) + E_f(x, \bar{x}),$$

where $|E_f(x, \bar{x})| \leq K|x - \bar{x}|^2$. Actually, this should be $E_f(x, \bar{x}) = o(|x - \bar{x}|)$, but, as with the Lipschitz condition, we find it pedagogically advantageous to "cheat" a little here and use an explicit quantitative bound for the remainder.

Dropping the remainder we get Newton's method for $f(x) = 0$:

$$\begin{aligned} &\text{solve for } h_n: && f'(x_n)h_n = -f(x_n), \\ &\text{update:} && x_{n+1} = x_n + h_n. \end{aligned}$$

This is also immediately generalized to systems of equations, in which case the derivative $f'(x)$ is a matrix of partial derivatives and h is a column vector. This motivates the study of matrices, systems of linear equations, and linear algebra.

Of course, each student implements Newton's method in Matlab. Also this algorithm is associated with a theorem, namely, "the inverse function theorem": Suppose that $f(x_0) = y_0$ and that the matrix $f'(x_0)$ is nonsingular. Then f has an inverse function $x = f^{-1}(y)$ near y_0. Proof: solve the equation $f(x) - y = 0$ by the quasi Newton method, $x_{n+1} = x_n - (f'(x_0))^{-1}(f(x_n) - y)$.

Possible applications: equation of state in physical chemistry, equilibrium equations in chemical reactions.

We now know how to construct new real numbers by solving algebraic equations $f(x) = 0$. We also have access to the following classes of functions:

- polynomials $p(x)$, $q(x)$;
- rational functions $\dfrac{p(x)}{q(x)}$;
- their inverses: $f(x) = y \Rightarrow x = f^{-1}(y)$.

What about $\log(x)$, $\exp(x)$, $\sin(x)$, $\cos(x)$? This question is answered in the following course.

6.2 Analysis and Linear Algebra B

Theme: system of ordinary differential equations $u'(x) = f(x, u(x))$.

We follow same constructive approach as for algebraic equations. Beginning with the simplest case,

$$\begin{aligned} &u'(x) = f(x), \quad a < x < b, \\ &u(a) = u_a, \end{aligned}$$

we introduce the algorithm (the rectangle rule),

$$U_n(x_i) = U_n(x_{i-1}) + h_n f(x_{i-1}), \quad h_n = 2^{-n}(b - a).$$

The proof that this generates a Cauchy sequence as the step length is decreased is rather technical, but we spend some effort to explain the idea in this simple situation. This results in a proof of the "fundamental theorem of calculus". Having done this, the (less ambitious) students may then skip the analogous proofs of the more general cases to come.

In this way we construct the *integral*:

$$u(x) = u_a + \int_a^x f(y)\, dy$$

and we explore its properties. A corresponding Matlab program is mandatory.

As a first application of this we integrate

$$u(x) = \int_1^x \frac{1}{y}\, dy.$$

This is a new function, which we call the *natural logarithm*, $u(x) = \log(x)$. All its well known properties are derived from the construction.

The next step is to solve

$$\begin{aligned} u' &= u, \quad x > 0, \\ u(0) &= 1. \end{aligned}$$

Students implement the same algorithm, $U_n(x_i) = U_n(x_{i-1}) + h_n U_n(x_{i-1})$, and construct the *exponential function*, $u(x) = \exp(x)$. Its properties are explored.

In order to construct the *trigonometric functions* we solve the second order equation

$$\begin{aligned} &w'' + w = 0, \quad x > 0, \\ &w(0) = 0,\ w'(0) = 1, \end{aligned}$$

by writing it as a system of two first order equations $u' = Au$, where $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} w \\ w' \end{bmatrix}$ and $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. The implementation of the algorithm $U_n(x_i) = U_n(x_{i-1}) + h_n A U_n(x_{i-1})$ is a simple modification of the program for the exponential function. We obtain $w(x) = u_1(x) = \sin(x)$, $w'(x) = u_2(x) = \cos(x)$. We prove that this coincides with the well known geometric definition together with other properties.

Finally, we solve the *general system of ordinary differential equations*:

$$\begin{aligned} u'(x) &= f(x, u(x)), \quad x > a, \\ u(a) &= u_a. \end{aligned}$$

The implementation of the algorithm $U_n(x_i) = U_n(x_{i-1}) + h_n f(x_{i-1}, U_n(x_{i-1}))$ is a simple modification of that for the trigonometric functions. In this way, step-by-step, each student writes his own ODE solver with the same syntax as Matlab's built-in ODE solvers. Depending on their individual skill and ambition, students also implement improved versions of the basic algorithm, including such things as higher order method, implicit method, adaptive step selection.

We also devote time to symbolic calculation (by hand) of integrals. Similarly, we solve simple special ordinary differential equations symbolically: separable nonlinear ODEs, linear ODEs with constant coefficients. The latter motivates further study of linear algebra: the eigenvalue problem, basis, subspace.

Applications: mechanical systems, mixing of fluids, chemical reaction kinetics, population dynamics.

6.3 Analysis and Linear Algebra C

Theme: partial differential equation $-\nabla \cdot (a(x)\nabla u(x)) = f(x)$**.**

Here we study the usual topics of calculus in several variables: partial derivative, gradient, divergence, multiple integral, curve integral, surface integral, and integral theorems.

Applications include: heat conduction, diffusion, electrostatics. In this course we aim at setting up such models involving the partial differential equations. We solve them in the following courses.

6.4 Differential Equations and Scientific Computing A, B

Theme: nonlinear system of partial differential equations $u_t - \nabla \cdot (a(x,u)\nabla u) = f(x,u)$**.**

In these courses students study the basic theory of partial differential equations and, starting from template programs, they implement the finite element method in Matlab in a style similar to Matlab's PDE Toolbox.

This makes it possible to do advanced applications projects involving coupled phenomena such as heat conduction, diffusion, convection, chemical reaction, and fluid flow.

7 Conclusion

The experience of the reform so far is encouraging: The key goal of giving each student the ability to effectively use computational mathematical modeling in applications has been reached to a large extent. At the end of the mathematics program in the first semester of the second year, the students are able to model, for example, complex reaction-diffusion-convection systems using their own computer programs, which are based on mathematics that they seem to master. This ability opens new possibilities for using mathematical modeling in courses in Chemical Reaction Engineering, Chemical Engineering Design, Transport Processes, Physical Chemistry, Thermodynamics, which play a central role in the chemical engineering education.

The mathematics reform program is being evaluated by U. Runesson from the Department of Education at Göteborg University. The final report is not yet available.

Detailed information about the mathematics reform program can be found on
http://www.phi.chalmers.se/education