A Computationally Oriented Textbook for Calculus and Linear Algebra

Stig Larsson

Department of Mathematical Sciences
Chalmers University of Technology
and University of Gothenburg

Öresundsdagen, Köpenhamn, 2018

Mechanical engineering at Chalmers

- Computationally oriented mathematics courses since 2007.
- Standard text book + additional lecture notes and computer exercises.
- 2014: recruited Anders Logg and Axel Målqvist
- Writing a complete text together with Logg and Målqvist. Preliminary version 2018–19. First edition 2019.
- Inspired by "Body and Soul" by Claes Johnson et al.
- Developed for the Chemical engineering program 1999–

Curriculum in year 1

- Period 1
  - Programming in Matlab (4.5 ECTS)
  - Introduction to mathematics (7.5 ECTS)
  - Introduction to mechanical engineering
- Period 2
  - Calculus in one variable (7.5 ECTS)
  - Computer aided design CAD (7.5 ECTS)
  - Introduction to mechanical engineering (7.5 ECTS), continued
- Period 3
  - Linear algebra (7.5 ECTS)
  - Statics and solid mechanics (7.5 ECTS)
- Period 4
  - Calculus in several variables (7.5 ECTS)
  - Solid mechanics (7.5 ECTS)

General vs special eqn / numerical vs symbolic solution

- General equations
  - algebraic equation: $f(x) = 0$
  - linear system: $Ax = b$
  - ordinary differential equation: $u'(t) = f(t, u(t))$
  - partial differential equation: $-\nabla \cdot (a(x)\nabla u(x)) = f(x)$
- Special equations
  - algebraic equation: $x^2 + ax + b = 0, \quad x = -\frac{a}{2} \pm \sqrt{(\frac{a}{2})^2 - b}$
  - ordinary differential equation: $u'' + \omega^2 u = 0,$ $u(t) = A\cos(\omega t) + B\sin(\omega t)$
  - partial differential equation: $-\Delta u = 0, \quad u(x) = -1/(4\pi|x|)$
- General equation: the solution is constructed as $u = \lim_{n\to\infty} u_n$ by some algorithm, can also be implemented on a computer.
- Special equation: symbolic solution.
- Special functions are solutions of special equations.
- Special and general functions cannot be computed exactly, but approximated to any desired accuracy.

Pedagocical advantages

By taking a constructive, algorithmic, computational approach we can

- do real mathematics, not just symbolic manipulations
- include general equations in the basic courses
- include interesting examples and applications
- train programming: logical thinking, abstraction, fun
- do computational simulations with realistic models

Real mathematics: definition, proof, convergence, ...

Pedagocical advantages

By taking a constructive, algorithmic, computational approach we can

- do real mathematics, not just symbolic manipulations
- include general equations in the basic courses
- include interesting examples and applications
- train programming: logical thinking, abstraction, fun
- do computational simulations with realistic models

Real mathematics: definition, proof, convergence, ...
Old fashioned mathematics.

Period 1: Introduction to mathematics

Equation: $f(x) = 0$
Sequence, convergence, Cauchy sequence, real number, decimal expansion.
Function, limit, continuity, Lipschitz continuity. Epsilon–delta.
Bolzano's theorem, Banach's fixed point theorem.
Derivative, linearization, Taylor's formula. Series.
Error estimates.
Computer exercises include:

1. Bisection algorithm
2. Fixed point iteration
3. Numerical derivative
4. Newton's method (with numerical derivative)

The students write their own programs implementing the mentioned algorithms in Matlab. Or Python in the future?

```
function Df=derivative(f,x)
function x=newton(f,x,tol)
```

Constructive mathematics

real number = decimal expansion = Cauchy sequence

Bolzano's theorem. A continuous function f with $f(a)f(b) < 0$ has at least one root in $[a, b]$.

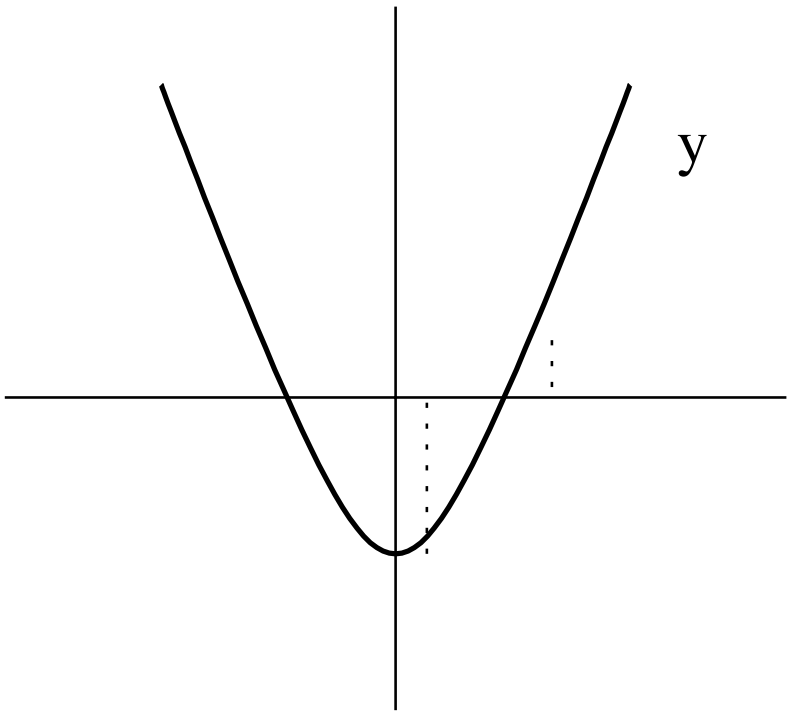
Constructive proof:
Solve $f(x) = 0$, i.e., construct decimal expansion = real number $x \in \mathbf{R}$.

1. bisection algorithm gives x_n
2. convergence (Cauchy sequence = decimal expansion) $x_n \to x$
3. x is a solution $f(x) = 0$
4. uniqueness (when possible)

Example: $f(x) = x^2 - 2 = 0, \quad x = 1.4142\ldots =: \sqrt{2}$

The bisection algorithm

example: $f(x) = x^2 - 2 = 0$



$y = x^2 - 2$

The bisection algorithm

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

Decimal expansions (Cauchy sequences):
$\bar{x} = 1.4142\ldots$
$\bar{y} = 1.41421\ldots$

Lipschitz condition $\Rightarrow$
$f(x_n) \to 0$
$f(y_n) \to 0$
$f(\bar{x}) = f(\bar{y}) = 0$

f monotone $\Rightarrow$ uniqueness:
$\bar{x} = \bar{y} = 1.4142\ldots =: \sqrt{2}$

Period 2: Analysis in one variable

Equation: $u' = f$
The integral, upper lower sums. Riemann sum. Fundamental theorem of calculus. Quadrature methods, convergence. Error estimate.
Ordinary differential equation. System of ODE. Existence and uniqueness via Picard iteration in $C([a, b])$. Construction of special functions as solutions of ODE ($\exp(x)$, $\cos(x)$, $\sin(x)$ ...) Numerical methods, convergence. Error estimate.
Laplace transform.
Computer exercises:

1. The integral
2. Euler's explicit method for systems of ODE
3. Implicit methods (`function [t,U]=myode(f,int,ua,h)`)
4. Boundary value problem (shooting method with Newton solver from period 1)

Period 3: Linear algebra

Equation: $Ax = b$
Geometry and vector algebra in $\mathbf{R}^3$.
Gauss elimination, determinant, inverse matrix. LU factorization.
Orthogonality, eigenvalue problem. Least squares method.
Computer exercises:

1. Geometry
2. Matrix algebra
3. Systems of linear equations, error analysis, condition number
4. Least squares (calibration of Norton's law for creeping)

Period 4: Analysis in several variables

Equation: $-\nabla \cdot (a(x)\nabla u(x)) = f(x)$
Partial derivative. Linearization, Jacobi matrix, Newton's metod. Taylor's formula. Optimization. Lagrangre multipliers. Both vector notation and linear algebra notation.
Curves and surfaces.
Double and triple integral. Curve integral, surface integral. Gauss divergence theorem.
Boundary value problems and the finite element method in one and several variables.
Computer exercises include:

1. Visualization of multivariable functions
2. Numerical Jacobi matrix and Newton's method (`A=jacobi(f,x)`, `x=newton(f,x,tol)`)
3. Optimization (Lagrange's method with `jacobi` and `newton`)
4. The finite element method in 1–D
5. The finite element method in 2–D (Matlab's PDE Toolbox)

In one variable

The fundamental theorem of calculus:

$$\int_a^b \mathrm{D}u \, \mathrm{d}x = \Big[u\Big]_a^b = u(b) - u(a)$$

Derivative of a product:

$$\mathrm{D}(uv) = \mathrm{D}u \, v + u \, \mathrm{D}v$$

They combine into the "integration by parts" formula:

$$\int_a^b \mathrm{D}u \, v \, \mathrm{d}x + \int_a^b u \, \mathrm{D}v \, \mathrm{d}x = \Big[uv\Big]_a^b$$

Analysis in several variables

A version of the fundamental theorem of calculus:
the Gauss divergence theorem

$$\iiint_D \nabla \cdot \mathbf{F} \, \mathrm{d}V = \iint_S \hat{\mathbf{N}} \cdot \mathbf{F} \, \mathrm{d}S$$

A product rule:

$$\nabla \cdot (\phi \mathbf{F}) = \nabla \phi \cdot \mathbf{F} + \phi \nabla \cdot \mathbf{F}$$

They combine into an "integration by parts" formula:

$$\iiint_V \nabla \cdot \mathbf{F} \, \phi \, \mathrm{d}V + \iiint_V \mathbf{F} \cdot \nabla \phi \, \mathrm{d}V = \iint_S \hat{\mathbf{N}} \cdot \mathbf{F} \, \phi \, \mathrm{d}S$$

Analysis in several variables

A version of the fundamental theorem of calculus:
the Gauss divergence theorem

$$\iiint_D \nabla \cdot \mathbf{F} \, \mathrm{d}V = \iint_S \hat{\mathbf{N}} \cdot \mathbf{F} \, \mathrm{d}S$$

A product rule:

$$\nabla \cdot (\phi \mathbf{F}) = \nabla \phi \cdot \mathbf{F} + \phi \nabla \cdot \mathbf{F}$$

They combine into an "integration by parts" formula:

$$\iiint_V \nabla \cdot \mathbf{F} \, \phi \, \mathrm{d}V + \iiint_V \mathbf{F} \cdot \nabla \phi \, \mathrm{d}V = \iint_S \hat{\mathbf{N}} \cdot \mathbf{F} \, \phi \, \mathrm{d}S$$

We can now:

- derive the heat equation with general boundary conditions
- derive the finite element method based on weak formulation
- point out the analogy with the equations of linear elasticity

Boundary value problem and FEM

- One variable:

$$
\begin{aligned}
-D(a(x)Du(x)) + c(x)u(x) &= f(x), && \text{for } x \in I = (K, L),\\
a(x)D_n u(x) + k(x)(u(x) - u_A) &= g(x), && \text{for } x = K,\ x = L.
\end{aligned}
$$

```
function [U,A,b]=MySolver(p,t,e,EqData,BdryData)
```

Boundary value problem and FEM

- One variable:

$$
\begin{aligned}
-D(a(x)Du(x)) + c(x)u(x) &= f(x), && \text{for } x \in I = (K, L),\\
a(x)D_n u(x) + k(x)(u(x) - u_A) &= g(x), && \text{for } x = K,\ x = L.
\end{aligned}
$$

```
function [U,A,b]=MySolver(p,t,e,EqData,BdryData)
```

- Several variables: $\mathbf{F} = -a\nabla u$ heat flux density

$$
\begin{cases}
-\nabla \cdot (a\nabla u) + cu = f & \text{in } D,\\
\hat{\mathbf{N}} \cdot (a\nabla u) + k(u - u_{\mathrm{A}}) = g & \text{on } S_2 \text{ (Neumann/Robin)},\\
u = u_{\mathrm{A}} & \text{on } S_1 \text{ (Dirichlet)}
\end{cases}
$$

Weak formulation and finite element method
Matlab's PDE Toolbox and Ansys (and FEniCS in the future?)

Analogy with solid mechanics

- Heat flux density: $\mathbf{F} = -a\nabla u$
- Heat equation: $\nabla \cdot \mathbf{F} = f$
- Stress: $\sigma(\mathbf{u})$= tensor function of displacement field $\mathbf{u}$
- Equilibrium equations: $\nabla \cdot \sigma = \mathbf{f}$

New textbook

S. Larsson, A. Logg, and A. Målqvist
Matematisk analys och linjär algebra

Preliminary version 2018–19, first edition Studentlitteratur 2019.

Four parts. One chapter — one week.

Del I. Differentialkalkyl

- 0. Ekvationen $f(x) = 0$
- 1. Reella tal
- 2. Funktioner
- 3. Gränsvärde och kontinuitet
- 4. Derivata och linjärisering
- 5. Taylor polynom och serier
- 6. Ekvationslösning: $f(x) = 0$
- 7. Tillämpningar

New textbook

Del II. Integralkalkyl

- 0. Ekvationen $u' = f$
- 1. Integralen
- 2. Integrationstekniker
- 3. Ordinära differentialekvationer
- 4. Laplacetransform
- 5. System av ODE
- 6. Numerisk lösning av ODE
- 7. Tillämpningar

Each chapter has:

- Exercises
- Problems
- Computer exercises

Solutions are provided. Both Matlab and Python code.