TMA 4215 Numerical Mathematics: Lecture 01

André Massing (`andre.massing@ntnu.no`)

Aug 19, 2019

1 Overall Course Organization

1.1 Learning Objectives for TMA 4215

1. **Knowledge.** The student has knowledge of basic numerical methods for approximation offunctions, numerical quadrature, numerical solution of ordinary differential equations, and numerical solution of linear and nonlinear equations. The student masters error analysis of numerical methods and understands the concepts stability and convergence.

2. **Skills.** The student masters basic techniques for analyzing a large selection of numericalalgorithms. The student is able to implement the algorithms in a chosen programming language, set up numerical experiments and interpret the results. The student has establishedsound routines for quality assurance in implemented algorithms.

3. **General competence.** The student can describe the chosen scientific method and communicate his or her findings in a written scientific report using precise language.

1.2 Teachers/Involved People

- André Massing, Assoc. Prof. (førsteamanuensis) at the Department of Mathematical Sciences, Room 1340, Sentralbygg 2, andre.massing@ntnu.no

- Henrik Lindell, PhD Candidate (started 1st of August) at the Department of Mathematical Sciences, Room 1201, Sentralbygg 2, henrik.lindell@ntnu.no

- Esten Nicolai Woien, PhD Candidate (started 1st of August) at the Department of Mathematical Sciences, Room ??, Sentralbygg 2, esten.n.woien@ntnu.no

- Kristian Aga, PhD Candidate, Department of Mathematical Sciences, 9th Floor, Sentralbygg 2, kristian.aga@ntnu.no (Help with Smia and interactive learning methods)

1.3 Lectures

- 28 lectures in total
- Wednesdays: 8:15 - 10:00 in S2, Sentralbygg 1
- Thursdays: 8:15 - 10:00 in Smia, Sentralbygg 2

1.4 Tutorials and Office Hours

- Wednesdays: 18:15 - 19:00 (Official) in Smia, Sentralbygg 2 (Supervised lab hours). This are official times, but we could book Smia from 17:15 - 20:00, aiming at 2 supervised lab hours in total (will send poll to determine best time slot)
- Wednesdays: 10:30 - 12:00 (Office hours)

1.5 Computer Labs and Project

- 5 *Computer lab assignments*, usually uploaded every other Wednesday starting from **28th of August** on, submission deadline usually 2 weeks later, precise deadline will be given on the assignments. Despite its name, computer lab assignments will consist of *theoretical* and *practical* problems.
- 1 larger *computer project* towards the end, will be uploaded 4 weeks by the latest before course ends.
- In contrast to the computer Project, computer labs are not graded only passed/not passed. If you don't pass a computer lab after 1. submission, improved solutions can be resubmitted. Teacher assistents will give you short written feedback on missing answers or major deficits. More detailed feedback can be obtained by simply talking to your teaching assistants :), e.g. during lab hours or office hourse.

- Computer lab assignments use the Python language (Python3) and packages from SciPy ecosystem and will be distributed as Jupyter notebooks. Completed assignement have to be submitted as (filled in) Jupyter notebooks and **only** via **Blackword** (don't use email!)
- We encourage you to pair up in teams of 3, *teams can consist of max 4 students if at least one is an exchange student*!

- To obtain a nice and feature-rich/full-fledged scientific python environment, we recommend to install the Anaconda Python distribution which is available for both Linux, MacOS, and Windows.

1.6 Reading Material

- Main textbooks we use for the course are
  - Numerical Mathematics by Alfio Quarteroni, Riccardo Sacco, Fausto Salerie. Can be downloaded from Springer Link (from inside the NTNU network). Referred to as the "Yellow Book" (**YEB**). Hardcopies (hardcover/fineprint) are available at Akademia book store. Personal MyCopy softcover version can be ordered from Springer Link for 25 Euro.
  - Programming for Computations – Python by Svein Linge, Hans Petter Langtangen, downloadable as open e-book. Referred to as the "Black and Orange Book" (**BOB**).
  - Occasionally, lecture material can built upon excerpt from the An Introduction to Numerical Analysis by Endri Süli and David Mayers.

  Online version available via NTNU's library services (You can PDF print up to 100 pages at once) Referred to as the "Blue Book" (**BLUB**).

1.7 Examen

- 1 written exam after the end of the course lecture curriculum/before Christmas
- To pass the course, you need to pass 4 out of 5 computer labs, the project and the examen. The final grade is determined by the grade for the project (30%) and the grade for the examen (70%).

2 Introduction to Numerical Mathematics

Question.

What is Numerical Mathematics?

> Numerical Mathematics is concerned with the solving mathematically formulated problems computationally. (H.Rutishauser)

Often, we need to solve mathematical problems computationally because it is either **not feasible** to compute an exact solution by hand (it takes too much time) or even **not possible** (analytical expression for the solution cannot be derived). Then we need to compute an **approximation** of the solution.

Numerical/computational mathematics is an interdisciplinary subject as the design of efficient and accurate algorithms can often not be separated from their point origin. In many engineering and science problems, we need to go through the following steps to arrive at solution:

1. **Mathematical modeling**: How can we describe the problem in mathematical terms?
2. **Analysis of the mathematical model**. Is the resulting model **well-posed**, in the sense that
   - Is there solution (**Existence**)?
   - Is there only one solution and can there be several (**Uniqueness**)?
   - Does the model depends continuously on the data (**Continuity**)?
3. **Numerical Methods**: Develop a numerical solution or approximation method for the model and understand the **computational complexity, stability and accuracy** of the proposed method
4. **Realization**: Devise an efficient implementation/algorithmic realization of the numerical methods (requires efficient implementation and efficient algorithms), and provide tools to interpret the computed results (e.g. visualization)
5. **Validation**:

All theses Steps are the cornerstones of the field of **Scientific Computing** which Numerical Mathematics is a part of (mostly concerned with 3 and 4).

2.1 Preliminary Outline of Topics

This is a preliminary and rough outline of topics of this course. Clearly, we can cannot cover everything, and some topics will be only mentioned briefly.

- **Module 1**: Linear Systems I
  - Gaussian Elimination
  - LU factorization
  - Pivoting
- **Module 1**: Stability and error analysis of numerical algorithms
  - Stability, forward, backward analysis

- Convergence

- **Module 1**: Linear Systems II
  - Undetermined systems and least-square problems
  - Cholesky decomposition
  - Eigenvalue and Eigenvector computations
  - (Iterative Methods?)

- **Module 2**: Nonlinear Systems
  - Banch fixpoint iteration
  - Newton's method
  - Gauss-Newton method for non-linear least-squares problems

- **Module 3**: Interpolation and approximation
  - Polynomial Interpolation
  - (Piecewise interpolation/splines?)
  - Orthogonal polynoms

- **Module 4**: Numerical Integration
  - Trapezoidal rule
  - Newton-Cotes Formula
  - Gauß-Christoffel quadrature
  - (Romberg/extrapolation methods)

- **Module 5**: Brief Intro to Numerical Methods for ODEs
  - Runge-Kutta methods explicit
  - Adams-Bashforth methods
  - Stiff Problems
  - Runge-Kutta methods implict

- (**Module 5**: Very Brief Intro to Numerical Methods for PDEs?)

3 Linear System

Notice.

We expect that you are familiar with fundamental concepts from linear algebra. For a quick recall, skim through Ch.1 in **YEB**, in particular Section 1.1-1.5, 1.7,1.8, 1.10.

This chapter is concerned with solving linear systems of the form

$$\left.\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{1n}x_n & = & b_2 \\ \vdots & \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{array}\right\} \tag{1}$$

As usual, we denote by $\mathbb{R}^{m\times n}$ the set of $m \times n$ matrices with entries in $\mathbb{R}$. The the system of linear equations can be written in matrix form

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{n1} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}}_{=:A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}}_{:=\boldsymbol{x}} = \underbrace{\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}}_{:=\boldsymbol{b}} \tag{2}$$

or more compact, $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{n\times n}$ is now a *square matrix*, $\boldsymbol{x}$ is the vector of unknows, and $\boldsymbol{b}$ is the right-hand side vector given by the data.

3.1 Cramer's Rule: A Way to Solve Linear Systems Numerically?

You have probably learned about one way to solve the linear system (2) a basic course on linear algebra, namely **Cramer's rule**,

$$x_i = \frac{D_i}{D} \text{ for } i = 1, \ldots, n, \tag{3}$$

where $D = \det(A)$ and D_i is the determinant of the matrix obtained by replacing the i-th column in A by $\boldsymbol{b}$.

Exercise 1: Computing the inverse A^{-1} using Cramer's rule

Show that inverse matrix A^{-1} of A is given by

$$A^{-1} = \frac{1}{\det A} \begin{pmatrix} A_{11} & A_{21} & \ldots & A_{1n} \\ A_{12} & A_{22} & \ldots & A_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ A_{n1} & A_{n2} & \ldots & A_{nn} \end{pmatrix} \tag{4}$$

where $A_{ij} := (-1)^{i+j}\mathrm{Cof}(a_{ij})$, and the **cofactor** $\mathrm{Cof}(a_{ij})$ denotes the **cofactor** of a_{ij} denotes the determinant of the $n-1 \times n-1$ matrix which is obtained by deleting row i and column j from the matrix A.

Hint 1: Recall the possibility to compute the determinant of a matrix by expanding it into subdeterminants,

$$\det(A) = a_{i1}A_{i1} + a_{i2}A_{i2} + \cdots + a_{in}A_{in} \tag{5}$$

for any row i (with a similar expansion for any column j)

Hint 2: Realize that to compute the inverse A^{-1} you have to solve n linear systems of the form $A\boldsymbol{x} = \boldsymbol{b}$ with a particular rhs vector $\boldsymbol{b}$ (one for each column of A^{-1})

Question.

Is Cramer's rule a feasible method to solve the linear system (2)? To answer this question, we have to take a closer look at the computational complexity when computing determinants.

Its time for some Blackboard writing....