

# Introduksjon til modellering

Matematiske modeller og simulering for å løse problemer

Feil som kan kom over

Hva er en model? Hvorfor modellere?

Beskrive et system så enkelt som mulig      -> skal svare på et spørsmål om et system

Hvordan sykler jeg?

Hvorfor går ikke sykkelen fremover?

Så enkle som mulig, men svare på spørsmålet

Beste modellen er den enkleste modellen, som fremdeles svarer på spørsmålet

- Bygge modellen ned og opp

Følger denne oppbygningen til hva som skal inn i modellen

1. Spørsmål eller problem      -> Hva skal løses?
2. System analyse      -> Hva er relevant for spørsmålet og problemet?
3. Modellering      -> Samle systemanalysen
4. Simulering      -> Bruke modelleringssmodellen for å svare på spørsmål
5. Validering?      -> Problem løst?

Repeat til steg 2 for å lage ny modell

En god modell består valideringen

## Warning 1.5: What is a *system*?

System = whatever we are analyzing. Can be single object or collection of processes. E.g. bike, falling skydiver, temperature in a hot water tank, etc.

## What is a model?

The system analysis part is usually the most challenging part of a modeling scheme. This step requires knowledge about the system to be studied and can be time consuming.

For a mathematical model, the system analysis step means to identify the equations and parameter values that are needed.

### Example 1.6: Skydiving

If someone jumps out of an airplane, what is the speed of that person after 10 seconds?

System analysis: Falls under gravity. Starts with speed of zero.

$$v = v_0 + a \cdot t$$

Modeling:

$$v = 9.8 \cdot t$$

Simulation:

$$v = 9.8 \cdot 10 = 98 \text{ m/s}$$

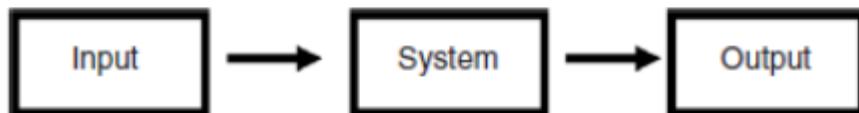
Validation? If no, what else should be included in the system analysis?

6

Mangler luftmotstand      -> dårlig modell

Hva er en matematisk modell?

- Bruker tall og matematiske operasjoner
- Svare på spørsmålet der svaret er et tall



Input / output system

Sammenhengen mellom tid og fart til fallskjermshopper

- antar sammenheng mellom 2 fenomen

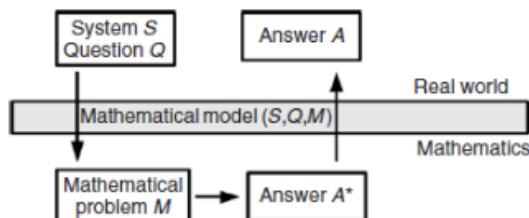
## What is a mathematical model?

We here follow the definition of Kai Velten:

### Definition 1.7: Mathematical model

A *mathematical model* is a triplet  $\{S, Q, M\}$ , where  $S$  is a system,  $Q$  is a question concerning  $S$ , and  $M$  is a set of mathematical statements which can be used to answer  $Q$ .

- ▶ System = Whatever we study.
- ▶ Question = What we want to find out.
- ▶ Mathematical statements = Our path to the answer of the question.



## What is a mathematical model?

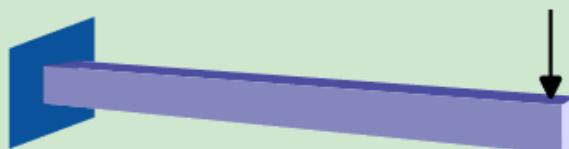
### Example 1.8: Height of people in this room

What is the average height of people in this room?

- ▶  $S$ : People in this room.
- ▶  $Q$ : What is the average height of  $S$ ?
- ▶  $M$ :  $\bar{h} = \frac{1}{n} \sum_{i=1}^n h_i$

### Example 1.9: A bending beam

How much will be steal beam of size  $10 \times 1 \times 1\text{m}$  attached on one side and under a load of  $10 \text{ MN}$  on the opposite side bend?



- ▶  $S$ : Steel beam of given size.
- ▶  $Q$ : How much does it bend under given load?
- ▶  $M$ : Partial differential equation (PDE) describing elastic properties of beam.

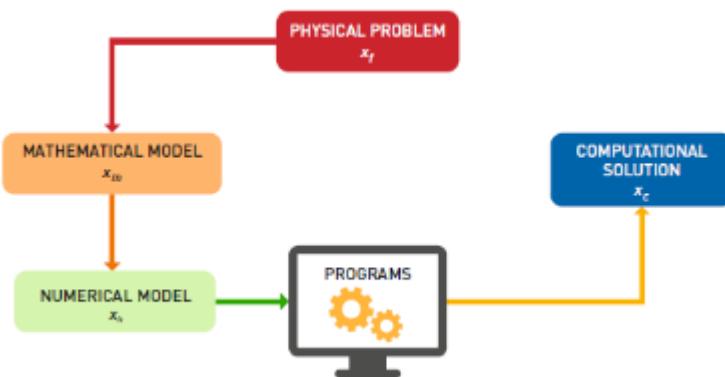
Hva er en simulering?

Utføre de matematiske operasjonene i simuleringsteget

- Kan må brukes en datamaskin vs penn og papir

Numerisk model er overgangen mellom den matematiske modellen og datamaskinen

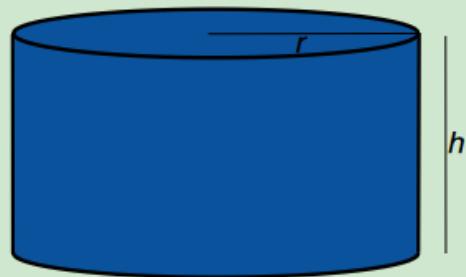
- En god nok løsning



## Example

### Example 1.10: Minimize surface area of hot water tank

Which radius and height should a cylindrical 400 liter hot water tank have to minimize its surface area? (You can assume the walls of the tank have zero width)



- ▶ Identify  $S$ ,  $Q$  and  $M$ .
- ▶ Perform the simulation (find the answer!)

S: Sylingertank

Q: Minimere overflateareal?

$$M = \min A(r, h)$$

$$V = 0.4 \text{ m}^3$$

## UTREGNING

### SIMULERING - LØSNING AV PROBLEM

#### VALIDERE

Finne ut om det er minimum ved en litt større/mindre r, for å sjekke at neste punkt er over

#### Tilstandsvariabel

- Beskrivelse av tilstand av systemet i en matematisk modell

#### Systemparametre

- Egenskaper ved systemer som er nødvendig, for å regne ut tilstandsvariabel

#### Example 1.10: Minimize surface area of hot water tank - continued

- State variables: Surface area.
- System parameters: Radius and height.

## Variables and parameters

When working with problems with much of information, it can be helpful to early on identify what state variables and system parameters are relevant when setting up the mathematical model.

#### Example 1.11: Transmitting fluid

Pipe A has diameter 2 cm and can transmit 1000 l/day of water when having a pressure drop of 100 Pa. Pipe B has diameter 3 cm and can transmit 2000 l/day of water with a pressure drop of 120 Pa. The costs of using pipe A and pipe B, is 100 and 400 NOK/day, respectively. If you need to transmit 4000 l and only have 500 NOK available, how can you transmit the water as quick as possible?

- S: Two pipes of different capacity and costs.
- Q: How to transmit as quick as possible with given budget?
- M:
  - State variables: Number of days using pipe A and pipe B:  $d_A, d_B$ .
  - System parameters: Prices per day (100, 400 NOK/day) and capacities per day (1000, 2000 l/day).
  - Not relevant system parameters: Sizes of pipes, pressure drops.

## SE PÅ SPØRSMÅL FOR Å FINNE RIKTIG TILSTANDSVARIABLER

- Støy
- Filtrere hva som er viktig

### Example 1.11: Transmitting fluid - continued

▶ M:

$$\begin{aligned} \text{Costs:} & d_A \times 100 + d_B \times 400 \leq 500 \\ \text{Transmitted:} & d_A \times 1000 + d_B \times 2000 = 4000 \end{aligned}$$

Solving second for  $d_A$  and insert into first gives

$d_B \leq 0.5$  while  $d_A = 4 - 2d_B$

To make sure the largest of them is as low as possible, best choice is

$$d_B = 0.5, \quad d_A = 3.$$

⇒ Can transmit the water by using pipe A for three days, and pipe B for half a day.

Sjekk enheter      -> hva får du når du ganger sammen?  
Les nøyte            -> Hva er relevant, hva er ikke?

3 steg for å sette opp modellen:

1. Finne antall ukjente -> les spørsmålet nøy
  2. Identifisere alle variabler og parametere med enheter -> Gi ukjente nyttige symbol
  3. Les nøyde -> Bruk info der til å sette opp

### Example 1.11: Transmitting fluid - continued

1. Number of unknowns: Two, days of using pipe A and days of using pipe B.
  2. Unknowns:  $d_A$  days,  $d_B$  days.  
Prices per day: 100 NOK/day, 400 NOK/day.  
Capacities: 1000 l/day, 2000 l/day.  
Sizes: 2 cm, 3 cm.  
Pressure drops: 100 Pa, 120 Pa.  
Budget: 500 NOK. Amount to transmit: 4000 l.
  3. Relevant parts to answer questions are budget and amount to transmit, along with prices and capacities per day.

$$\underbrace{d_A \text{ days} \times 100 \text{ NOK/day}}_{\text{NOK}} + \underbrace{d_B \text{ days} \times 400 \text{ NOK/day}}_{\text{NOK}} \leq 500 \text{ NOK.}$$

$$\underbrace{d_A \text{ days} \times 1000 \text{ l/day}}_{\text{liters}} + \underbrace{d_B \text{ days} \times 2000 \text{ l/day}}_{\text{liters}} \leq 4000 \text{ liters.}$$

## HJELPEVARIABLER

- Trenger for å formulere oppgaven og med i løsning av spørsmåletet, men ikke til den virkelige verden

### Example 1.12: Mixing alloys

The alloys A, B, C contain the following mass fractions (in %) of constituents m, n:

	A	B	C
m	11	7	12
n	14	2	44

If making 1 kg of a new mixture by combining alloys A, B and C, such that 0.5 kg of A is used and the mixture has 100 g of constituent m, how many grams will it contain of constituent n?

- S: Three alloys of different composition.
- Q: How much of constituent n in new mixture?
- M:

### Example 1.12: Mixing alloys - continued

1. Number of unknowns: One, grams of n.

2. State variable:  $M_n$  g.

System parameters: Amount of m in A:  $11\% = 0.11$  (dimensionless), etc.

Auxiliary variables: Grams of B and C in new mixture:  $M_B, M_C$ .

3. Choose to do everything using grams, to avoid mixing kg and g:

$$\text{Mass of m in new mixture: } 0.11 \times 500 + 0.07 \times M_B + 0.12 \times M_C = 100$$

$$\text{Mass of new mixture: } 500 + M_B + M_C = 1000$$

$$\text{Mass of n in new mixture: } 0.14 \times 500 + 0.02 \times M_B + 0.44 \times M_C = M_n$$

Two first gives  $M_B = 300$  g and  $M_C = 200$  g, hence the third equation gives us

$$M_n = 164 \text{ g}$$

### Warning 1.13: Number of variables and equations

As a rule of thumb: The sum of number of auxiliary variables and state variables should be equal to the number of equations.

## REPITISJON:

Modell -> noe vi bruker for å svare på spørsmål om noe      -> enkel som mulig

Matematisk modell      -> Tall involvert i modellen      -> {S, Q , M }

## MODELERINGSSKJEMA

1. Spørsmål eller problem      -> Hva skal løses?
2. System analyse      -> Hva er relevant for spørsmålet og problemet?
3. Modellering  
(numerisk modell)      -> Samle alt fra steg 2 og lag en løsningsstrategi
4. Simulering      -> Bruke modelleringsmodellen for å svare på spørsmål
5. Validering?      -> Problem løst?

### Statistical example

Mathematical models can also deal with given data as input-output system.

#### Example 1.14: Braking of a car

Given the following braking distances of a car under dry conditions<sup>1</sup>:

km/h	m
40	8
50	12.5
60	18
80	32

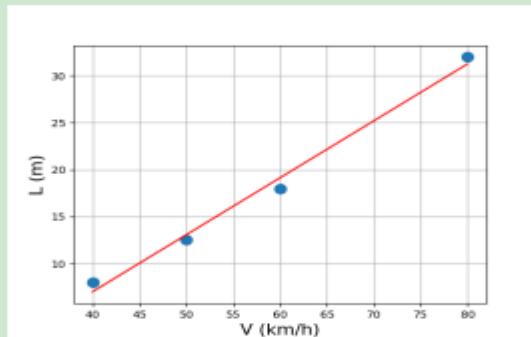
Assume a linear relation between speed and braking distance. What is the estimated braking distance for 70 km/h?

- ▶ S: Braking car.
- ▶ Q: When going 70 km/h, what is the expected braking distance?
- ▶ M: Linear regression<sup>2</sup>, use expression to find estimate for velocity of 70 km/h.

### Example 1.14: Braking of a car - continued

Performing the linear regression gives

$$L = 0.6071 \times V - 17.2857$$



For a velocity  $V = 70$  km/h, the estimate is  $L = 25.2$  m.

Example 1.14 did not require us to know anything about the car itself or how braking works. The given data were sufficient to solve the problem.

The problem could have been considered as a black box. Given the table

Input	Output
40	8
50	12.5
60	18
80	32

and assuming a linear relation between input and output, what is the expected output for an input of 70?

## TYPER AV MODELLER

### Definition 1.15: Phenomenological and mechanistic models

A mathematical model  $\{S, Q, M\}$  is called

- ▶ *phenomenological* when it is constructed from experimental data only, using no a priori information about  $S$ .
- ▶ *mechanistic* when statements in  $M$  are based on a priori information about  $S$ .

#### Phenomenological

-> "Svart boks" siden man mangler kunnskap

- Fenomenet foran oss
- Eksperimentelle data
- Input / Output data
- Lett å sette opp
- Vanskelig å generalisere

#### Mekanistiske

- Info om system for å lage hele
- Informasjon om systemet for å lage en modell

- Må vite om systemet og derfor vanskeligere å sette opp
- Gir bedre forståelse av systemet
- Lettere å generalisere
- Mer tid å sette opp

## Types of models

### Definition 1.16: Stationary and instationay models

A mathematical model  $\{S, Q, M\}$  is called

- *instationary* (or time-dependent), if at least one of its system parameters or state variables depend on time.
- *stationary* otherwise.

### Definition 1.17: Distributed and lumped models

A mathematical model  $\{S, Q, M\}$  is called

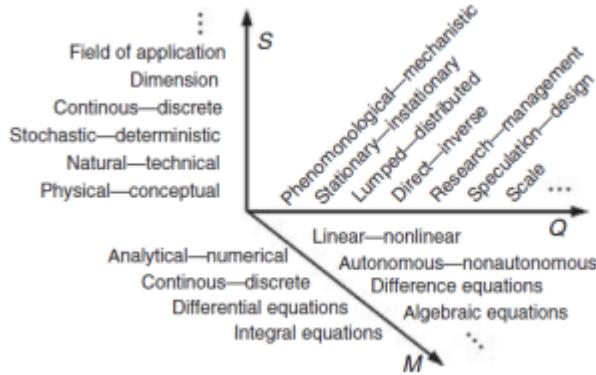
- *distributed* (or space-dependent), if at least one of its system parameters or state variables depend on a space variable.
- *lumped* otherwise.

### Example 1.18: Classify examples

Go through Examples 1.6, 1.8, 1.9, 1.10, 1.11, 1.12 and 1.14. Are they phenomenological or mechanistic? Stationary or instationary? Distributed or lumped?

We will consider in particular phenomenological models in Part 2, while we otherwise focus mainly on mechanistic models in Part 3 and 4. Instationary models will play a strong role in Part 3 (ODE models) and distributed models become important in Part 4 (PDE models).

- |                             |                         |            |
|-----------------------------|-------------------------|------------|
| 1.6 Mekanistisk             | , Instrationaly, lumped |            |
| 1.8 Phenomanlogsik/mekanisk | stasjonær               | lumped     |
| 1.9 Mekansik                | stasjonær               | Distrubert |
| 1.10 Mekansik               | Stasjonær               | lumped     |
| 1.11 Mekanisk               | Institutionary          | lumped     |
| 1.12 Mekanistisk            | Strasjnær               | lumped     |
| 1.14 Phenomenological       | Stasjonær               | lumped     |



## FEIL AV MATEMATISKE MODELLERING OG SIMULERING

### Errors of mathematical modeling and simulation

When asking a question, say the *true* answer is  $x_f$ .

We take the “detour” via a mathematical model to answer the question, and the mathematical model gives us the solution  $x_m$ .

We call the *modeling error* the difference

$$e_m = x_f - x_m.$$

#### Example 1.19: Example of modeling error

In Example 1.6, the model predicted that our skydiver had a velocity of 98 m/s (=352.8 km/t) after 10 s, while a measurement of a real skydiver would probably not be over 200 km/t (at any time). Hence, we have a substantial modeling error.

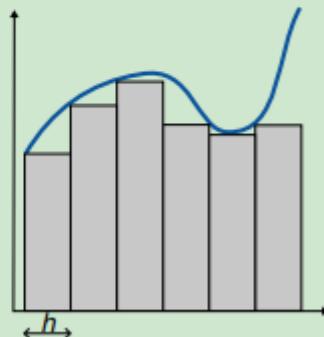
Finding  $x_m$  assumes we could perform the mathematical statements of the model exactly. Sometimes we need to use a numerical model to find an approximate solution  $x_n$ .

We call the *numerical error* the difference

$$e_n = x_m - x_n.$$

### Example 1.20: Example of numerical error

Say the mathematical statements involves calculating an integral where we cannot analytically calculate it. We then rely on approximations via e.g. Riemann sums:



When using a computer for the numerical model, we are also limited by the computer not being able to represent all numbers. Hence, the computer gave us  $x_c$ .

We call the *roundoff error* the difference

$$e_r = x_n - x_c.$$

### Example 1.21: Example of roundoff error

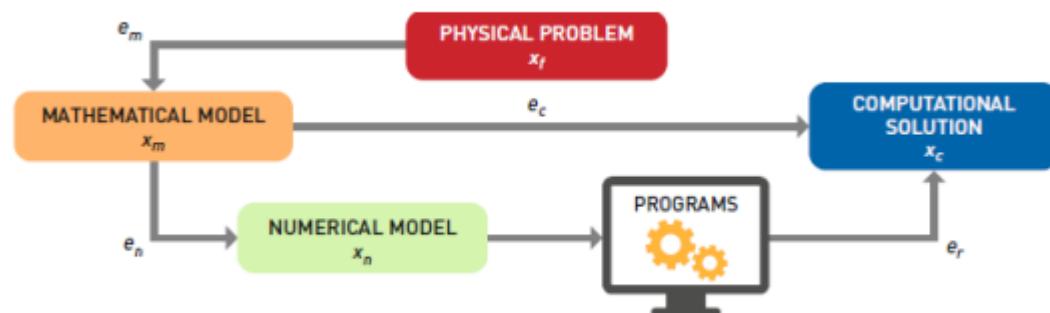
If using  $\pi$  in a calculation (e.g. volume/area of cylinder in Example 1.10), the computer cannot use the *actual* value of  $\pi$ , but has an approximate representation of it.

The sum of the numerical and roundoff error is called *computational error*:

$$e_c = e_n + e_r = x_m - x_c.$$

In the end we hope that

$$x_c \approx x_f.$$



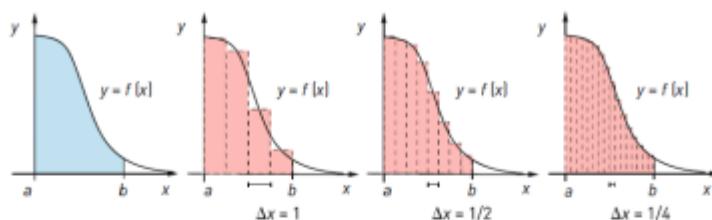
Kan vi kontrollere eller estimere feilene?

- Roundoff feil er typisk velkjent, og kommer an på pc og software -> vanligvis liten
- Numeriske feil kommer videre ann på den numeriske metoden som er brukt. Kan vanligvis bruk med å kvantifisere ved å bruke en diskret parameter  $h$
- Modelleringfeilen kommer an på hvor godt modellen forklarer et fysisk problem. Generelt vanskelig å kvantifisere

## Steg 5 VALIDERING

Vi vil validere hele modellen ved å sjekke  $x_c \approx x_f$ .

Vi ønsker å verifisere den numeriske modellen ved å undersøke  $x_n \approx x_m$ . Dette kan vi gjøre ved å undersøke feilen av den numeriske metoden



Say the numerical error  $e_n$  depends on a discretization parameter  $h$ .

### Definition 1.22: Convergence of numerical method

A numerical method is said to be *convergent*, or that it converges, when

$$e_n(h) \rightarrow 0 \text{ as } h \rightarrow 0.$$

The numerical method *converges with order p* (or has convergence of order  $p$ ) when

$$|e_n(h)| < Ch^p \text{ as } h \rightarrow 0$$

for some constant  $C$ .

Mindre  $h \rightarrow$  Mer å gjøre for datamaskinen -> Lengre tid

### Warning 1.23: Always verify the numerical method!

If you used a numerical method as part of your modeling, always verify the numerical method to make sure you can control the numerical error. Remember that

$$x_f - x_c = e_m + e_c = e_m + e_n + e_r.$$

Hence, to validate the model, it is necessary to verify the numerical model by addressing the numerical error.

To be able to make the final validation of the model, we rely on observational or experimental data of the system we are modeling.

### Warning 1.24: Verify or validate?

The terms "verification" and "validation" show up in many contexts and can be difficult to distinguish. For us the main difference can be summarized as:

- ▶ Verification answers the question "Did we make the model correctly?"
- ▶ Validation answers the question "Did we make the correct model?"

Hence, verification means investigating if each part of the model behaves as intended from a theoretical perspective. Validation means comparing to the real world to check if we get a reasonable answer.

## Regresjon og parameterestimering

Phenomenological model

Deskriptiv statistikk

Lineær regresjon

Velten kap 2.1 - 2.2 (unntatt sannsynligheter)

Hands on

*Def 4.2 Phenomenologisk modell*

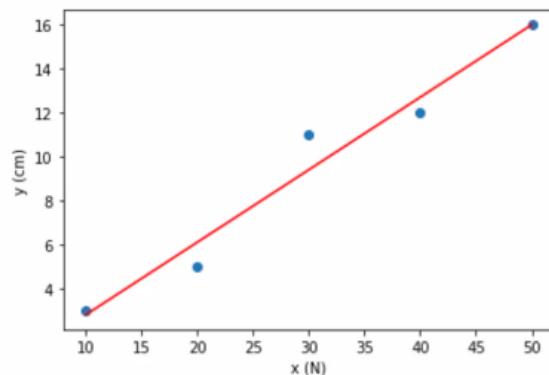
Modell basert på data og observasjoner

Det er gitt ved et sett av datapunkter, og dermed finne en funksjon  $f(x)$  som passer best

### Definition 4.2: Phenomenological model

Model based on data and observations.  
There are no a priori assumptions on the underlying mechanisms. See def 1.6.1

Given a set of data points  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ , find a function  $f(x, y)$  that best fits the data given some mathematical constraints on  $f$ .



### Discussion:

Why do we need constraints on the function  $f$ ? Can any set of observations be fitted with a polynomial?

### Fordeler

- Lett å bruke uten mye kunnskap om fenomenet av studie
- Kan lett bli brukt for statistiske tester -> lett å koble opp til statistikk

### Ulempe

- Vil kun finne ut relasjon av output og input, men kan ikke bli brukt til å gi en forklaring
- Kan være komplisert å gi relasjoner mellom input og output

### Descriptive statistics

Given a set of observations, various measures of central tendency and variability may be extracted

Given a sample of  $n$  values  $x_1, x_2, \dots, x_n$

### Central tendency

- Mean:  $\bar{x} = \frac{\sum_{i=1}^n}{n}$
- Median: The  $n/2$ -th value in the sequence of sorted  $x$ -values, (or mean of two values if  $n$  is even).

### Variability

- Standard deviation  $\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$
- Variance,  $\text{Var} = \sigma^2$

### Sample variance / standard deviation

We will only work with samples. Not complete populations. We therefore only estimate the central tendency and variability

**Covariance/correlation: To which extent are two samples varying in the same way?**

Given two equally sized sample sets of  $n$  values  $x_1, x_2, \dots, x_n$  and  $y_1, y_2, \dots, y_n$

### covariance

$\sigma_{xy} = \frac{\sum((x_i - \bar{x})(y_i - \bar{y}))}{n-1}$  Measures if the two sample sets have a similar variability.

### correlation

$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y}$  Scaled version of the covariance.

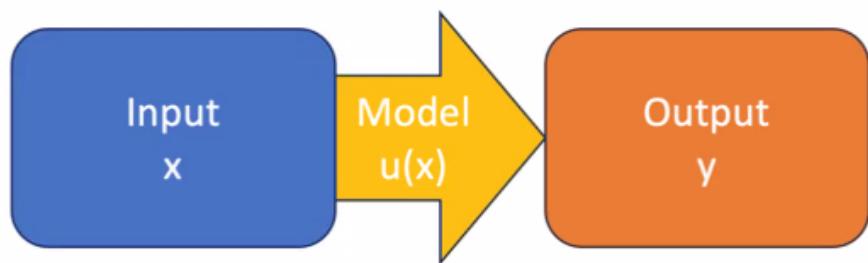
$\rho_{xy} \in [-1, 1]$ . With correlation 1, the two sample sets have a perfect positive liner relation. With correlation -1 the two sample sets has a perfect negative linear relation.

Viktige begrep i deskreptiv statestikk - MAT102

median  
mean  
Standard avvik  
Varians  
Kovarians  
Korrelasjon

## Regresjon

Can we find the mathematical relation between the two sample sets?



- ▶ Assuming there is a linear relation. Find the slope and the offset.
- ▶ Input  $x_i$  and output  $y_i$  are known, but the function is unknown.
- ▶ There exist simple formulas to find the function.
- ▶ Possible to derive using simple statistics

Linear function  $y = u(x)$      $u(x) = a_1x + a_0$ .

The two parameters in this model are the slope  $a_1$  and the constant term  $a_0$ .

Can we expect the data to fit the model perfectly?

- ▶ Inaccurate measurements
  - ▶ Noise
  - ▶ more?
- 
- ▶ Cost function: Measure how well the data fit the model:
  - ▶ In general  $C = \sum_i (u(x_i) - y_i)^2$  (sum of squared differences)

$$C(a_0, a_1) = \sum_i (a_0 x_i + a_1 - y_i)^2$$

$$\min_{a_0, a_1} C(a_0, a_1) = \min_{a_0, a_1} \sum_i (a_0 x_i + a_1 - y_i)^2$$

## Regression parameters

$$a_1 = \frac{\sum_i (x_i y_i) - n \bar{y} \bar{x}}{\sum_i (x_i - \bar{x})^2}, \quad a_0 = \bar{y} - a_1 \bar{x} \quad (2)$$

where  $\bar{x}$  and  $\bar{y}$  is the mean values of the input and output respectively.

### Note

The expression for  $a_1$  may also be expressed as  $a_1 = \frac{\sigma_{xy}^2}{s_{xx}}$ , where  $\sigma_{xy}^2$  is the variance of the  $x$ -values and  $\sigma_{xy}$  is the covariance of the  $x$  and  $y$  values.

## Utledning av lineær regresjon

Find the partial derivatives of the cost function with respect to the coefficients  $a_0$  and  $a_1$  (note, we do not differentiate w.r.t.  $x$  or  $y$ ):

$$\begin{aligned} \frac{\partial}{\partial a_0} \sum_i (a_0 x_i + a_1 - y_i)^2 &= 2 \sum_i x_i (a_0 x_i + a_1 - y_i) \\ \frac{\partial}{\partial a_1} \sum_i (a_0 x_i + a_1 - y_i)^2 &= 2 \sum_i (a_0 x_i + a_1 - y_i) \end{aligned}$$

Set each equation equal to 0 and split the sums,

$$\begin{aligned} a_0 \sum_i x_i^2 + a_1 \sum_i x_i - \sum_i x_i y_i &= 0 \\ a_0 \sum_i x_i + n a_1 - \sum_i y_i &= 0 \end{aligned}$$

The factor 2 falls out (by division).

This is a (nonlinear) system of equations with two unknowns ( $a_0$  and  $a_1$ ) and two equations. It may be solved in numerous ways.

## Partiell derivere - MAT102

We clean up the system using the definition of the mean,  
 $\bar{y} = \frac{1}{n} \sum_i y_i$  and  $\bar{x} = \frac{1}{n} \sum_i x_i$ .

$$\begin{aligned} a_0 \sum_i x_i^2 + a_1 n \bar{x} - \sum_i x_i y_i &= 0 \\ a_0 n \bar{x} + n a_1 - n \bar{y} &= 0 \end{aligned}$$

We divide the second equation by  $n$  and solve for  $a_0$ . This gives us the second of the regression formulas.

$$a_0 = \bar{y} - a_1 \bar{x} \quad (3)$$

Set this into the first equation and solve for  $a_1$ .

$$C(a_0, a_1) = \sum_i (a_0 x_i + a_1 - y_i)^2$$

$$\min_{a_0, a_1} C(a_0, a_1) = \min_{a_0, a_1} \sum_i (a_0 x_i + a_1 - y_i)^2 \quad (1)$$

## Regression parameters

$$a_1 = \frac{\sum_i (x_i y_i) - n \bar{y} \bar{x}}{\sum_i (x_i - \bar{x})^2}, \quad a_0 = \bar{y} - a_1 \bar{x} \quad (2)$$

where  $\bar{x}$  and  $\bar{y}$  is the mean values of the input and output respectively.

### Note

The expression for  $a_1$  may also be expressed as  $a_1 = \frac{\sigma_{xy}^2}{s_{xx}}$ , where  $\sigma_{xy}^2$  is the variance of the  $x$ -values and  $\sigma_{xy}$  is the covariance of the  $x$  and  $y$  values.

derivere med hensyn til  $a_0$  og  $a_1$

## Example

If we know the water temperature at 0.5m, can we make a model for the temperature at 1.5m?

[ektedata.no](http://ektedata.no)

Kode i vei - besvare spørsmålet over

## Regression

### Coefficient of determination

Proportion of the variation in the dependent variable that is predictable from the independent variable(

- ▶ Often just referred to as  $R^2$  (R squared)

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

$$SS_{\text{res}} = \sum_i (y_i - u_i)^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

$$u_i = u(x_i)$$

- ▶ Model unknown
- ▶ Samples of input ( $x$ ) and output ( $y$ ). (Independent/dependent variables)
- ▶ Assume that the model is linear
- ▶ Find the slope and the intersection in such a way that the model fits the observation.
- ▶ Plot the model and the data
- ▶ Assess visually if the model fits the data
- ▶ Compute  $R^2$
- ▶ (Perform a statistical test to determine if the slope is significantly different from 0)

Scipy.stats.linregress      -> funksjon

## Non-linear regression

In some situations it is reasonable to fit the data to a non-linear function.

- ▶ May require more parameters

### Regression vs. parameter estimation

**Regression:** Fit a curve to data points

**Parameter estimation:** Given a model with unknown parameters, use data to determine the parameters.

## Exponential growth

Differential equation:

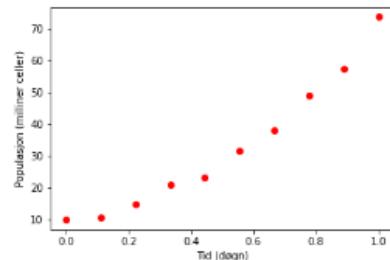
$$y'(x) = a_1 y(x)$$

$x$  is here time and  $y$  is the population size.

Analytical solution:

$$y(x) = a_0 e^{a_1 x}$$

(we will return to differential equations later)



### Trick for the exponential case (exponential regression)

$$\log y(x) = \log(a_0 e^{a_1 x})$$

$$\log y(x) = \log a_0 + a_1 x$$

Change variable:  $A_0 = \log a_0$ . Take the log of all y-values, perform linear regression, change back using  $a_0 = e^{A_0}$

$$\min_{\mathbf{a}} \sum (U_i(\mathbf{a}) - y_i)^2$$

General (non-linear) models with  $n$  model parameters

$$\mathbf{a} = [a_0, a_1, a_2 \dots a_n]$$

- ▶ In general difficult to construct an analytical solution.
- ▶ May require a numerical solution.

# Optimization

## Optimization

Assume  $A \subset \mathbb{R}^n$  is some space of parameters and that  $F : A \rightarrow \mathbb{R}$  is a function we wish to optimize.

Find some value  $\mathbf{x}_m \in A$  such that  $F(\mathbf{x}_m) < F(\mathbf{x})$  for all  $\mathbf{x} \in A$ .

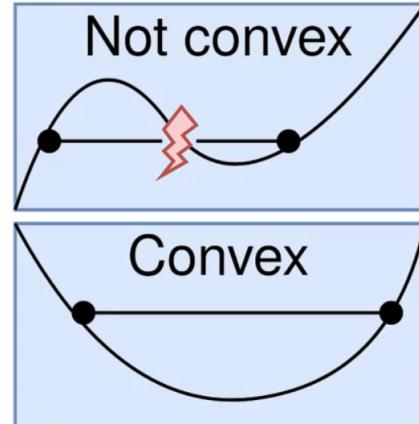
- ▶ "Optimization = minimization".  
If we want to find the maximum, we simply work with  $-F$  instead of  $F$ .
- ▶  $F$  could be any kind of function. We will work with smooth (differentiable) functions.

## Necessary condition

$$\nabla F(\mathbf{x}_m) = 0$$

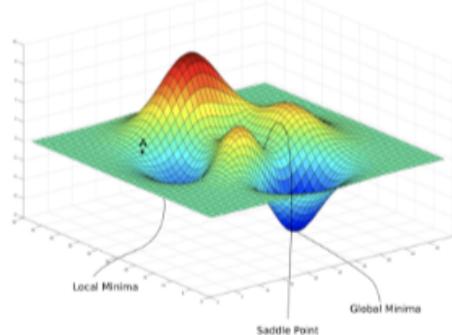
The *necessary* condition is not *sufficient*.

- ▶ We may have saddle points
- ▶ We may have a *local minima*
- ▶ A sufficient condition will involve the second (partial) derivatives



## Local minima

$\mathbf{x}_l$  is a local minima of  $F$  if  $F(\mathbf{x}_l) < F(\mathbf{x})$  for all  $\mathbf{x}$  in a small region around  $\mathbf{x}_l$ .



## Convex functions

A function is *convex* if all local minima also are a global minimum.

## Steepest descent

- Recall: The gradient points in the direction of maximum change. By following the direction of  $-\nabla F$  we will get a smaller value of  $F$ .

### Steepest descent algorithm

- Guess a starting value  $\mathbf{x}_0$
- Select a step-length  $\gamma$
- Iterate until you reach a minimum using:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma \nabla F(\mathbf{x}_n)$$

### Notation

Here  $\mathbf{x}_n$  is the  $n$ -th iteration (update) of the vector valued variable  $\mathbf{x}$ . While  $x_n$  is the  $n$ -th element in the vector  $\mathbf{x}$

## Example with one variable

$$\min_x f(x),$$

$$\text{with } f(x) = (x - 1)^2$$

$$\text{and } f'(x) = 2(x - 1)$$

.

## Example with two variables

$$\min_{x,y} F(x, y)$$

$$F(x, y) = x^2 + (y - 1)^2 + xy$$

$$\nabla F = ?$$

## Steepest descent - more general algorithm

- ▶ Set a tolerance  $tol$
- ▶ Guess a starting value  $\mathbf{x}_0$
- ▶ Iterate:
  - Estimate a steplength  $\gamma$
  - $\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma \nabla F(\mathbf{x}_n)$
  - if  $\|\nabla F(\mathbf{x}_n)\| < tol$ , break

## Stopping criteria

Are there other possible criteria for stopping the iteration?

### Recall

A norm of a vector basically measures the size/length of the vector (in some way). Any norm has the properties:

- ▶  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$
- ▶  $\|s\mathbf{x}\| = |s| \|\mathbf{x}\|$
- ▶ If  $\|\mathbf{x}\| = 0$  then  $\mathbf{x} = \mathbf{0}$

En norm er et størrelse / mål på lengde av vektor

## LEAST SQUARE PROBLEMS

-> partiell derivering

For a general least square problem,  
we need the gradient of  $F(\mathbf{a}) = \sum_i (U_i(\mathbf{a}) - y_i)^2$ .

- ▶ Recall that the function  $F$  is a function of the parameter vector  $\mathbf{a}$

$$\nabla F(\mathbf{a}) = \begin{bmatrix} \frac{\partial F}{\partial a_1} \\ \frac{\partial F}{\partial a_2} \\ \vdots \\ \frac{\partial F}{\partial a_n} \end{bmatrix} = \begin{bmatrix} 2 \sum_i (U_i(\mathbf{a}) - y_i) \frac{\partial U_i}{\partial a_1} \\ 2 \sum_i (U_i(\mathbf{a}) - y_i) \frac{\partial U_i}{\partial a_2} \\ \vdots \\ 2 \sum_i (U_i(\mathbf{a}) - y_i) \frac{\partial U_i}{\partial a_n} \end{bmatrix} = 2 \sum_i (U_i(\mathbf{a}) - y_i) \nabla U_i(\mathbf{a}) \quad (5)$$

## MER OPTIMALISERINGSTEORI

Cost function  $C : A \rightarrow \mathbb{R}$ , where  $A$  is some subset of  $\mathbb{R}^n$ .

We look for a minimum using two different solution strategies:

- ▶ Analytical solutions
- ▶ Steepest descent methods

### Conditions on the cost function

- ▶  $C$  is differentiable
- ▶  $C$  is bounded (from below)

### Example

$$C(\mathbf{x}) = x_1^2 + x_2^2$$

- ▶  $\mathbf{x}$  is distances (in meters), and can not be negative.  $\rightarrow A = \mathbb{R}_+^2 \subset \mathbb{R}^2$
- ▶  $C$  consists of two squares and can not be negative  $\rightarrow C$  is bounded from below.

### Note

Functions that don't satisfy the conditions may also be minimized, but not with the methods we use

We have not made any assumption about convexity:

#### $C$ is convex

- ▶ One unique minimum
- ▶ An iterative solver (steepest descent) will always converge, but may be slow.

#### $C$ is not convex

- ▶ May have several local minima
- ▶ The analytical solution may give a "formula" for the full set of solutions.
- ▶ A numerical solver will only give one solution. Which one will depend on the initial guess.
- ▶ For simple functions, we may use a plot to guide us towards the global minimum.

## Typical difficult situations

- ▶ Many variables
- ▶ Complicated expression
- ▶ Involving data ↗

## How to determine if a function is convex?

- ▶ Use the definition (we have not learned the details)
- ▶ Use the Hessian matrix/second derivatives. (We did not introduce this)
- ▶ **Look for known convex functions in the expression of the cost function**

## Theorem - Operations that preserve convexity

If  $f(x)$  and  $g(x)$  are convex and  $a$  is a real number.

- ▶  $f + a$  and  $f - a$  is convex
- ▶  $a f(x)$  is convex
- ▶  $f(x) + g(x)$  is convex
- ▶ If  $g(x)$  in addition is non-decreasing (increasing),  $g(f(x))$  is convex.

## Functions known to be convex

Linear functions       $x^2$        $e^x$

### Example

$$C(\mathbf{x}) = x_1^2 + x_2^2$$

$C$  is a sum of two squares  $\rightarrow C$  is convex.

### Example

$$C(\mathbf{x}) = (e^{x_1} + e^{x_2})^2$$

$e^{x_1} + e^{x_2}$  is convex. The power operation ( $x^2$ ) is also convex and increasing for  $x > 0$ . In this case the argument is always  $> 0 \rightarrow C(\mathbf{x})$  is convex

### Example

$$C(\mathbf{x}) = \sin(x_1 + x_2)^2$$

$\sin$  is not convex. the power does not help  $\rightarrow C(\mathbf{x})$  may not be convex

The arguments we have made will only work one way. I.e. we can only use it to show that something **is** convex. There are many examples of functions not satisfying the conditions above, that are still convex.

### Example

$$C(\mathbf{x}) = \sqrt{(x_1 - a_1)^2 + (x_2 - a_2)^2}$$

is convex, (even if the square root is not a convex operator)

### Norms

All norms are convex

## PYTHON

- ▶ `scipy.optimize`
- ▶ equivalent methods in python as in R (the textbook)

### Sin-function and temperature data

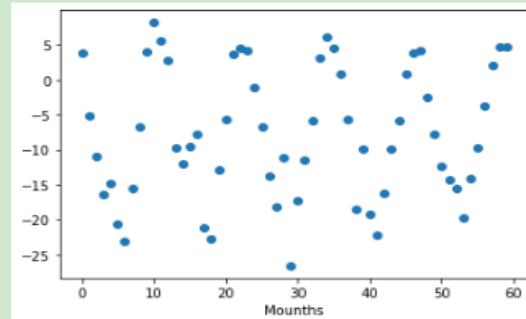
**Fit a periodic function to the temperature data from Svalbard lufthavn.**

- ▶ We select only a part of the data (5 years),  $(x_i, y_i)$ ,  $i = 1, 2, \dots$
- ▶ Set up a parameterized function

$$F(\mathbf{a}, x) = a_0 \sin(a_1 x + 2) + a_3$$

- ▶ Set up a least squares cost function:

$$C(\mathbf{a}) = \sum_i (F(\mathbf{a}, x_i) - y_i)^2$$



### Sin-function and temperature data

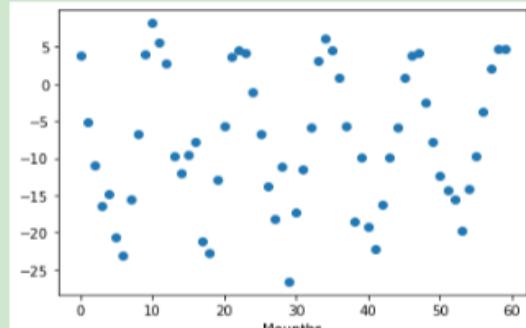
**Fit a periodic function to the temperature data from Svalbard lufthavn.**

- ▶ Compute the gradient of  $F$ :

$$\nabla_{\mathbf{a}} F = \begin{bmatrix} \sin(a_1 x + 2) \\ a_0 x \cos(a_1 x + a_2) \\ a_0 \cos(a_1 x + a_2) \\ 1 \end{bmatrix}$$

- ▶ Set up the gradient of  $C$ :

$$2 \sum_i (F(\mathbf{a}, x_i) - y_i) \nabla F(\mathbf{a}, x_i)$$



## Derivative-free methods

### Gradient based methods

- ▶ Requires an analytical expression for the derivatives
- ▶ Converge fast

### Derivative-free methods

- ▶ Only require a cost function
- ▶ Converge slower (particular for problems with many variables)
- ▶ May work even if the function is not smooth or discontinuous.
- ▶ Approximate the derivative using finite difference or search in a special pattern.

## Constrained optimization

We may be in a situation where We have some conditions on the parameters

$$\begin{aligned} & \min_{\mathbf{x}} C(\mathbf{x}) \\ \text{s.t. } & g_j(\mathbf{x}) < 0 \\ & h_k(\mathbf{x}) = 0 \end{aligned}$$

### Example

$$F(x, y) = x^2 + (y - 1)^2 + xy$$

$$\begin{aligned} & \min F(x, y) \\ \text{s.t. } & x + y = 1 \end{aligned}$$

### Solvers

There are several methods to handle the constraints. We will only use optimization software capable of handling the constraints as input.

## Constrained optimization

### Role of constraints

The constraints may have several roles:

- ▶ Enforcing a physical (or logical) relation. E.g: Weights should not be negative
- ▶ Guiding the solver to a reasonable regions.

### Example

$$F(\mathbf{a}, x) = a_0 \sin(a_1 x + 2) + a_3$$

$$C(\mathbf{a}) = \sum_i (F(\mathbf{a}, x_i) - y_i)^2$$

$$\min C(\mathbf{a})$$

$$\text{s.t. } a_1 < 1$$

$$a_1 > 0.3$$

## Summary optimization and parameter estimation

- ▶ Fitting data to a nonlinear curve - Non-linear regression
- ▶ The curve may be expressed with a set of parameters  
→ parameter estimation
- ▶ We set up a cost function

$$C(\mathbf{a}) = \sum_i (F(\mathbf{a}, x_i) - y_i)^2$$

where  $F$  is a curve ( $F(x)$ ) with parameters  $\mathbf{a}$

- ▶ We minimize  $C$  with respect to  $\mathbf{a}$
- ▶ Various optimization methods may be used
  - ▶ Steepest descent
  - ▶ Pre-implemented libraries (scipy)
  - ▶ Methods using the gradient are generally faster than methods only using the cost function
- ▶ We can not generally assume that  $C$  is convex
  - ▶ Several local minimum may exist. The solution will depend on initial guess.

# Ordinære differensialligninger (ODE)

## Motivasjon

In this part we will go through the role of ordinary differential equations (ODEs) in mathematical modeling, with a special focus on how to solve ODEs numerically.

Questions that we will address are

- ▶ When do we need a model with ODEs and how do we create an ODE model?
- ▶ How to classify common types of ODEs?
- ▶ When is an ODE model well-posed?
- ▶ How to solve ODEs numerically and what is the error by doing that?
- ▶ How to interpret the solution of ODEs?
- ▶ How to use ODEs in modeling and simulating physical phenomena?

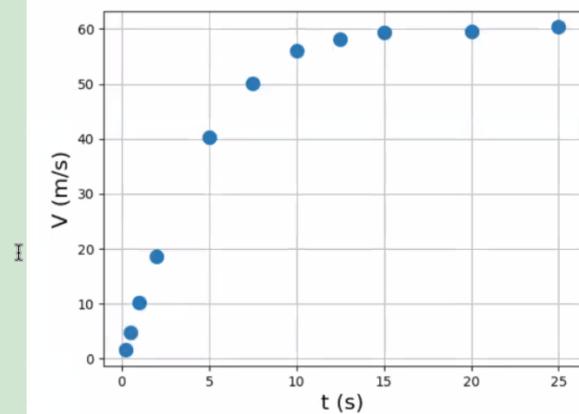
This part is based on Chapter 3 of the book of Kai Velten, and Section 2.4.5 of the book of Hans-Joachim Bungartz et al. Also note there are some supplementary information in the slides.

We go back to the skydiver from Example 1.6.

### Example 3.1: Skydiving

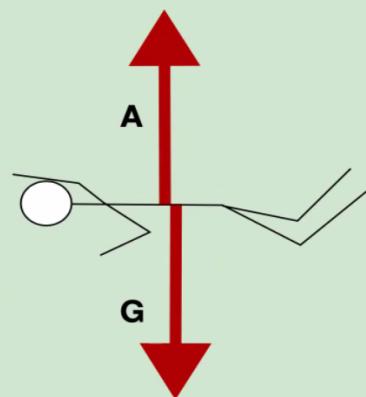
We are given the following observation data of velocities after various times:

s	m/s
0.2	1.56
0.5	4.73
1	10.2
2	18.6
5	40.2
7.5	50.2
10	56.0
12.5	58.1
15	59.3
20	59.6
25	60.3



Instead of doing fitting, we will try to come up with an ODE that can explain these data.

Our original attempt in Example 1.6 predicted  $v = 9.8t$ , which corresponds to linear increase in velocity with time. Perhaps suitable for first 2 seconds. Not only gravity is active: also air resistance. Set up forces and use Newton's second law.



Use positive direction downwards. Two forces, gravity and air resistance. Since velocity increase gradually smaller, air resistance appears to increase with velocity. Try a linear relationship:

$$\underline{\mathbf{G}} = mg\mathbf{k}$$

$$\underline{\mathbf{A}} = -\gamma v\mathbf{k}$$

for some proportionality constant  $\gamma$ . Then,

$$\sum \mathbf{F} = m\mathbf{a} = m \frac{dv}{dt} \mathbf{k} = mg\mathbf{k} - \gamma v\mathbf{k}$$

We hence get an ODE for velocity, and arrive at an initial value problem we can solve:

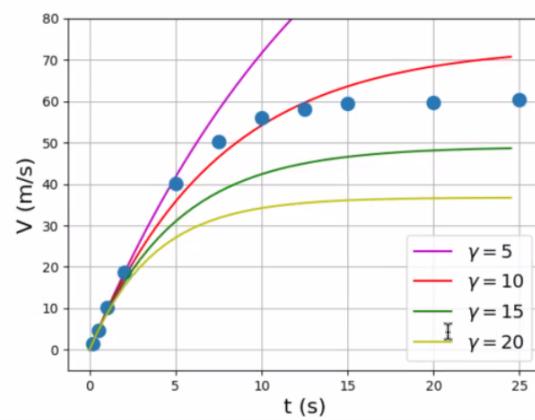
$$\frac{dv}{dt} = g - \frac{\gamma}{m} v, \quad v(0) = 0.$$

Solve for  $v(t)$  and find

$$v(t) = \frac{mg}{\gamma} \left( 1 - e^{-\frac{\gamma t}{m}} \right).$$

What is  $\gamma$ ? For an example person of  $m = 75$  kg, can try to adjust to match observation data.

No obvious good fit.



A possible adaption is to use a different expression for the air resistance. Since linear air resistance was too strong early on, can try quadratic relationship:

$$\mathbf{A} = -\gamma v^2 \mathbf{k},$$

for a (different) proportionality constant  $\gamma$ . Then

$$\mathbf{F} = m a \mathbf{k} = m \frac{dv}{dt} \mathbf{k} = mg \mathbf{k} - \gamma v^2 \mathbf{k}$$

and we get the initial value problem

$$\frac{dv}{dt} = g - \frac{\gamma}{m} v^2, \quad v(0) = 0,$$

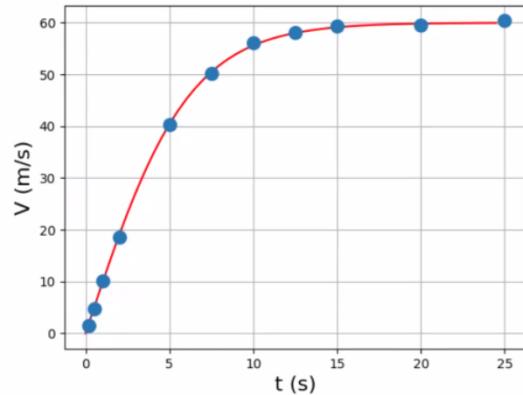
which has solution

$$v(t) = \sqrt{\frac{mg}{\gamma}} \tanh\left(\sqrt{\frac{\gamma g}{m}} t\right).$$

To find  $\gamma$ , use again that terminal velocity appears to be  $v_\infty = 60$  m/s. Then the new ODE gives

$$\gamma = \frac{mg}{v_\infty^2} \approx 0.204$$

for our example person.



This model appears to describe the observed data very well.

## RULES OF DIFF LIGNING

- ▶ Working with functions and derivatives opens a new mathematical world with many possibilities.
- ▶ Derivatives and hence differential equations allow us to accurately describe more types of dependencies.
- ▶ Setting up an ODE for a system reveals mechanisms.
- ▶ Appearing parameters can be interpreted through the ODE instead of being purely a fitting parameter. Opens for measuring the relevant parameters.
- ▶ Mechanistic models gives us theoretical insight that a pure phenomenological model cannot.
- ▶ Observations are needed to confirm or disregard the mathematical model.

Assuming we found the correct ODE model to describe the skydiver, we learned the following:

- ▶ Air resistance of a skydiver is initially small, but becomes gradually stronger as velocity increases.
- ▶ The size of the air resistance scales with velocity squared.
- ▶ There is a terminal velocity where the air resistance has same size as the gravity force.
- ▶ The proportionality constant in the expression for the air resistance can be easily found if we can estimate the terminal velocity.

We however have no proof that this is true. For that we would still need to measure the air resistance directly. But given that the model describes the observed data so well, we can adopt it until other observation data force us to disregard this model.

### Strategi for formulering av ODE

We can separate between three approaches for formulating ODE models:

- ▶ Phenomenological
  - ▶ Setting up a function that fit the data, and then find an ODE that this expression fulfills.
- ▶ Theoretical
  - ▶ Read up on the underlying physics and use to formulate ODE.
- ▶ Rate of change
  - ▶ From observed changes in data, estimate relations for rate of change and connections to other observed changes.

In Example 3.1, we used Newton's second law (theoretical) together with the behavior of the velocity data (rate of change) to set up an ODE.

### Typer diff ligning

Let's say you found an ODE for your problem.

- ▶ What type of ODE is it?
- ▶ Can we expect a unique solution?
- ▶ If yes, how to find it?

## Klassifisering av diff ligning - hvilken orden?

The order of an ODE refers to the order of its highest derivative that appears:

### Definition 3.3: Order of an ODE

Given a function  $F$ , which is a function of  $t$ ,  $y$  and derivatives of  $y$  with respect to  $t$ , then an ODE for  $y$  of order  $n$  is of the form

$$F(t, y, y', y'', \dots, y^{(n-1)}, y^{(n)}) = 0$$

where  $y^{(n)}$  has to appear in the above equation.

We will generally assume that  $F$  is a continuous function, but this is technically not a requirement. A solution of the ODE is any function  $y(t)$  that fulfill the above equation.

### Example 3.4: Order of ODEs

- ▶  $v'(t) = g - \frac{\gamma}{m}(v(t))^2$  is a first-order ODE.
- ▶  $y''(t) = -y(t) + \sin t$  is a second-order ODE.
- ▶  $y^{(3)}(x) = a_0(x)y(x) + a_1(x)y'(x) + a_2(x)y''(x)$  is a third order ODE.
- ▶  $\tan(y^{(4)}(t)) = (y(t))^2$  is a fourth order ODE.

### Definition 3.5: Explicit and implicit ODE

An ODE for  $y$  of order  $n$  is said to be explicit if it can be written on the form

$$y^{(n)}(t) = f(t, y, y', y'', \dots, y^{(n-1)}).$$

Otherwise the ODE is implicit.

### Example 3.6: Explicit and implicit ODEs

- ▶  $v'(t) = g - \frac{\gamma}{m}(v(t))^2$  is an explicit ODE.
- ▶  $y''(t) = -y(t) + \sin t$  is an explicit ODE.
- ▶  $y^{(3)}(x) = a_0(x)y(x) + a_1(x)y'(x) + a_2(x)y''(x)$  is an explicit ODE.
- ▶  $\tan(y^{(4)}(t)) = (y(t))^2$  is an implicit ODE.

### Warning 3.7: Explicit ODEs

We will mainly deal with explicit ODEs in this course. Normally, explicit ODEs are just called "ODEs".

### Definition 3.8: Autonomous ODE

An ODE for  $y$  of order  $n$  is said to be autonomous if the dependence on the variable  $t$  is only through  $y$  and derivatives of  $y$ . That is, if

$$F(y, y', y'', \dots, y^{(n-1)}, y^{(n)}) = 0.$$

Otherwise the ODE is nonautonomous.

### Example 3.9: Autonomous ODEs

- ▶  $v'(t) = g - \frac{\gamma}{m}(v(t))^2$  is an autonomous ODE as  $g, \gamma, m$  are all constants.
- ▶  $y''(t) = -y(t) + \sin t$  is a nonautonomous ODE.
- ▶  $y^{(3)}(x) = a_0(x)y(x) + a_1(x)y'(x) + a_2(x)y''(x)$  is a nonautonomous ODE.
- ▶  $\tan(y^{(4)}(t)) = (y(t))^2$  is an autonomous ODE.

### Definition 3.10: Linear ODE

An ODE for  $y$  of order  $n$  is said to be linear if  $y$  and derivatives of  $y$  can be written as linear combinations. That is, if the ODE can be written

$$y^{(n)}(t) = b(t) + a_0(t)y(t) + a_1(t)y'(t) + \cdots + a_{n-1}(t)y^{(n-1)}(t).$$

Otherwise the ODE is nonlinear.

### Warning 3.11: Linear ODE

Note that the functions  $b(t), a_0(t), a_1(t), \dots, a_{n-1}(t)$  do **not** have to be linear. In general, they can be any continuous function.

### Example 3.12: Linear ODEs

- ▶  $v'(t) = g - \frac{c}{m}(v(t))^2$  is a nonlinear ODE.
- ▶  $y''(t) = -y(t) + \sin t$  is a linear ODE.
- ▶  $y^{(3)}(x) = a_0(x)y(x) + a_1(x)y'(x) + a_2(x)y''(x)$  is a linear ODE.
- ▶  $\tan(y^{(4)}(t)) = (y(t))^2$  is a nonlinear ODE.

We will also work with system of ODEs:

### Definition 3.13: System of ODEs

When  $m$  functions  $y_1, y_2, \dots, y_m$  and their derivatives are coupled so that they can be written

$$\begin{aligned} y_1^{(n)} &= f_1(t, \mathbf{y}, \mathbf{y}', \mathbf{y}'', \dots, \mathbf{y}^{(n-1)}) \\ y_2^{(n)} &= f_2(t, \mathbf{y}, \mathbf{y}', \mathbf{y}'', \dots, \mathbf{y}^{(n-1)}) \\ &\vdots \\ y_m^{(n)} &= f_m(t, \mathbf{y}, \mathbf{y}', \mathbf{y}'', \dots, \mathbf{y}^{(n-1)}), \end{aligned}$$

where  $\mathbf{y} = [y_1(t), y_2(t), \dots, y_m(t)]$ , we have an (explicit) system of ODEs of order  $n$  and dimension  $m$ .

This system of ODEs can also be written using matrix notation

$$\mathbf{y}^{(n)} = \mathbf{f}(t, \mathbf{y}, \mathbf{y}', \dots, \mathbf{y}^{(n-1)})$$

where  $\mathbf{f} = [f_1, f_2, \dots, f_m]$ . There is also an implicit analogue for system of ODEs.

Note that all definitions covered in the previous slides can be directly applied to a system of ODEs by extending the definitions to vector notation. Note that a system of linear first order ODEs can be written in matrix form

$$\mathbf{y}'(t) = \mathbf{A}(t)\mathbf{y}(t) + \mathbf{b}(t)$$

for  $\mathbf{y}(t) \in \mathbb{R}^m, \mathbf{A}(t) \in \mathbb{R}^{m \times m}, \mathbf{b}(t) \in \mathbb{R}^m$ .

### Example 3.14: System of ODEs

The SIR model (S, I, R = susceptible, infected, recovered) for modeling infectious diseases is given by

$$\frac{d}{dt} \begin{bmatrix} S(t) \\ I(t) \\ R(t) \end{bmatrix} = \begin{bmatrix} -\frac{\beta}{N}SI \\ \frac{\beta}{N}SI - \gamma I \\ \gamma I \end{bmatrix}$$

for given constants  $\beta, \gamma, N$ . This is an (explicit) first order system of ODEs of dimension 3. The system is autonomous and nonlinear.

## System av ODE

Every explicit ODE of order  $n$  can be written into a system of  $n$  first order ODEs. Given

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)}),$$

we can define a vector  $\mathbf{u} = [u_1, u_2, \dots, u_n] = [y, y', \dots, y^{(n-1)}]$  which then fulfills

$$\mathbf{u}'(t) = \frac{d}{dt} \begin{bmatrix} y \\ y' \\ \vdots \\ y^{(n-2)} \\ y^{(n-1)} \end{bmatrix} = \begin{bmatrix} y'(t) \\ y''(t) \\ \vdots \\ y^{(n-1)} \\ y^{(n)} \end{bmatrix} = \underbrace{\begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_n \\ f(t, u_1, u_2, \dots, u_n) \end{bmatrix}}_{\mathbf{f}(t, \mathbf{u})}.$$

Hence, as long as we restrict ourselves to explicit ODEs, we can always rewrite them to a system of first order ODEs.

### Example 3.15: Rewrite to system of ODEs

The third order ODE  $y^{(3)}(x) = a_0(x)y(x) + a_1(x)y'(x) + a_2(x)y''(x)$  can be rewritten to a system of first order ODEs by defining

$$\mathbf{u}(x) = \begin{bmatrix} u_1(x) \\ u_2(x) \\ u_3(x) \end{bmatrix} = \begin{bmatrix} y(x) \\ y'(x) \\ y''(x) \end{bmatrix}.$$

Then,

$$\mathbf{u}'(x) = \frac{d}{dx} \begin{bmatrix} y(x) \\ y'(x) \\ y''(x) \end{bmatrix} = \begin{bmatrix} y'(x) \\ y''(x) \\ y'''(x) \end{bmatrix} = \begin{bmatrix} u_2(x) \\ u_3(x) \\ a_0(x)u_1(x) + a_1(x)u_2(x) + a_2(x)u_3(x) \end{bmatrix}$$

That means we can stick to (system of) first order ODEs when discussing properties and numerical methods of explicit ODEs. Hence, keep in mind that explicit higher order ODEs can always be written into a system of first order ODEs!

## INITIELL VERDI PROBLEM      -> IVP

Typically when dealing with first order ODEs, we are also given a starting point or initial value which the solution of the ODE is given to fulfill.

### Definition 3.16: Initial value problem

An initial value problem (IVP) is an ODE together with an initial condition which the unknown function in the ODE should fulfill. For a first order ODE, this means

$$u'(t) = f(t, u) \text{ for } f : D \rightarrow \mathbb{R}^m, \quad u(t_0) = u_0 \text{ where } (t_0, u_0) \in D$$

Note that this definition works straightforwardly for both scalar- and vector-valued  $u$ .

A solution of an IVP is any function  $u(t)$  that fulfills both the ODE and the initial condition.

### Example 3.17: Skydiver

The skydiving example included an IVP as we had to solve

$$\frac{dv}{dt} = g - \frac{\gamma}{m} v^2, \quad v(0) = 0.$$

Here, a possible domain can be  $D = [0, \infty) \times [0, \infty)$  as  $t, v$  meaningful for positive times and speeds.

### Example 3.18: SIR model

If we are given values for  $S, I, R$  at an initial time, so that we have

$$\frac{d}{dt} \begin{bmatrix} S(t) \\ I(t) \\ R(t) \end{bmatrix} = \begin{bmatrix} -\frac{\beta}{N} SI \\ \frac{\beta}{N} SI - \gamma I \\ \gamma I \end{bmatrix}, \quad \begin{bmatrix} S(0) \\ I(0) \\ R(0) \end{bmatrix} = \begin{bmatrix} S_0 \\ I_0 \\ R_0 \end{bmatrix},$$

we have an IVP. Here a possible domain is  $D = [0, \infty) \times \mathbb{R}_{\geq 0}^3$  as only positive values make sense.

## BOUNDARY VALUE PROBLEMS      -> BVP

There are a large range of different boundary value problems (BVPs), and we will here in the ODE part only consider a special case, namely a two-point BVP. Then some information about the solution is given at two different points in time.

### Definition 3.19: Boundary value problem

Given an ODE  $u'(t) = f(t, u)$  for  $f : \mathbb{I} \times \Omega$  where  $\mathbb{I} = [t_1, t_2]$ , then a (two-point) boundary value problem is

$$u'(t) = f(t, u), \quad r(u(t_1), u(t_2)) = 0 \text{ where } u(t_1), u(t_2) \in \Omega,$$

where  $r$  is some given relation defined on  $\Omega \times \Omega$ . The BVP is linear when  $f$  and  $r$  are linear with respect to  $u$ . That is, if

$$u'(t) = A(t)u(t) + b(t), \quad R_1 u(t_1) + R_2 u(t_2) - r_0 = 0$$

for  $A(t) \in \mathbb{R}^{m \times m}$ ,  $b(t) \in \mathbb{R}^m$  continuous functions and  $R_1, R_2 \in \mathbb{R}^{m \times m}$ ,  $r_0 \in \mathbb{R}^m$ .

A solution of a BVP is any function  $u(t)$  fulfilling both the ODE and the boundary conditions.

### Example 3.20: Throwing a ball

You throw a ball (without air resistance) with a starting velocity  $\mathbf{v}_0$  m/s and it lands 100 m away from you 10 seconds later. What trajectory did it follow?

This can be formulated as a BVP. We use as unknown the position of the ball  $[x_1(t), x_2(t)]$ , where  $x_2(t)$  is the vertical coordinate. When only gravity is acting, we have the ODE

$$\frac{d^2}{dt^2} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{pmatrix} 0 \\ -g \end{pmatrix}.$$

The starting velocity and landing information tells us that

$$\begin{bmatrix} x'_1(0) \\ x'_2(0) \end{bmatrix} = \mathbf{v}_0 \text{ and } \begin{bmatrix} x_1(10) \\ x_2(10) \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \end{bmatrix}.$$

This is not written entirely according to the Definition 3.19. Rewrite the ODE to a system of four first order ODEs. Is the resulting BVP linear?

EKSMEPEL 3.20

Fra oppgaven

$$\frac{d^2}{dt^2} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{pmatrix} 0 \\ -g \end{pmatrix}.$$

dag 14. januar 2022

10:33

$$\ddot{\bar{x}} = \begin{bmatrix} 0 \\ -g \end{bmatrix}$$

$$\begin{bmatrix} x_1(10) \\ x_2(10) \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \end{bmatrix}.$$

$$\bar{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \bar{x}(10) = \begin{bmatrix} 100 \\ 0 \end{bmatrix}$$

Oppretter nye første ordens ligningssystem

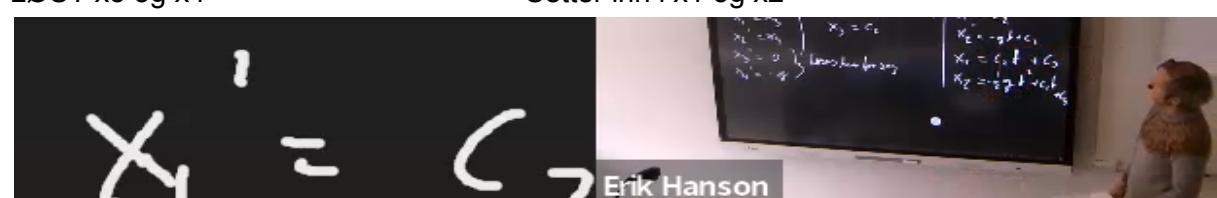
$$\begin{array}{l} \overset{\text{"}}{x_1} = 0 \quad (\text{aks i horisontal retn}) \\ \overset{\text{"}}{x_2} = -g \quad (-\text{"} \text{ vertikal } \text{"}) \\ \overset{\text{'}}{x_3} = \overset{\text{'}}{x_1} \quad (\text{fort i h. retn.}) \\ \overset{\text{'}}{x_4} = \overset{\text{'}}{x_2} \quad (" \text{ i v. } ") \end{array}$$

$$\begin{array}{l} \overset{\text{'}}{x_1} = \overset{\text{'}}{x_3} \\ \overset{\text{'}}{x_2} = \overset{\text{'}}{x_4} \\ \overset{\text{'}}{x_3} = 0 \\ \overset{\text{'}}{x_4} = -g \end{array}$$

$$\begin{aligned} x_1' &= x_3 \\ x_2' &= x_4 \\ x_3' &= 0 \\ x_4' &= -gt \end{aligned} \quad \left. \begin{array}{l} x_4 = -gt + c_1 \\ x_3 = c_2 \\ \text{lores hør for seg} \end{array} \right\}$$

LØST x3 og x4

-&gt; Setter inn i x1 og x2



$x_1' = c_2$

$x_2' = -gt + c_1$

$x_1 = c_2 t + c_3$

$x_2 = -\frac{1}{2}gt^2 + c_1 t + c_4$

Betingelsene:

$$\begin{array}{ll} x_1(10) = 100 & x_1(0) = 0 \\ x_2(10) = 0 & x_2(0) = 0 \end{array}$$

$$\begin{aligned} x_1(0) &= C_2 * 0 + C_3 = 0 & \rightarrow C_3 = 0 \\ x_2(0) &= -\frac{1}{2}g * 0^2 + C_1 * 0 + C_4 = 0 & \rightarrow C_4 = 0 \end{aligned}$$

$$\begin{aligned} x_1(10) &= C_2 * 10 = 100 & \rightarrow C_2 = 10 \\ x_2(10) &= -\frac{1}{2}g * 10^2 + C_1 * 10 & \rightarrow C_1 = 5g \end{aligned}$$

$$\bar{x}(t) = \begin{bmatrix} 10t \\ -\frac{1}{2}gt^2 + 5gt \end{bmatrix}$$

**OBS :** Vi trenger ikke V<sub>0</sub> nå!

## Veldefinerte problem

## - Well-posedness

### Definition 3.21: Well-posedness

A model problem involving differential equations is said to be well-posed when the following conditions are fulfilled:

- *existence*: there is a solution.
- *uniqueness*: there is only one solution.
- *stability*: the solution depends continuously on the data of the problem.

The “data” of the problem can be parameters or coefficient functions appearing in the ODE’s right-hand side, or values for initial/boundary conditions. Hence, a small change of any of these should only give a small change in the solution.

There is a classical Theorem, stating the conditions of a **unique** solution of an ODE:

### Theorem 3.22: Picard-Lindelöf

Given the initial value problem

$$u'(t) = f(t, u), \quad u(t_0) = u_0.$$

If  $f(t, u)$  is continuous in the first argument, and Lipschitz-continuous in the second argument, then the initial value problem has a unique solution  $u(t)$  in a neighborhood around  $t_0$ .

### Lipschitz-continuity

A more strict term than **continuous**, but as strict as **continuously differentiable**

$\exists L > 0$  such that  $\|f(t, u) - f(t, \tilde{u})\| \leq L \|u - \tilde{u}\|$   
for all  $u, \tilde{u}$  in a neighborhood around  $u_0$  and all  $t$  in a neighborhood around  $t_0$

$$\left| f(x) - f(\tilde{x}) \right| \leq L |x - \tilde{x}|$$
$$\left| \frac{f(x) - f(\tilde{x})}{x - \tilde{x}} \right| \leq L$$

Det vil si et område uten *uendelig* som stigningstall

- The theorem also applies to system of ODEs.
- For a single ODE we can use absolute values instead of norms.
- If  $f(t, u)$  is continuously differentiable with respect to  $u$ , it is Lipschitz-continuous.
- In its current form, the theorem only tells us there will be “some” time interval where we find a unique solution - not how large it is.

### Extension of the theorem

The theorem is more precise if we look at a bounded regions for  $t$  and bounded functions  $f$ .

If  $f$  is Lipschitz-continuous (or continuously differentiable) and bounded for all  $t \in [a, b]$  and for all  $u \in \mathbb{R}^m$ , then there will be a unique solution  $u(t)$  for  $t \in [a, b]$ .

### Example 3.18: SIR model - continued

In the SIR model, we had the right-hand side

$$\mathbf{f}(S, I, R) = \begin{bmatrix} -\frac{\beta}{N}SI \\ \frac{\beta}{N}SI - \gamma I \\ \gamma I \end{bmatrix}.$$

Although only positive values makes sense for the IVP, we have that  $\mathbf{f}$  itself is continuously differentiable and therefore Lipschitz continuous for all  $\begin{bmatrix} S \\ I \\ R \end{bmatrix} \in \mathbb{R}^3$ .

But  $\mathbf{f}$  is not bounded for all  $\begin{bmatrix} S \\ I \\ R \end{bmatrix} \in \mathbb{R}^3$ . Therefore we can only guarantee existence of a unique solution in a neighborhood around the initial time  $t_0 = 0$ .

### Example 3.15: Linear system of ODEs - continued

If adding an initial condition such that we have the IVP

$$\mathbf{u}'(x) = \begin{bmatrix} u'_1(x) \\ u'_2(x) \\ u'_3(x) \end{bmatrix} = \begin{bmatrix} u_2(x) \\ u_3(x) \\ a_0(x)u_1(x) + a_1(x)u_2(x) + a_2(x)u_3(x) \end{bmatrix}, \quad \begin{bmatrix} u_1(x_0) \\ u_2(x_0) \\ u_3(x_0) \end{bmatrix} = \mathbf{u}_0$$

and assuming that  $a_0, a_1, a_2$  are continuous functions for  $x \in [x_1, x_2]$  and that  $x_0 \in [x_1, x_2]$ , we have that all conditions of Theorem 3.22 are fulfilled. Also the extension applies, hence the IVP has a unique solution  $\mathbf{u}(x)$  for all  $x \in [x_1, x_2]$ .

### Warning 3.23: Linear ODE

IVPs with linear ODEs or system of linear ODEs will always fulfill the conditions of Theorem 3.22 and will therefore have a unique solution for the entire interval where the weight functions ( $b(t), a_0(t), \dots$  or  $\mathbf{A}(t)$ ) are continuous. If the weight functions are continuous on entire  $\mathbb{R}$ , then there will be a unique solution defined on entire  $\mathbb{R}$ .

### Example 3.24: Ill-posed problem

The IVP

$$y'(t) = \sqrt{|y - 1|}, \quad y(t_0) = y_0$$

does not fulfill Theorem 3.22 if  $y_0 = 1$ .

- a) If  $y_0 \neq 1$ , then is  $f(t, y) = \sqrt{|y - 1|}$  continuously differentiable with respect to  $y$  in a neighborhood around  $y_0$  (as long we do not get close to  $y = 1$ ). Hence, unique solution is expected near  $t_0$ .
- b) If  $y_0 = 1$ ,  $f(t, y)$  is not Lipschitz continuous in the needed neighborhood. With  $\tilde{y} = 1$ , have that

$$\frac{|f(t, y) - f(t, 1)|}{|y - 1|} = \frac{\sqrt{|y - 1|}}{|y - 1|} \rightarrow \infty \text{ as } y \rightarrow 1$$

Hence, not possible to find a fixed  $L$  such that  $|f(t, y) - f(t, \tilde{y})| \leq L|y - \tilde{y}|$  for all  $y, \tilde{y}$  in a neighborhood around  $y_0 = 1$ .

### Example 3.24: Ill-posed problem - continued

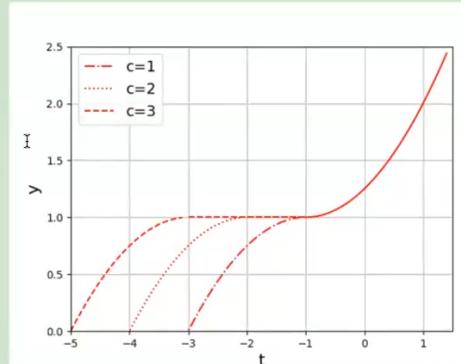
Solving

$$y'(t) = \sqrt{|y - 1|}, \quad y(1) = 2$$

using separation of variables gives the solution

$$y(t) = \begin{cases} 1 + \frac{1}{4}(t + 1)^2 & \text{for } t > -1 \\ 1 & \text{for } -c \leq t \leq 1 \\ 1 - \frac{1}{4}(t + c)^2 & \text{for } t < -c \end{cases}$$

for any  $c \in \mathbb{R}$ .



Hence, we see that we only get a unique solution when we are staying away from  $y = 1$ .

### Theorem 3.25: Stability of IVPs

When the same continuity conditions as in Theorem 3.22 are fulfilled, the unique solution of the IVP is **stable** with respect to the initial condition.

- ▶ If considering two slightly different initial conditions  $u_0$  and  $\tilde{u}_0$ , we only get a small difference between the solutions.
- ▶ In mathematical terms:  
If

$$\begin{aligned} u'(t) &= f(t, u), & u(t_0) &= u_0, \\ \tilde{u}'(t) &= f(t, \tilde{u}), & \tilde{u}(t_0) &= \tilde{u}_0, \end{aligned}$$

then

$$\|u(t) - \tilde{u}(t)\| \leq e^{L(t-t_0)} \|u_0 - \tilde{u}_0\|$$

for any  $t$  inside the interval where the IVP has a unique solution.

- ▶ The theorem applies to both single ODEs and systems of ODEs.
- ▶ Similar results can be formulated also for small changes in  $f$  and  $t_0$ . Hence, we have stability when the conditions of Theorem 3.22 are fulfilled.

### EKSEMPLER PÅ IKKE-UNIKE BVP

The study of existence and uniqueness of **boundary value problems** requires more advance theory. We will only consider some examples of non-uniqueness.

- ▶ Initial value problems IVP:  
Existence and uniqueness is controlled by the smoothness of the problem.
- ▶ Boundary value problems (BVP): Existence and uniqueness may be effected by the boundary terms.

MATTE 2?

$y'' = -y$  (deler opp  
i 2. l. orden)

$y(t) = C_1 \sin(t) + C_2 \cos(t)$

C1 og C2 bestemmes av initial løsningen

Rand verdier betingelser:

$$\text{I} \quad Y(0) = 1, \quad Y\left(\frac{\pi}{2}\right) = 0$$

$$\text{II} \quad Y(0) = 0, \quad Y(\pi) = 0$$

$$\text{III} \quad Y(0) = 0, \quad Y(2\pi) = 1$$

Problem 1:

$$\text{I} \quad Y(0) = C_1 \sin 0 + C_2 \cos 0 = 1 \quad \text{III} \quad Y(0) =$$
$$Y\left(\frac{\pi}{2}\right) = C_1 \sin \frac{\pi}{2} + C_2 \cos \frac{\pi}{2} = 0 \quad C_1 = 0$$

Løsning

$$\boxed{\text{I} \quad Y(t) = \cos(t)}$$

II

$$\begin{aligned} C_1 \sin 0 + C_2 \cos 0 &= 0 \\ C_2 &= 0 \\ C_1 \sin \pi + C_2 \cos \pi &= 0 \\ C_1 \sin \pi &= 0 \end{aligned}$$

Vi kan ikke bestemme  $C_1$   
 $y(t) = C_1 \sin(t)$

$C_1$  kan ikke bestemmes

III

$$\begin{aligned} y(0) &= C_1 \sin 0 + C_2 \cos 0 = 0 \\ C_2 &= 0 \\ y(2\pi) &= C_1 \sin(2\pi) + C_2 \cos(2\pi) = 1 \\ C_2 \cos(2\pi) &= 1 \\ C_2 &= 1 \end{aligned}$$

$C_2$  - kan ikke være 1 og 0 samtidig

-> INGEN LØSNING

## Løse ODE

Let's say you have figured out that your IVP has a unique solution and you want to find it. There are two options:

- ▶ Finding an analytical solution by solving the ODE.
- ▶ Finding an approximate solution numerically.

### Finne analytiske løsninger av ODE

Finding analytical solutions of ODEs only works for special cases. The most common are:

- ▶ Separable 1st order.
- ▶ (System of) linear ODEs.

Finding such analytical solutions is not focus of this course. Most interesting problems are non-linear and non-separable without known analytical solutions. In this case we rely on numerical methods.

### Finne tilnærmede løsninger av ODE

Will instead focus on numerical methods for finding approximate solutions of ODEs.

- ▶ Which method(s) suitable for the ODE at hand?
- ▶ How to implement self or use existing numerical methods in python?
- ▶ What is the expected error in using a numerical method?
- ▶ How to control the error?

In the following we use

$$u' = f(t, u), \quad u(t_0) = u_0$$

as an example problem. Here,  $f$  can be a scalar or vectorial function, linear or nonlinear.

Numerical methods are based on estimating  $u$  in discrete points  $t$ . We will call these points  $t^n$  and the corresponding estimates  $u^n$ . That is,

$$u^n \approx u(t^n), \quad n = 0, 1, 2, \dots, N$$

Since we have an initial value, a natural choice is

$$t^0 = t_0, \quad u^0 = u_0.$$

All methods presented in the following will work (more or less) straightforwardly to both single ODEs and system of ODEs.

## Eulers eksplisitte metode

The simplest numerical method for solving ODEs is Euler's method, which is also called Euler forward or Euler explicit.

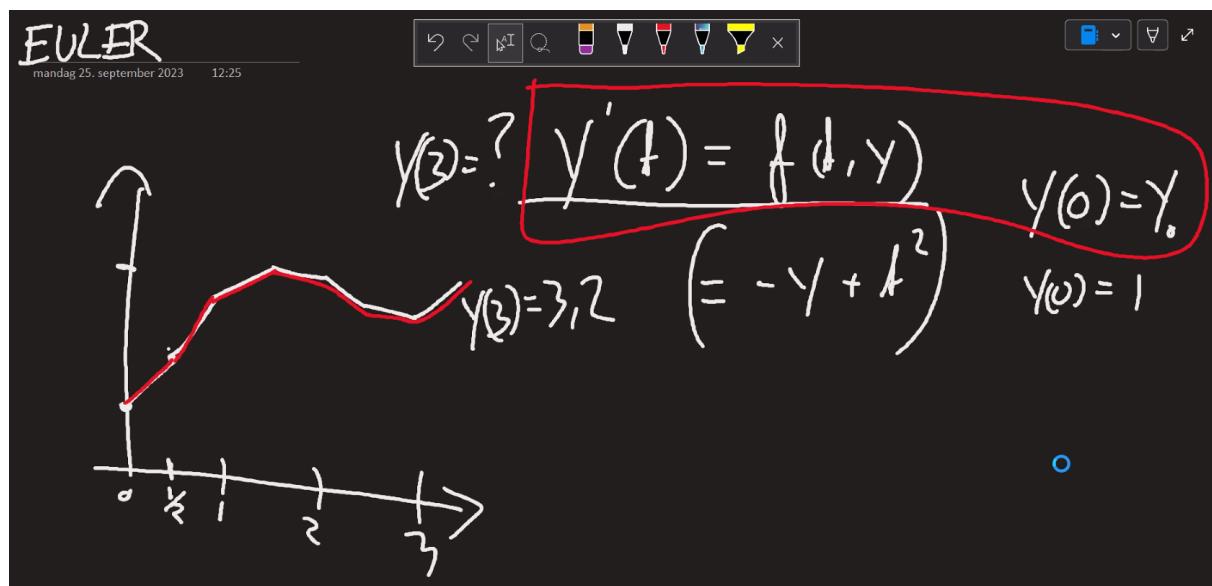
This method is based upon linearizing and calculating successive steps:

$$L(t) = u(t_0) + u'(t_0)(t - t_0) \implies u^1 = L(t^1) = u^0 + f(t^0, u^0)(t^1 - t^0)$$

Using a fixed time-step size  $h = t^1 - t^0$  for all later steps we get

$$u^{n+1} = u^n + f(t^n, u^n)h$$

with  $u^0 = u_0$ .



## Euler Baklengs

A modification of the Euler forward/explicit is to change where  $f$  is evaluated. Instead of using  $f(t^n, u^n)$  one uses instead  $f(t^{n+1}, u^{n+1})$ . This gives us the Euler backward/implicit method, where each  $u^{n+1}$  is given by

$$u^{n+1} = u^n + f(t^{n+1}, u^{n+1})h$$

with  $u^0 = u_0$ .

I

However, this requires to solve a nonlinear equation to actually find  $u^{n+1}$  when  $f$  is nonlinear. In the case that  $f$  is linear in  $u$ , one can easily rewrite the above equation to solve for  $u^{n+1}$ .

## En stiv ODE

### Example 3.27: A stiff ODE

Consider the IVP

$$y'(t) = -15y, \quad y(0) = 1$$

The exact solution is  $y(t) = e^{-15t}$ . Will apply both Euler forward and backward to the ODE to compare.

Euler forward:

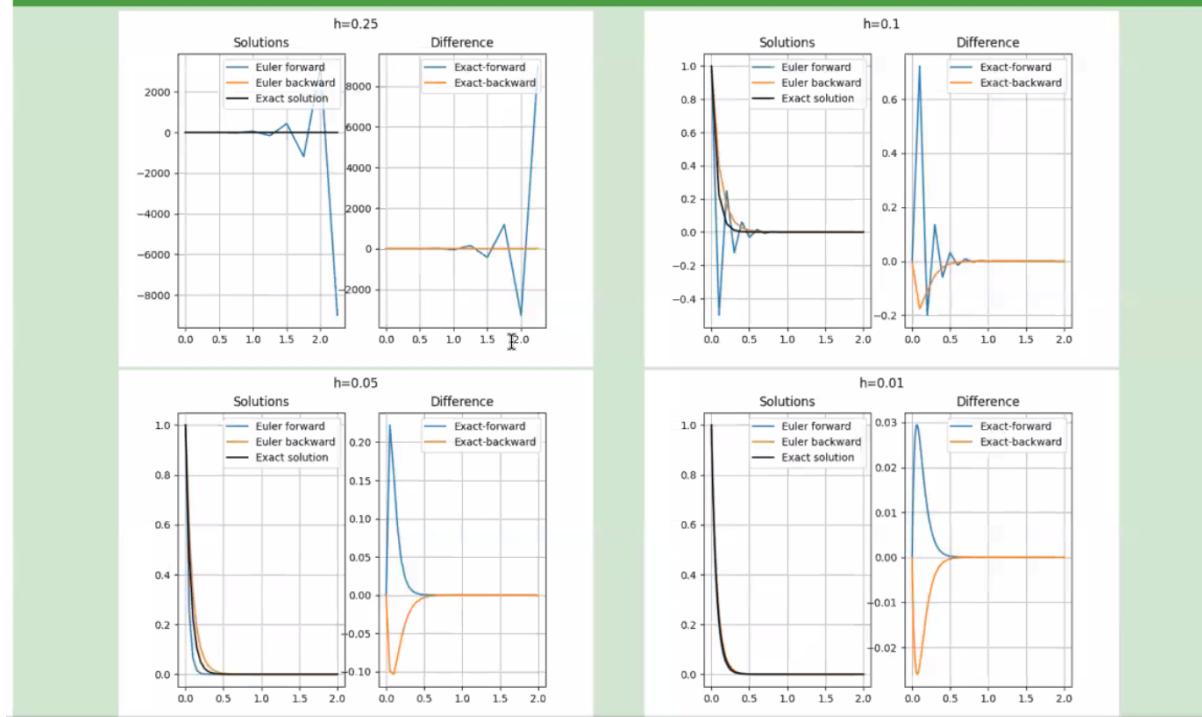
$$y^{n+1} = y^n - 15y^n h, \quad y^0 = 1$$

Euler backward:

$$\begin{aligned} y^{n+1} &= y^n - 15y^{n+1} h, \quad y^0 = 1 \\ &\Downarrow \\ y^{n+1} &= \frac{1}{1 + 15h} y^n, \quad y^0 = 1 \end{aligned}$$

Kode i python

## Example 3.27: A stiff ODE - continued



## Explicit and implicit methods



From the example we can note the following:

- ▶ Although Euler forward worked eventually, a very small time-step size was needed to get reasonable results.
- ▶ Such behavior is typical when using an explicit method to solve a *stiff* problem.
- ▶ Euler backward behaved much better at same time-step size and gave a smaller error.

A *stiff* problem is simply an ODE where explicit numerical methods are unstable unless using a very small time-step size. Typical when solution shows different types of behavior.

For stiff problems, implicit methods are generally more stable and provides better approximations - although potentially too smooth.



As general advice: Try first an explicit method. If that fails or you suspect your problem is stiff, go for an implicit method. Each step is more expensive for an implicit methods, but implicit methods can be overall cheaper for stiff problems since longer time-step sizes can be used.

## Implicit methods and nonlinear problems

In Example 3.27, a linear ODE was considered, and  $y^{n+1}$  could still be easily found. In general, for a nonlinear  $f$ , we need to solve the nonlinear equation

$$u^{n+1} = u^n + f(t^{n+1}, u^{n+1})h$$

for  $u^{n+1}$ . If rephrasing to

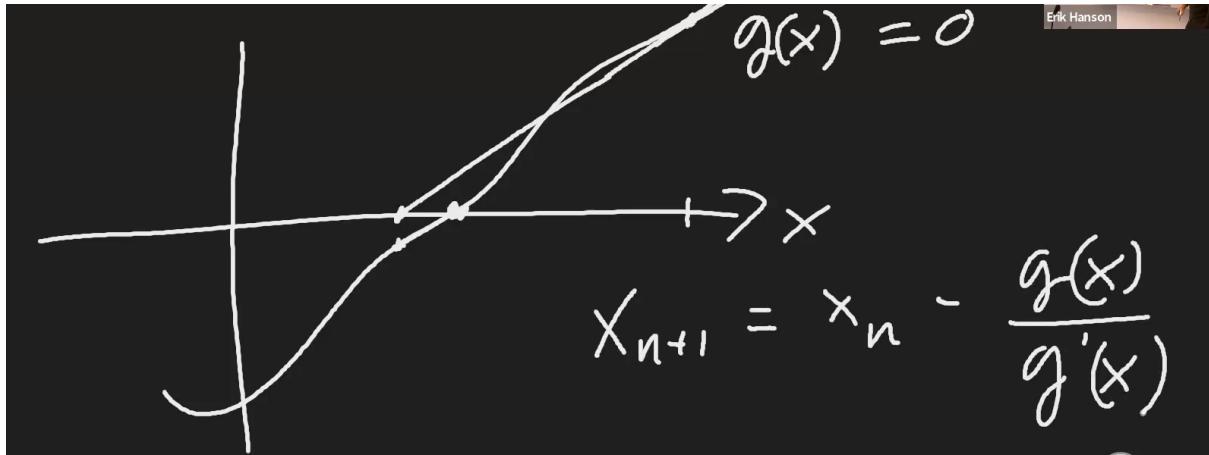
$$u^{n+1} - u^n - f(t^{n+1}, u^{n+1})h = 0 \Leftrightarrow g(u^{n+1}) = 0,$$

we see this can be seen as finding a root of a nonlinear equation. This can be done for example by Newton's method:

$$u^{n+1,j+1} = u^{n+1,j} - (g'(u^{n+1,j}))^{-1}g(u^{n+1,j})$$

Here,  $j$  is the Newton iteration index. Two things are needed:

- ▶ An initial guess  $u^{n+1,0}$  to start the iterations from. Often the previous time step is used:  $u^{n+1,0} = u^n$ .
- ▶ The derivative  $g'$ . In case  $f$  is vectorial (i.e. system of ODEs), this is a Jacobian matrix.



## Implicit methods and nonlinear problems

### Example 3.28: Skydiver IVP with backward Euler

We here set out to solve the IVP from Example 3.1 and 3.17 with Euler backward and Newton's method:

$$\frac{dv}{dt} = \underbrace{g - \frac{\gamma}{m}v^2}_{f(t,v)}, \quad v(0) = 0.$$

The steps in Euler backward are

$$v^{n+1} = v^n + hf(t^{n+1}, y^{n+1}), \quad v^0 = 0.$$

To use Newton's method to solve the nonlinear equation for  $v^{n+1}$ , formulate

$$F(u) = u - v^n - hf(t^{n+1}, u).$$

In python, this can be done by calling `newton_krylov` in each time step.

- ▶ `newton_krylov` can be used to solve system of nonlinear equations as well.
- ▶ `newton_krylov` uses krylov approximation to estimate the Jacobian matrix when it is not given as input to the method.
- ▶ Other options for solving nonlinear (system of) equations in python are `root` and `fsolve`. All are part of `scipy`.

## Implicit methods and nonlinear problems in Python



A python script for the time stepping using Euler backward and Newton's method could contain a segment like seen below, where a nonlinear function is defined at every step and solved using `newton_krylov`.

```

1 from scipy.optimize import newton_krylov
2
3 ## ... ##
4
5 while t[-1] <= t_max:
6     def F(u):
7         return u - y[-1] - h * f(t[-1] + h, u)
8     y_new = newton_krylov(F, y[-1])
9     t_new = t[-1] + h
10    t.append(t_new)
11    y.append(y_new)
12 return np.array(t), np.array(y)

```

## Local and global error



When using a numerical method to find an approximate solution to an IVP, we make errors. We can separate between local and global discretization errors:

- ▶ The *local discretization error* is the error obtained at every time step due to approximating  $u'$  instead of using the exact  $u'$  to find the value at the next time step:

$$u(t^{n+1}) - u^{n+1}$$

where  $u(t^{n+1})$  represents the exact solution of the IVP.

- ▶ The *global discretization error* is the error accumulated over the course of the whole computation:

$$\max_n |u(t^n) - u^n|$$

A calculation of the local error for time step  $t^{n+1}$  assumes one has the correct solution at time step  $t^n$  available. However, since one makes a local error at every time step, the local errors add up and the global error is therefore larger.



## Convergence order

The local discretization error of Euler forward can be found as following:  
One step of forward Euler uses

$$u^{n+1} = u^n + h f(t^n, u^n).$$

The exact solution would however calculate, using Taylor expansions

$$u(t^{n+1}) = u(t^n + h) = u(t^n) + h \underbrace{u'(t^n)}_{f(t^n, u^n)} + \frac{1}{2} h^2 u''(t^n) + O(h^3).$$

The local discretization error is the difference between these, assuming  $u^n = u(t^n)$ :

$$u(t^{n+1}) - u^{n+1} = \frac{1}{2} h^2 u''(t^n) + O(h^3) = O(h^2)$$

Since the local error is  $O(h^2)$ , the global error (which will add up over  $1/h$  timesteps) is  $O(h)$ . Hence, the Euler forward (and backward) is known as a *first-order method*.

## Convergence order

Since the Euler forward and backward methods are first-order methods having global error behaving as  $O(h)$ , we expect the following:

When halving the time-step size of a first-order method,  
we expect the global error to be halved as well.

Calculating the global error for a range of time-step sizes in Example 3.27 we get the following:

$h$	Euler forward		Euler backward	
	global error	factor from previous	global error	factor from previous
0.1	0.7231	-	0.1769	-
0.1/2	0.2224	3.25	0.1034	1.70
0.1/2 <sup>2</sup>	0.0817	2.72	0.0600	1.72
0.1/2 <sup>3</sup>	0.0375	2.18	0.0320	1.88
0.1/2 <sup>4</sup>	0.0179	2.09	0.0166	1.92
0.1/2 <sup>5</sup>	0.0088	2.04	0.0085	1.96

Both approach an improvement factor of 2, meaning that the global error is halved as the time-step size is halved.

# Numerisk løsning av ODE

## ORDEN:

Euler (implisitt) orden 1:

Hvis  $h$  reduseres, så reduseres  
feilen (ca) like mye

$$h_1 = 0,2$$



$$e_1 \approx 0,4$$

$$h_2 = \frac{1}{2} h_1 = 0,1$$

$$e_2 \approx \frac{1}{2} e_1 \approx 0,2$$

## Høyere orden metoder

Higher order methods can be achieved when improving how the derivative of  $u$  is approximated.

### Definition 3.29: Higher order methods

A discretization method of an IVP is said to be an  $p$ -th order method when the global error behaves as  $O(h^p)$  and the local error as  $O(h^{p+1})$ , that is, when

$$\max_n |u(t^n) - u^n| = O(h^p)$$
$$u(t^{n+1}) - u^{n+1} = O(h^{p+1})$$

for a fixed time-step size  $h$ .

When halving the time-step size of a second-order method, we expect the global error to be reduced with a factor of 4. When halving the time-step size of a  $p$ -th order method, we expect the global error to be reduced by a factor  $2^p$ .

We will here consider two ways of achieving higher order methods: multi-stage and multi-step methods.

### Definition 3.30: Multi-stage and multi-step methods

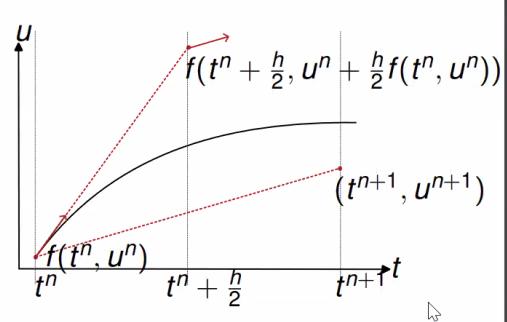
- ▶ A multi-stage method uses stages between  $u^n$  and  $u^{n+1}$  to approximate the slope of  $u$  between these two points.
- ▶ A multi-step method uses information from earlier time steps ( $u^{n-1}, u^{n-2}, \dots$ ) to approximate the slope of  $u$  between  $u^n$  and  $u^{n+1}$ .

Multi-stage and multi-step methods can be explicit or implicit, depending on whether only known information is needed or if  $u^{n+1}$  itself is needed to calculate  $u^{n+1}$ . This does not influence their order, but whether they are applicable to stiff problems or not.

The most common multi-stage methods are *Runge-Kutta* methods.

- ▶ By using  $p$  stages (including the previous time step), an  $p$ -th order method can be found. Several possibilities to do this.
- ▶ The two-stage method also known as the midpoint rule is a second-order method:

$$u^{n+1} = u^n + hf(t^n + \frac{1}{2}h, u^n + \frac{h}{2}f(t^n, u^n))$$



- ▶ Many different versions exist! We mention in particular a four-stage version which is a fourth-order method:

$$u^{n+1} = u^n + \frac{h}{6}(T_1 + 2T_2 + 2T_3 + T_4) \quad \text{where}$$

$$T_1 = f(t^n, u^n), T_2 = f(t^n + \frac{h}{2}, u^n + h\frac{T_1}{2}), T_3 = f(t^n + \frac{h}{2}, u^n + h\frac{T_2}{2}), T_4 = f(t^n + h, u^n + hT_3)$$

- ▶ There are also implicit Runge-Kutta methods, e.g. an implicit midpoint rule:

$$u^{n+1} = u^n + hf(t^n + \frac{h}{2}, \frac{1}{2}(u^n + u^{n+1}))$$



The most common explicit multi-step methods are the *Adams-Basforth* methods.

- ▶ By using  $p$  earlier time steps, a  $p$ -th order method can be found by combining the earlier steps linearly. Only one unique way to do this for explicit methods.
- ▶ Two-step second-order method:

$$u^{n+1} = u^n + \frac{3}{2}hf(t^n, u^n) - \frac{1}{2}hf(t^{n-1}, u^{n-1}).$$

- ▶ Higher order methods also exist.

The most common implicit multi-step methods are the *backwards differentiation formulas (BDFs)*.

- ▶ Two-step second order method (BDF2):

$$u^{n+1} - \frac{4}{3}u^n + \frac{1}{3}u^{n-1} = \frac{2}{3}hf(t^{n+1}, u^{n+1})$$

- ▶ Higher order methods also exist.

Note that a higher order multi-step method cannot be used for the first time step(s). Then a multi-stage or a first order method (Euler forward/backward) must be used.

## Stabilitet, konsistens og konvergens

When choosing which discretization method to use for your ODE, you should take into account the stability, consistency and convergence properties of the discretization method.

### Definition 3.31: Stability

A discretization method is said to be (numerically) stable if numerical errors and round-off errors are damped and not allowed to grow unbounded.  $\square$

### Definition 3.32: Consistency

A discretization method is said to be consistent if the local discretization error approaches zero as the time-step size  $h$  approaches zero.  $\square$

### Definition 3.33: Convergence

A discretization method is said to be convergent if the global discretization error approaches zero as the time-step size  $h$  approaches zero.

Checking convergence of the discretization method is the verification step mentioned in Part 1!

- ▶ The consistency is checked by looking at one time step (like for Euler forward on slide 50).
- ▶ Convergence is what we want - but is less easy to obtain!
- ▶ As a rule of thumb:

$$\text{stability} + \text{consistency} \implies \text{convergence}$$

$\square$

- ▶ In Example 3.27, we were outside the stability region of Euler forward for  $h = 0.25$ . We had a consistent method, but without stability we cannot have convergence.
- ▶ Investigating stability of time stepping methods is not part of the curriculum, but be aware that:
  - ▶ Most well-used discretization methods have known stability regions.
  - ▶ Explicit methods have more limited stability regions, and are typically only stable for small time-step sizes.
  - ▶ Higher-order methods have smaller stability regions than lower-order methods.

## Adaptivity

Discretization methods for ODEs can be adaptive in order to control their (estimated) local error at every time step. The local error can be estimated in two ways:

- ▶ Comparing the difference when using a time step  $h$  and  $h/2$ .
- ▶ Comparing the difference when using a lower-order method and higher-order method for same time-step size.

If the estimated local error is *low enough*, the current time-step size is kept or even increased. If the error is too large, the time-step size is reduced.

## ODE i Python



We will focus on two methods available in `scipy`:

- ▶ `odeint`
- ▶ `solve_ivp`

Both can solve systems of nonlinear first order ODEs using adaptivity, but differ in which methods they apply and how the input and output is.

Both methods rely on controlling the local error through tolerances `atol` (absolute tolerance) and `rtol` (relative tolerance) such that

$$\ell_i^n \leq \text{rtol}_i \cdot |u_i^n| + \text{atol}_i$$

Here,  $\ell_i^n$  is the estimated local error at time step  $n$  for component  $i$  (in case of system of ODEs), while  $u_i^n$  is the  $i$ -th component of the solution.

`odeint` uses an old FORTRAN implementation called “lsoda”. This is a much used implementation, also covered by the book of Velten for R.

- ▶ lsoda has stiffness detection and can automatically switch between explicit and implicit methods at every time step.
- ▶ Uses Adams-Bashforth (order 1-12) in case of nonstiff and BDF (order 1-5) together with Newton iterations in case of stiff.
- ▶ Step size and order of method chosen adaptively to ensure stability and low enough local error.
- ▶ If BDF is used, the method estimates the needed Jacobian matrix itself if not provided by the user.
- ▶ Input of `odeint`:
  - ▶ Right-hand side  $f(u, t)$  (Note order of  $u$  and  $t$ ).
  - ▶ Initial condition.
  - ▶ Vector of times where solution is requested.
  - ▶ All other inputs are optional. Includes Jacobian, `atol`, `rtol`, maximum step size, to mention some.
- ▶ Output of `odeint`:
  - ▶ Solution.
  - ▶ Can give additional information if requested, e.g. how many Newton iterations and which method were used at every time step, etc.
- ▶ For documentation, read <https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html> or in python, type `help(odeint)`.

`solve_ivp` lets the user specify which method is to be applied. Options are

- ▶ Explicit Runge-Kutta methods of two different orders to control the error.  
Default is using a 5th order method for time stepping and a 4th order method to estimate error.
  - ▶ Implicit Runge-Kutta method with error control.
  - ▶ BDFs up to order 5.
  - ▶ Isoda as described for `odeint`.
- 
- ▶ Input of `solve_ivp`:
    - ▶ Right-hand side  $f(t, u)$  (Note order of  $u$  and  $t$ ).
    - ▶ Start and end time.
    - ▶ Initial condition.
    - ▶ All other inputs are optional. Includes which method to use, additional times where solution is sought, Jacobian, `atol`, `rtol`, maximum step size, to mention some.
  - ▶ Output of `solve_ivp`:
    - ▶ Time points and solution.
    - ▶ Additional information, e.g. how many Newton iterations were used, termination reasons, etc.
  - ▶ For documentation, read [https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve\\_ivp.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html) or in python, `help(solve_ivp)`.

## Python - parameter estimation

- ▶ An efficient ODE-solver can be combined with a numerical minimization to do parameter estimation
- ▶ ODE

$$y'(t) = f(t, y, p), \quad y(0) = y_0,$$

where  $p$  are some parameters in the ODE.

- ▶ Both  $y$  and  $p$  may be vectors
- ▶ The model is now a solution of the ODE with a given set of  $p$ -values:

$$M(t, p) = y(t, p)$$

- ▶ Cost function

$$C(p) = \|M(t_i, p) - y_i\|,$$

where  $y_i$  are the observations.

NOTE: The model contains a numerical solution of the ODE

### Wine fermentation (from Velten)

Basic concept: Yeast eat sugar and produce alcohol

**S:** Closed fermentation container

**Q** How can the temperature and nutritious be altered to make a fast and complete fermentation?

We will set up a model describing the relationship between the amount of yeast cells, the amount of sugar, the amount of nitrogen and the amount of alcohol (Ethanol).



## Gjær

$$\frac{dX}{dt} = \mu \cdot X - k_d \cdot X$$

Growth of yeast cells:

$$\mu = \mu_{\max} \cdot \frac{N}{K_N + N}$$

Growth rate is limited by the nitrogen:

The maximum growth rate is dependent on temperature:

$$\mu_{\max}(T) = 0.18 \cdot \exp\left(14200 \cdot \frac{T - 300}{300RT}\right) - 0.0054 \cdot \exp\left(121000 \cdot \frac{T - 300}{300RT}\right)$$

Death rate depends on the alcohol content:  $k_d = k \cdot E$

## Alkohol

The Alcohol production depends on the amount of yeast cells:

$$\frac{dE}{dt} = \beta X$$

The alcohol production rate depends on the availability of sugar:

$$\beta = \beta_{\max} \cdot \frac{S}{K_S + S}$$

where temperature also have a role:

$$\beta_{\max}(T - 273.15) = \beta_{\max,24^{\circ}\text{C}} \cdot (0.00132 \cdot T^2 + 0.00987 \cdot T - 0.00781)$$

## Nitrogen og sukker

The change in the sugar and nitrogen is related to the yeast growth:

$$\frac{dS}{dt} = -\frac{\beta}{Y_{ES}} X$$

$$\frac{dN}{dt} = -\frac{\mu}{Y_{XN}} \cdot X$$

In this setup, it is possible to add nitrogen during the process:

$$\frac{dN}{dt} = r(t) - \frac{\mu}{Y_{XN}} \cdot X$$

# Wine fermentation model

$$\frac{dX}{dt} = \mu(T, N) \cdot X - k \cdot E \cdot X$$

$$\frac{dN}{dt} = r(t) - \frac{1}{Y_{XN}} \cdot \mu(T, N) \cdot X$$

$$\frac{dE}{dt} = \beta(T, S) \cdot X$$

$$\frac{dS}{dt} = -\frac{1}{Y_{ES}} \beta(T, S) \cdot X$$

Parameter	Description	Unit	Value	Source
$\mu$	Specific growth rate	$\text{h}^{-1}$	(3.284)	[121]
$\mu_{\max}$	Maximum specific growth rate	$\text{h}^{-1}$	(3.286)	[122, 124, 125]
$T$	Temperature	K	Data	[122]
$K_N$	Monod constant for nitrogen	g nitrogen/l	0.01	[121]
$k_d$	Death constant	$\text{h}^{-1}$	(3.287)	[121]
$k$	Specific death constant	l/g ethanol//h	Unknown	
$Y_{XN}$	Stoichiometric yield coefficient of biomass on nitrogen	g biomass/g nitrogen	18	[127]
$\beta$	Specific ethanol production rate	g ethanol/g biomass/h	(3.289)	[121]
$\beta_{\max}$	Maximum specific ethanol production rate	g ethanol/g biomass/h	(3.290)	[121,126]
$\beta_{\max, 24^\circ\text{C}}$	Maximum specific ethanol production rate at $24^\circ\text{C}$	g ethanol/g biomass/h	0.3	[121]
$K_S$	Michaelis–Menten-type constant for sugar	g sugar/l	10	[121]
$Y_{ES}$	Stoichiometric yield coefficient of ethanol on sugar	g ethanol/g sugar	0.47	[121]
$X_0$	Initial yeast concentration at $t = 0$	g biomass/l	0.2	fermentation.csv
$E_0$	Initial ethanol	g ethanol/l	0	fermentation.csv

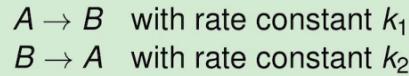
PYTHON - KODE

## Phase plan plots

Especially for systems of two or three coupled ODEs, it can be useful to investigate the *phase plane* of the system. We show how via an example.

### Example 3.34: Chemical system

We consider a reversible chemical reaction involving two species  $A$  and  $B$ , say water and ice.  $A$  can become  $B$  and vice versa, where the reaction rates in each direction depend on the concentration of the ingredient scaled with a rate constant. That is

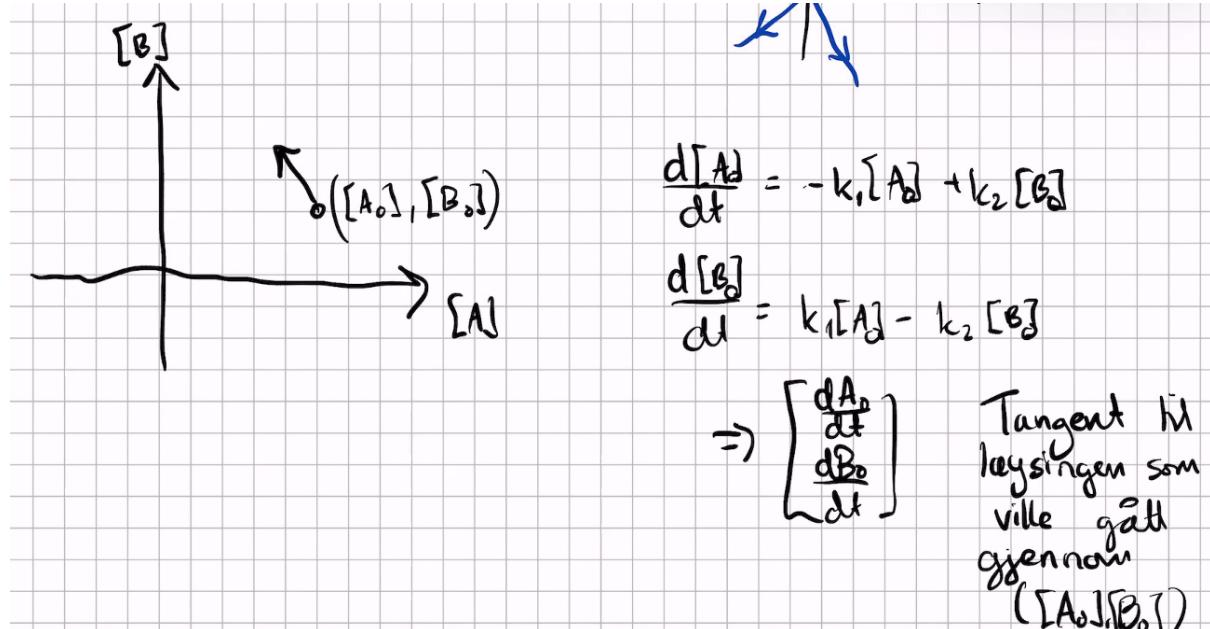


The concentrations of  $A$  and  $B$  can hence be described with two coupled ODEs:

$$\begin{aligned} \frac{d[A]}{dt} &= -k_1[A] + k_2[B] \\ \frac{d[B]}{dt} &= k_1[A] - k_2[B] \end{aligned}$$

Without solving, we can investigate how the solutions would behave by using the phase plane.

EKS:



### Example 3.34: Chemical system - continued

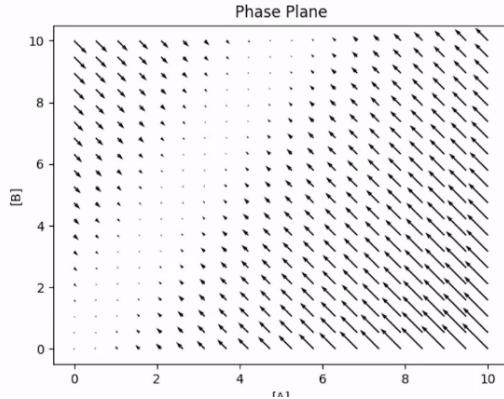
In the phase plane, the variables  $[A]$  and  $[B]$  are on the axes. Time only present through directions.

If have a solution with values  $([A_0], [B_0])$ , this solution would then change with time through

$$\begin{aligned}\frac{d[A]}{dt} \Big|_{([A_0], [B_0])} &= -k_1[A_0] + k_2[B_0] \\ \frac{d[B]}{dt} \Big|_{([A_0], [B_0])} &= k_1[A_0] - k_2[B_0]\end{aligned}$$

The values of the time derivatives hence give us a direction in the phase plane. Can indicate how the solution would have acted. By choosing these directions for many points, get an idea of behavior of solution.

For  $k_1 = 1$  and  $k_2 = 0.5$ , get the following phase plane



### Example 3.34: Chemical system - continued

What does phase plane tell us? Arrows seem to go toward and then stop at line  $[B] = 2[A]$ . From equations, this makes sense as

$$\begin{aligned}\frac{d[A]}{dt} &= -1[A] + 0.5[B] = 0 \Rightarrow [A] = 0.5[B] \\ \frac{d[B]}{dt} &= 1[A] - 0.5[B] = 0 \Rightarrow [A] = 0.5[B]\end{aligned}$$

Both concentrations have zero derivatives here; which corresponds to solutions that are constant in time. Physically, all solutions fulfilling  $[B] = 2[A] \Leftrightarrow [A] = 0.5[B]$  are in chemical equilibrium.

- ▶ Phase plane plots can be made without solving the ODEs.
- ▶ Give a visual understanding of how system behaves.
- ▶ Especially suitable for system of two ODEs, can work with system of three ODEs as well.
- ▶ Especially useful to combine with stationary points. Can decide whether a stationary point is *stable*.

## Stasjonært stabile punkt

### Definition 3.35: Stable stationary point

A stationary point in the phase plane is stable if small perturbations of the corresponding solution does not significantly change the solution.

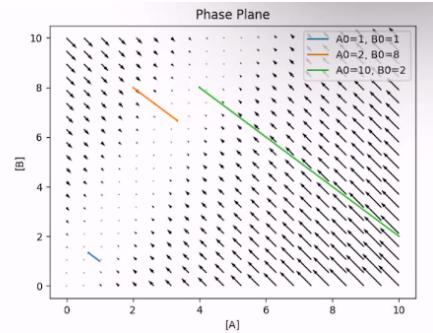
Rule of thumb: If arrows in the phase plane point towards the point, it is stable. If arrows point away, it is unstable. If there is a combination of arrows pointing towards and away from the point, it can be called semistable.

For Example 3.34, the entire line  $[A] = 0.5[B]$  consists of stable points since all arrows point towards it.

### Example 3.36: Pendulum

Which two stationary points does a pendulum have? Are they stable or unstable?

To make phase plane plots even more visual, we can combine with some solution curves too. They can be found by choosing some initial conditions, solving with e.g. `odeint` or `solve_ivp`, and including solution(s) in phase plane plot.



General advice:

- ▶ It can be tricky to find a good range for the solution variables in the phase plane. But if you know the stationary points, probably a good idea to include them in the range.
- ▶ Do not include too many points in each direction. More is not necessarily better.
- ▶ Same with including solution curves: More solution curves is not necessarily better. Rather try to choose 4-5 initial conditions wisely to get a good visualization of behavior.

## Avdimensjonalisering

Sometimes our model equations contain very different size of numbers. E.g. if measuring something small in  $\mu\text{m} = 10^{-6} \text{ m}$ , which takes one year =  $3.16 \times 10^7 \text{ s}$ .

Dealing with numbers of very different sizes can be problematic when using numerical methods on a computer.

- ▶ Your resulting equations can be *ill-conditioned*. Round-off errors more prominent.
- ▶ Numerical errors can be issue if tolerances chosen without being careful.

To avoid such issues, we can non-dimensionalize the equations.

When non-dimensionalizing our equations, we bring all variables into a form without physical dimensions. This is done by scaling: time variable  $t$  is scaled with a reference time, spatial variable  $x$  is scaled with a reference length, etc.

New variables are therefore dimensionless:

$$\hat{t} = \frac{t}{t_{\text{ref}}}, \quad \hat{x} = \frac{x}{x_{\text{ref}}}$$

This is inserted into the model equations, and we obtain new model equations using the dimensionless variables, where the reference values appear. We then choose values for the reference values:

- ▶ Can choose according to what we know/expect about the size of our variables. E.g. size of domain and initial condition, length of time interval of interest.
- ▶ If no such information: Can choose them such that new equations have (non-dimensional) parameters that are roughly 1.

### Example 3.37: Skydiver - again

We return to our skydiver:

$$\frac{dv}{dt} = g - \frac{\gamma}{m} v^2, \quad v(0) = 0.$$

Two variables,  $v$  and  $t$ . We non-dimensionalize them by

$$\hat{v} = \frac{v}{v_{\text{ref}}}, \quad \hat{t} = \frac{t}{t_{\text{ref}}}.$$

Non-dimensional equations are

$$\frac{d\hat{v}}{d\hat{t}} = \frac{gt_{\text{ref}}}{v_{\text{ref}}} - \frac{\gamma v_{\text{ref}} t_{\text{ref}}}{m} \hat{v}^2.$$

Have a typical/expected velocity in the form of terminal velocity:  $v_{\infty} = \sqrt{\frac{mg}{\gamma}}$ .

Choose as  $v_{\text{ref}}$ .

$$\frac{d\hat{v}}{d\hat{t}} = t_{\text{ref}} \sqrt{\frac{g\gamma}{m}} - t_{\text{ref}} \sqrt{\frac{g\gamma}{m}} \hat{v}^2$$

No typical time. Can instead choose  $t_{\text{ref}} = \sqrt{\frac{m}{g\gamma}}$  so that equation has parameters that are roughly 1. Then:

$$\frac{d\hat{v}}{d\hat{t}} = 1 - \hat{v}^2, \quad \hat{v}(0) = 0.$$

Benefits of non-dimensionalizing mathematical models:

- ▶ Can make equations easier to deal with: peel away “noise”.
- ▶ Can make applying numerical methods much easier as less likely to run into problems due to numerical or round-off errors.
- ▶ Easier to make plots when all variables have values around 1.
- ▶ The non-dimensional model can include typical non-dimensional parameters that can be used to classify system. E.g. for fluid flow, large Reynolds number  $Re = \frac{\rho_{\text{ref}} v_{\text{ref}} L_{\text{ref}}}{\mu_{\text{ref}}}$  means one will have turbulence.

But: to give physical interpretation to solution, remember to re-dimensionalize your numbers afterwards.

# ODE i hav siruklasjon

# Partielle differensial ligninger (PDE)

## Motivasjon

In this part we will go through the role of partial differential equations (PDEs) in mathematical modeling, with a special focus on how to solve PDEs numerically.

Questions that we will address are

- ▶ When do we need a model with PDEs and how do we create a PDE model?
- ▶ How to classify common types of PDEs?
- ▶ When is a PDE model well-posed?
- ▶ How to solve PDEs numerically using finite differences and finite element methods?
- ▶ How to interpret the solution of PDEs?
- ▶ How to use PDEs in modeling and simulating physical phenomena?

This part is based on Chapter 4 of the book of Kai Velten and Section 2.4.6, 14.1 and 14.2 of the book of Hans-Joachim Bungartz et al.

## Rollen til PDE

- ▶ In the previous part we worked with ODEs: solution function of one variable, e.g. time.
- ▶ Some phenomena also vary with space, for example temperature distribution in engine, solute concentration in water tank or bending of a beam under stress.
- ▶ Need to describe rate of change with space (and potentially also time); partial derivatives. Hence need distributed instead of lumped models (cf. Definition 1.17)
- ▶ PDEs come in many different shapes. Will only cover a few though common here.
- ▶ Will especially focus on the heat equation, which describes heat conduction but also diffusion of a solute concentration.

## Sette opp PDE

Will look into how to set up a PDE model based on the heat equation:

$$\frac{\partial T}{\partial t} = \alpha \Delta T$$

where  $\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$  is the Laplace operator,  $T(t, \mathbf{x})$  is temperature and  $\alpha$  is a proportionality constant. Describes temperature distribution and changes through a material.

- ▶  $T$  has dimension K (Kelvin) and will vary with time and space.
- ▶  $\alpha = \frac{k}{c\rho}$  is called the thermal diffusivity of the material.
  - ▶ Thermal conductivity  $k$  has dimension  $\text{W K}^{-1} \text{m}^{-1}$  and describes the material's ability to conduct heat.
  - ▶ Specific heat capacity  $c$  has dimension  $\text{J kg}^{-1} \text{K}^{-1}$  and describes the material's ability to store heat (per kg).
  - ▶ Density  $\rho$  has dimension  $\text{kg m}^{-3}$ .

The heat equation arises from Fourier's law together with conservation of energy.

Conservation of energy is a physical principle and says that the total energy of an isolated system remains constant.

This means that the rate of change of energy inside a domain can only be due to energy fluxes across the boundary of that domain.

### Remark 4.1: Conservation principles

Many PDEs arise from conservation principles: conservation of energy, conservation of mass and conservation of momentum give rise to the most commonly used PDEs for fluid flow, solute transport and temperature variability.

If we consider a volume  $V$ , the (heat) energy  $Q$  inside that volume is expressed by

$$Q = \int_V c\rho T dV$$

Energy has dimension  $J$ , while specific heat capacity  $c$  and density  $\rho$  quantifies how much energy a certain material of a certain temperature can store in a point. Hence, multiplying with the temperature (in  $K$ ), and integrating over the points in the current volume gives the amount of heat stored in the volume. Heat capacity and density of material can be measured, and we consider them here material constants.

The rate of change of energy inside the volume is hence

$$\frac{dQ}{dt} = \frac{d}{dt} \int_V c\rho T d\mathbf{x} = \int_V c\rho \frac{\partial T}{\partial t} dV$$

Fourier's law is an empirical law relating heat flux (also called heat flow rate) with temperature gradient:

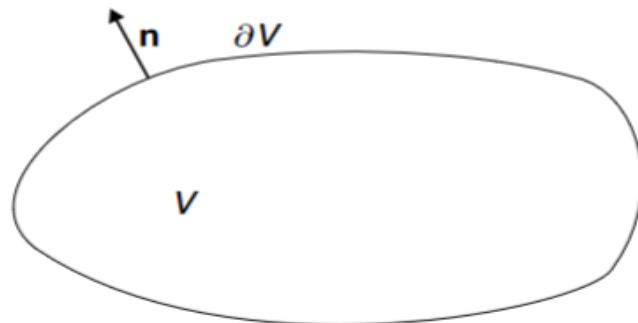
$$\mathbf{q} = -k \nabla T$$

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = -k \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{bmatrix}$$

Heat flux  $\mathbf{q}$  has dimensions  $\text{W m}^{-2}$ , hence energy per second per surface area. It represents heat going through an imagined surface. Fourier's law says that heat goes in opposite direction of temperature gradient; from high to low temperatures. And  $k$  is the proportionality constant in this process. Values of  $k$  can be measured and depends on type of material. A high value of  $k$  means that the material is good at transmitting heat when there is a temperature gradient. Insulating materials should have a low value of  $k$ .

By conservation of energy, the rate of change of energy inside our reference volume  $V$  is only due to heat flux across boundary of  $V$ :

$$\frac{dQ}{dt} = \underbrace{\int_V c\rho \frac{\partial T}{\partial t} dV}_{\text{Rate of change of energy}} = \underbrace{\int_{\partial V} k \nabla T \cdot \mathbf{n} dS}_{\text{Heat flux into } V}$$



By the Gauss' (divergence) theorem we have that

$$\int_V c\rho \frac{\partial T}{\partial t} dV = \int_V \nabla \cdot (k \nabla T) dV$$

### Theorem 4.2: Gauss' Theorem

For a subset  $V \subset \mathbb{R}^n$ , which is closed and bounded and has a piecewise smooth boundary  $\partial V$ , and if  $\mathbf{F}$  is a continuously differentiable vector field, then

$$\int_V \nabla \cdot \mathbf{F} dV = \int_{\partial V} \mathbf{F} \cdot \mathbf{n} dS$$

where  $\mathbf{n}$  is the unit normal pointing out of the set  $V$ .

We hence have

$$\int_V c\rho \frac{\partial T}{\partial t} dV = \int_V \nabla \cdot (k \nabla T) dV$$

Since this is true for any reference volume, it is also true for a point. Hence

$$\begin{aligned} c\rho \frac{\partial T}{\partial t} &= \nabla \cdot (k \nabla T) \\ \frac{\partial T}{\partial t} &= \frac{k}{c\rho} \nabla^2 T \end{aligned}$$

### Remark 4.3: Derivation of heat equation

The book of Kai Velten gives a slightly different derivation of the heat equation by considering one-dimensional differentials and then extending to several dimensions. The benefit is that Gauss' theorem is then not needed.

Depending on the type of material, different versions of the heat equation exist:

- ▶ For an **inhomogeneous/heterogeneous** material, material properties  $c$ ,  $\rho$  and  $k$  vary with location:

$$\frac{\partial T}{\partial t}(t, \mathbf{x}) = \frac{1}{c(\mathbf{x})\rho(\mathbf{x})} \nabla \cdot (k(\mathbf{x}) \nabla T(t, \mathbf{x}))$$

This is typical if there are impurities in the material.

- ▶ For an **anisotropic** material, the flux is direction dependent. Hence,

$$\mathbf{q} = -\mathbf{K} \nabla T \text{ for a matrix } \mathbf{K} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} :$$

$$\frac{\partial T}{\partial t}(t, \mathbf{x}) = \frac{1}{c\rho} \nabla \cdot (\mathbf{K} \nabla T(t, \mathbf{x}))$$

This is typical if there is e.g. some fiber structure in the material.

## Varme / diffusjon Ligninger

The heat equation also describes changes in a concentration  $C$  of a substance, which can change due to diffusion; e.g. salt dissolved in water. By conservation of mass and Fick's law of diffusion one has

$$\frac{\partial C}{\partial t} = \nabla \cdot (D \nabla C)$$

This is technically the heat equation, but also called the diffusion equation.

- ▶ Concentration  $C(t, \mathbf{x})$  has dimension  $\text{kg m}^{-3}$  (other possibilities too).
- ▶ Diffusivity  $D$  has dimension  $\text{m}^2 \text{s}^{-1}$  and describes the ability of the substance (solute) to move through the background substance (solvent).

## Klassifisere PDE

### Definition 4.6: Linear PDEs

A PDE for  $u(t, \mathbf{x})$  is linear if  $u$  and partial derivatives of  $u$  only appear as linear combinations. Otherwise the PDE is nonlinear.

### Example 4.7: Heat equation is linear

The heat equation

$$\frac{\partial T}{\partial t} = \alpha \Delta T = \alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right)$$

is a linear PDE for  $T(t, \mathbf{x})$ .

When solving PDEs, we will limit our attention to linear PDEs in this course. However, the general idea to handle nonlinear PDEs is to linearize them, using e.g. Newton's method.

$$\frac{\partial u}{\partial t} = k \cdot u \cdot \frac{\partial u}{\partial x} \quad \text{is linear}$$

### Definition 4.8: System of PDEs

When  $m$  functions  $u_1, u_2, \dots, u_m$ , which can depend on  $(t, \mathbf{x})$ , and their partial derivatives up to (and including some of) order  $n$  are coupled through PDEs, we have system of PDEs of order  $n$  and dimension  $m$ .

### Example 4.9: Heat equation in a system of coupled PDEs

Say that the heat conductivity  $k$  varies with location (and time) as it depends on the local concentration of a solute  $C$ . The solute concentration changes with diffusion and its diffusivity  $D$  depends on temperature  $T$ . This can be written

$$\begin{aligned} c\rho \frac{\partial T}{\partial t} &= \nabla \cdot (k(C)\nabla T) \\ \frac{\partial C}{\partial t} &= \nabla \cdot (D(T)\nabla C) \end{aligned}$$

which is a system of PDEs for  $T(t, \mathbf{x}), C(t, \mathbf{x})$ . Additionally the relations  $k(C)$ ,  $D(T)$  are needed. These are generally known as *equations of state* as they describe properties characterizing the system (conductivity, diffusivity) as functions of state parameters (temperature, concentration).

Especially for second-order PDEs, we include a common way to characterize PDEs which influences how to discretize them as they give generally different types of behavior.

In 2D, and omitting time, the general form for a second-order linear PDE is

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + F = 0$$

where  $A, B, C, D, E, F$  are real numbers, but could vary with  $x, y$ . Using the discriminant

$$\mathcal{D} = AC - B^2$$

we can classify this PDE.

#### Definition 4.10: Elliptic, parabolic and hyperbolic PDEs

A second-order linear PDE is called

- ▶ Elliptic if  $\mathcal{D} > 0$
- ▶ Parabolic if  $\mathcal{D} = 0$
- ▶ Hyperbolic if  $\mathcal{D} < 0$

The classification can be generalized to more spatial variables and time, but simple expressions are not easy to obtain.

For the interested reader:

#### Remark 4.11: Elliptic, parabolic and hyperbolic PDEs

For a second-order linear PDE on the form

$$-\sum_{i,j=0}^d p_{ij}u_{x_i x_j} + \sum_{i=0}^d q_i u_{x_i} + ru - f = 0$$

where  $p_{ij}, q_i, r, f$  are real-valued and could depend on  $x_j$ . Assume also that the matrix  $\mathbf{P} = \{p_{ij}\}_{i,j=0}^d$  is symmetric with real eigenvalues  $\lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_d$ . Then the PDE is

- ▶ Elliptic if  $\lambda_i \neq 0 \forall i$  and  $\text{sign}(\lambda_0) = \dots = \text{sign}(\lambda_d)$
- ▶ Parabolic if  $\exists! j : \lambda_j = 0$  and  $\text{rank}(\mathbf{P}, q) = d + 1$ .
- ▶ Hyperbolic if  $\lambda_j \neq 0 \forall i$  and  $\exists! j : \text{sign}(\lambda_j) = \text{sign}(\lambda_k)$  for all  $k \neq j$ .

A second-order linear PDE could also be neither of the above categories.

## NOEN KJENTE PDE - VI HAR 3 STK

We instead focus on three important PDEs:

### Example 4.12: Elliptic PDE

The Poisson equation

$$\Delta u + f(\mathbf{x}) = 0$$

is an elliptic PDE. It can describe the potential of an electric field, elasticity (bending of beam) and steady-state heat conduction.

### Example 4.13: Parabolic PDE

The heat equation

$$\frac{\partial u}{\partial t} = \alpha \Delta u$$

is a parabolic PDE. It can describe heat conduction and solute diffusion. Variants also used in financial mathematics, probability theory and quantum physics.

### Example 4.14: Hyperbolic PDE

The wave equation

$$\frac{\partial^2 u}{\partial t^2} = \alpha \Delta u$$

is a hyperbolic PDE. It can describe propagation of water/sound/light waves, but also shock waves, e.g. in case of earthquakes.

## Well-posed - PDE

The same definition used for ODEs still applies for PDEs.

### Definition 4.15: Well-posedness

A model problem involving partial differential equations is said to be well-posed when the following conditions are fulfilled:

- ▶ *existence*: there is a solution.
- ▶ *uniqueness*: there is only one solution.
- ▶ *stability*: the solution depends continuously on the data of the problem.

Well-posedness of PDE problems depends strongly on the initial and boundary data, but we will not cover any general results for PDEs when this is obtained. Instead we will give some examples of choices of initial and boundary conditions for when a PDE is well-posed.

## Initial betingelser og rand

A PDE with derivatives with respect to time and space, is usually to be solved for a time interval and on a given spatial domain. E.g.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}, \quad t > 0, x \in (0, 1)$$

represents the temperature distribution and evolution on the interval  $x \in (0, 1)$ . Such a PDE will therefore need both data for an initial condition (at  $t = 0$ ) and for boundary conditions (at  $x = 0, x = 1$ ) for us to be able to solve it.

Although no guarantee, a simple rule for getting a well-posed problem when dealing with a PDE on square/cubic domains, is to add the same number of initial and boundary conditions as you have derivatives with time and space.

#### Remark 4.16: Number of boundary and boundary conditions

For a PDE solved in a square  $(x, y) = (0, 1)^2$ , the boundary has four parts. That is,  $x$  and  $y$  has each two boundaries ( $x = 0$  and  $x = 1$ , same for  $y$ ). If the Poisson equation is solved on this domain, one would need one boundary condition for each part of the boundary. Then there are in total two boundary conditions for  $x$  which matches the second-order derivatives with respect to  $x$ , and same for  $y$ .

But note that every part of the boundary only got one boundary conditions. For second-order PDEs this matches the “rule of thumb” by Kai Velten:

#### Remark 4.17: Rule of thumb

- ▶ Elliptic PDE: add a boundary condition. (one for each part of the boundary)
- ▶ Parabolic PDE: add a boundary condition and an initial condition at  $t = 0$ .
- ▶ Hyperbolic PDE: add a boundary condition and two initial conditions at  $t = 0$ .

There are three common types of boundary conditions:

- ▶ **Dirichlet boundary condition.**

$$u(t, \mathbf{x}) = f_D(t, \mathbf{x}) \quad \mathbf{x} \in \partial\Omega$$

When prescribing the value of the unknown on the boundary.

- ▶ **Neumann boundary condition.**

$$\nabla u(t, \mathbf{x}) \cdot \mathbf{n} = f_N(t, \mathbf{x}) \quad \mathbf{x} \in \partial\Omega$$



where  $\mathbf{n}$  is the unit normal on  $\partial\Omega$ . When prescribing the normal derivative (or flux) of the unknown on the boundary.

- ▶ **Robin boundary condition.**

$$a(t, \mathbf{x})u(t, \mathbf{x}) + b(t, \mathbf{x})\nabla u(t, \mathbf{x}) \cdot \mathbf{n} = f_R(t, \mathbf{x}) \quad \mathbf{x} \in \partial\Omega$$

When prescribing a linear combination of the value and the normal derivative of the unknown on the boundary.

When the right-hand side is equal to zero, it is known as a homogeneous Dirichlet/Neumann/Robin boundary condition. A homogeneous Neumann condition is also known as a “no flux” condition.

## Dimensjonalitet til PDE

### Definition 4.18: Dimensionality of a PDE problem

The *dimensionality* of a PDE problem refers to the number of (spatial) variables the solution depends on.

### Remark 4.19: Dimensionality of a PDE problem

The dimensionality of a PDE is not necessarily the same as the dimension of the domain the PDE is defined on. But will be equal to dimension of domain, or less.

### Example 4.20: Dimensionality of a PDE problem

The problem  $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad (x, y, z) \in (0, 1)^3$   
 $T(x, y, 1) = T_t \quad (x, y) \in (0, 1)^2$   
 $T(x, y, 0) = T_b \quad (x, y) \in (0, 1)^2$   
 $\nabla T(x, y, z) \cdot \mathbf{n} = 0 \quad (x, y, z) \text{ on sidewalls.}$

has solution

$$T(x, y, z) = T(z) = T_b + (T_t - T_b)z$$

Although the domain was three-dimensional, the solution only varies with  $z$ . Hence, the dimensionality of the problem is 1.

If solving the PDE in Example 4.20 numerically, it would be sufficient to only discretize the  $z$ -direction, since the solution depends only on  $z$ . To see this without knowing the solution, one can investigate the boundary conditions and the PDE itself.

### Remark 4.21: Discretizing PDEs and dimensionality

To reduce computational effort, one should always discretize PDEs using the lowest possible dimension.

## Symmetri

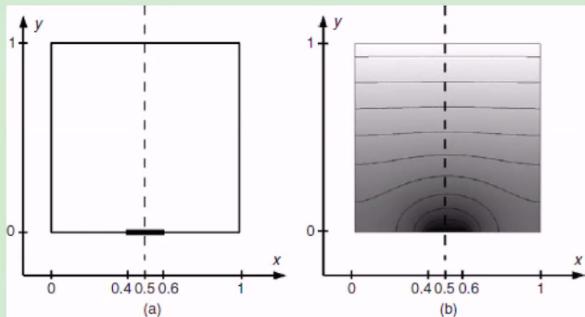
Another way to make discretization of a PDE cheaper is to take advantage of symmetry.

### Example 4.22: Symmetry of a PDE problem

The problem

$$\begin{aligned}\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} &= 0 & (x, y) \in (0, 1)^2 \\ T(x, 1) &= T_b & 0 \leq x \leq 1 \\ T(x, 0) &= T_f & 0.4 \leq x \leq 0.6 \\ \nabla T(x, y) \cdot \mathbf{n} &= 0 & \text{otherwise.}\end{aligned}$$

has solution seen in (b).



This problem is symmetric around the center line  $x = 0.5$  (mirror symmetry). Hence, only half the domain would need to be discretized. On the symmetry line, no flux boundary conditions can instead be applied.

### Remark 4.23: Discretizing PDEs and symmetry

To reduce computational effort, one should always investigate any expected symmetry when discretizing PDEs.

- ▶ If a setup is expected to e.g. only vary in the vertical and radial direction in a cylinder (rotational symmetry), computational efforts can be reduced by switching to cylindrical coordinates  $(r, \theta, z)$  and then reduce the dimensionality of the problem by only discretizing  $r$  and  $z$ .
- ▶ If a setup is expected to e.g. only vary in the radial direction in a sphere (spherical symmetry), computational efforts can be reduced by switching to spherical coordinates  $(r, \theta, \phi)$  and then reduce the dimensionality of the problem by only discretizing  $r$ .

PDE i porøse medier

PDE i bilde prosessering

EKSAMEN - 5.12