

Introduction

Anne Kværnø, Andre Massing

23.03.2021

1 Welcome to TMA4130 - Calculus 4N

1.1 Course organization

For detailed information, we refer to our [course wiki page](#). The instructor team consists of Elisabeth Anna Sophia Köbis, Andre Massing (Instructors), Diego Caudillo and and Ludwig Rahm (Teaching assistants).



Figure 1: Elisabeth Anna Sophia Köbis (Instructor)



Figure 2: André Massing (Instructor)

1.2 Course content

Mathematical modeling and scientific computing. 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



Figure 3: Diego Caudillo (Teaching assistant)

- (a) Is there solution (*Existence*)?
 - (b) Is there only one solution and can there be several (*Uniqueness*)?
 - (c) Does the model depends continuously on the data (*Continuity*/*Stability*)?
3. **Solve the resulting model.** Two approaches are typically considered:
- (a) *Analytical methods* to compute the *exact solution* for the model
 - (b) *Numerical methods* to compute an solution for the model by *computational means*. The obtained solution can be either *exact* or just an *approximation* to the exact solution.
4. **Realization:** Devise an efficient algorithmic realization of the chosen solution method
5. **Postprocessing** of the computed solution to make it interpretable e.g. through visualization
6. **Verification and Validation**
- (a) *Verification*: "Do I solve the problem correctly", i.e. does the realization of the method in 3. really solve the model problem in 2.?
 - (b) *Validation*: "Do I solve the correct problem?", i.e. does the model developed in 2. describe the problem in 1. correctly/sufficiently?

This entire process (except for 3.a) is at the heart of the subject *Scientific Computing*

Question.

What is *numerical mathematics*? "Numerical Mathematics is concerned with the solving mathematically formulated problems computationally". (H.Rutishauser)

As such it mostly concered with 3. B), 4. and 5. The main challenges are to understand

- whether and when the numerical method itself results in a *well-posed* problem
- the *complexity* of the proposed method/algorithm. How many operations do I (or rather the computer) have to perform to compute the solution?
- the accuracy of the proposed method (often in comparison to the work complexity). How "close" is the computed solution to the actual solution?

In this course we will learn a lot about analytical and numerical methods for various equations, including algebraic equations, ordinary differential equation and partial differential equations:

1. Module: Introduction to numerical mathematics

- Interpolation of functions (N)
 - Numerical integration a.k.a quadrature (N)
 - Solving non-linear algebraic equations numerically (N)
2. Module: Analytical and numerical methods for ordinary differential equations (ODEs)
- Laplace transform (A)
 - Runge-Kutta methods (N)
3. Module: Analytical and numerical methods for partial differential equations (PDEs)
- Fourier series and transforms (A) (also extremely useful in the analysis of signals!)
 - Fourier-based methods to solve the heat and wave equation analytically (A)
 - Finite-difference methods to solve the heat and wave equation numerically (N)

1.3 Whetting your appetite

Example: Mathematical modeling. Motivated by the ongoing corona virus pandemic, we consider a (simple!) model for the spreading of an infectious disease, which goes under the name **SIR model**.

The SIR models divides the population into three population classes, namely

- $S(t)$: number individuals **susceptible** for infection,
- $I(t)$: number **infected** individuals, capable of transmitting the disease,
- $R(t)$: number **removed** individuals who cannot be infected due death or to immunity after recovery

The model is of the spreading of a disease is based on moving individual from S to I and then to R . The final ODE system is given by

$$\begin{aligned} S' &= -\beta SI \\ I' &= \beta SI - \gamma I \\ R' &= \gamma I, \end{aligned}$$

where

- β denotes the **infection rate**, and
- γ the **removal rate**.

Having this ODE in mind, an epidemic model can be simply specified by writing down a *state transition diagram*

$$S \xrightarrow{\beta} I \xrightarrow{\gamma} R$$

This is rather simpler model. Later in this course you will be asked to

- consider a local outbreak in Trondheim
- extend this model to account for the hospitalized population and solve it numerically
- determine at which infection rates the number of hospitalized people will rise above the number of available hospital beds.

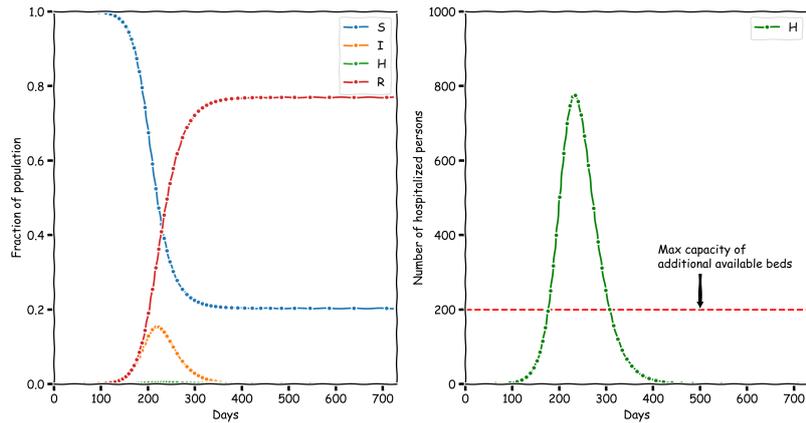


Figure 4: SIR-models **Remark.** This course focuses on solving such models (either analytically or numerically), but not so much about the modeling part itself. Nevertheless, a at few occasion, we try to explain how certain models can be derived.

Example: Accuracy. We use a simple numerical method (which we will get to know as explicit Euler method) to numerically solve the ODE

$$y'(t) = \lambda y(t), \quad y(t_0) = y_0$$

with the analytical solution

$$y_{ex}(t) = y_0 e^{\lambda(t-t_0)}.$$

Solved for $\lambda = 1.0$, $t_0 = 0$ and $y_0 = 1$. Below N refers to the number of points we used to compute an approximate solution y .

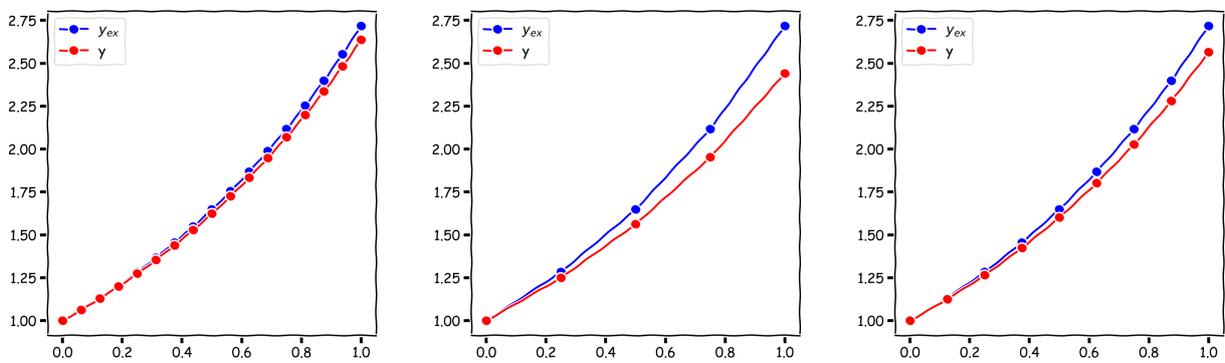


Figure 5: Numerical solution using explicit Euler with $N = 5$, $N = 9$ and $N = 17$ solution points.

Clearly, increasing the number of points N will increase the computer work. But what do we get back for the additional in terms of accuracy? How fast does the numerical solution approach the real solution? Can we estimate the error $y - y_{ex}$ as a function of N ?

Example: (conditional) stability. We consider the same model problem and same numerical method as in the previous example, but now we consider negative λ .

This is an example of an unstable/only conditionally stable algorithm. Depending on the absolute value of λ , a certain amount of (equi-distant) points has to be used, otherwise we get a completely meaningless and wrong numerical solution.

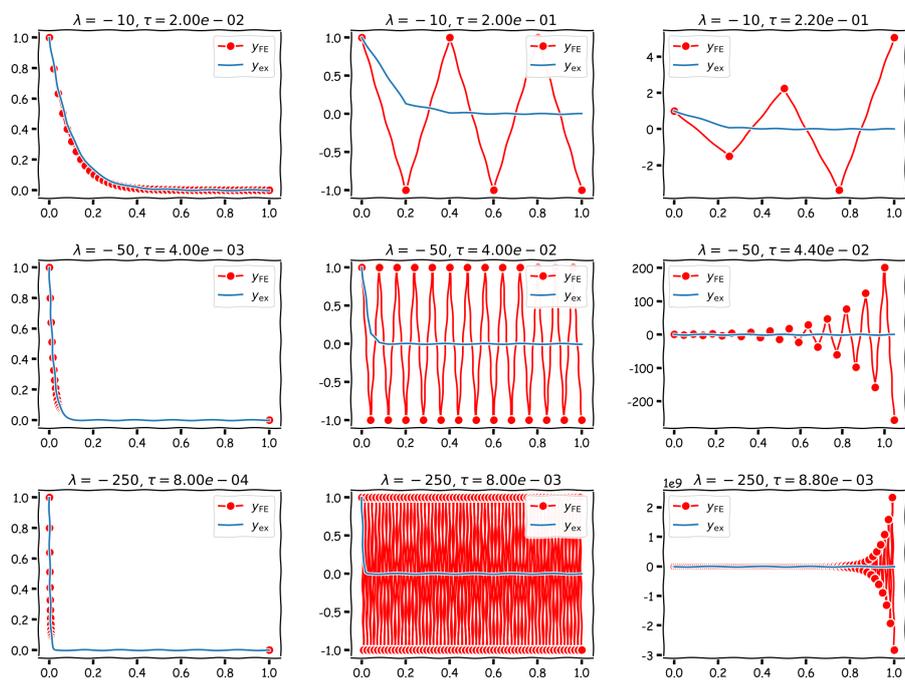


Figure 6: Stiff problem