

# Kort om kursene IN1900, MAT-IN1105, IN-KJM1900

Joakim Sundnes<sup>1</sup>

Ole Christian Lingjærde<sup>1</sup>

<sup>1</sup>Department of Informatics, University of Oslo

Aug 21, 2019

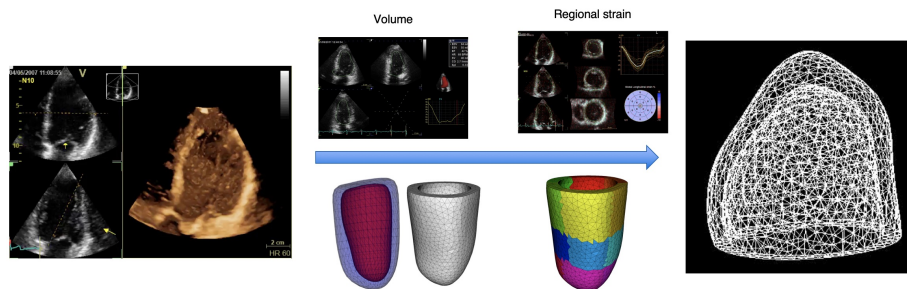
## Plan for 19 august

- Info om kursene IN1900, MAT-IN1105, IN-KJM1900
  - Generell info
  - Kort gjennomgang av info på semestersidene
  - Nyttige tips om programvare og arbeidsform
- Første introduksjon til programmering
  - Live-programmering; "Hello world!"
  - Programmering av matematiske formler

## Kort om min bakgrunn

- Master (Cand Scient) i mekanikk fra UiO (1998)
- PhD i Informatikk fra UiO (2002)
- Forsker på Simula/UiO
- Undervist IN1900 siden 2015

## Forskning; matematiske modeller av hjertet

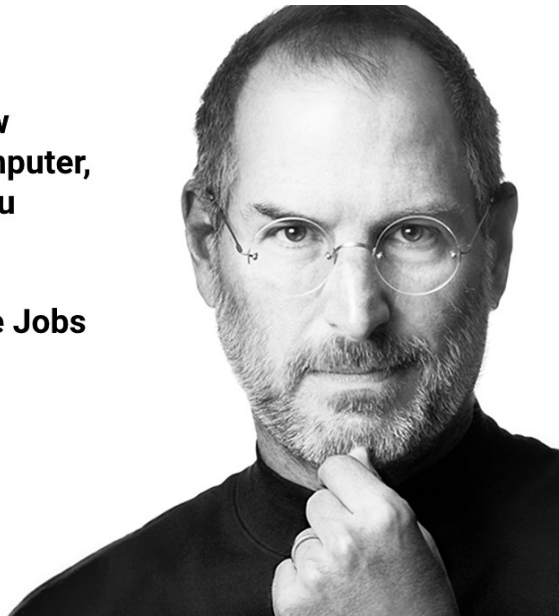


- Fysikk-baserte modeller for elektrofysiologi og mekanikk
- Systemer av differensiallikninger - umulig å løse analytisk
- Løses med ulike numeriske metoder - programmert i C++ og Python

## Hvorfor lære programmering?

**"Everyone should know how to program a computer, because it teaches you how to think."**

**Steve Jobs**



## Hvorfor lære programmering?

- Programmering blir viktigere og viktigere i industri og forskning, og brukes i flere og flere fagfelt
- Programmering vil bli brukt i veldig mange senere emner
- Hvorfor?  
Programmeringen gjør matematikk mye mer anvendbar
- Tre perspektiver på matematikk i høst:
  - tradisjonell kalkulus (MAT1100/MAT1001)
  - numerisk (datamaskinvennlig) matematikk (MAT-INF1100)
  - programmering av numerisk matematikk (IN1900/MAT-IN1105/IN-KJM1900)

## Hva lærer man i kursene

1. Generell programmering:
  - Hvordan oversetter vi et matematisk problem til kode?
  - Struktur på programmer og kode
  - Generelle verktøy og begreper, felles for alle språk
2. Python-spesifikt:
  - Python-syntaks
  - Python-spesifikke funksjoner og programmeringsstil
3. Verktøy:
  - Terminalvindu, Atom, Spyder, iPython, etc.

## **MAT-IN1105 = IN1900 uke 1-6 + MAT-INF1100**

- Uke 34: Følger forelesninger i INF1100 og MAT-INF1100
- Uke 35-40: Sammenfallende undervisning med INF1100:
  - forelesninger: uke 34-39

- gruppeøvelser (obliger): uke 35-40
- Samme midtveiseksamen som IN1900
- Fullstendig sammenfallende undervisning med MAT-INF1100 etter det

### **IN-KJM1900 = IN1900 uke 1-8 + prosjekt**

- Fullstendig sammenfallende undervisning og obliger med IN1900:
  - forelesninger: uke 34-41
  - gruppeøvelser (obliger): uke 35-42
- Samme midtveiseksamen som IN1900
- Eget kjemi-relatert programmeringsprosjekt etter det

### **All informasjon og alle beskjeder ligger på nettsidene**

- IN1900: <http://www.uio.no/studier/emner/matnat/ifi/IN1900/h19>
- MAT-IN1105: <http://www.uio.no/studier/emner/matnat/math/MAT-IN1105/h19>
- IN-KJM1900: <http://www.uio.no/studier/emner/matnat/ifi/IN-KJM1900/h19>
- Se spesielt IN1900 undervisningsplan for info om hva som skjer hver uke

### **Undervisningen består av øvelser og forelesninger**

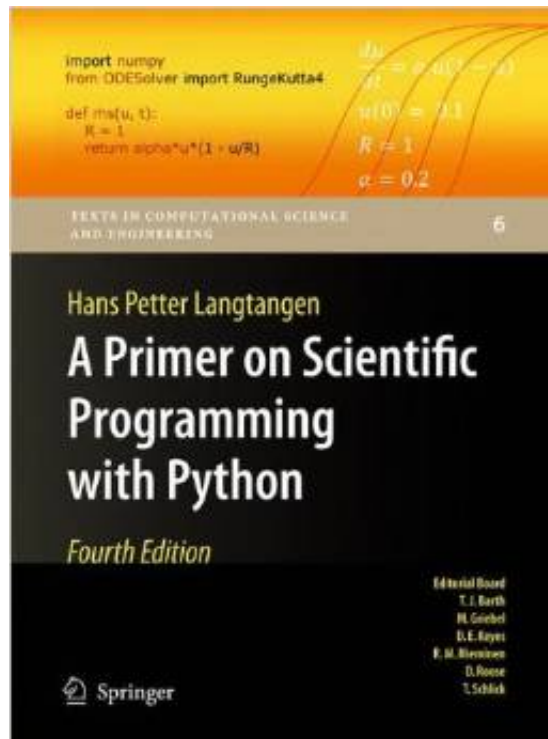
- Plenumsundervisning mandager 10.15-12 og torsdager 14.15-16 i Sophus Lies auditorium
  - Oppgaveløsning (live-programmering)
  - Gjennomgang av nytt stoff (lysbilder, notebooks)
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Delta på *alle* undervisningstimene!

## Om forelesningene

- Normalt ca 50/50 splitt mellom foiler og live-programmering
- Undervisning basert på lysbilder blir lett kjedelig, men er nyttig for å introdusere nytt stoff.
- Fra 2018: Litt høyere prioritering av live-programmering, fortsetter i 2019
- Noe introduksjon av nytt stoff blir rent selvstudium
  - Morsommere og mer lærerike forelesninger
  - Større krav til forberedelser!

## Undervisningsmaterieill

- Lærebok skrevet spesielt for IN1900/INF1100
- Vi bruker 5. utgave i høst
- 4. utgave er mulig å bruke, men nummerering av oppgaver og kapitler er forskjellig. 3. utgave og tidligere anbefales ikke
- Vi går gjennom ca ett kapittel pr uke, stort sett med ett til to nye sentrale begreper



## Undervisningsmaterieell

Nye "forelesningsnotater" i 2019.

- Legges ut etterhvert som de er klare
- Kompakt og noe oppdaterte versjoner av hvert kapittel i boka
- Legges ut som pdf-filer og Jupyter Notebooks (nytt i år, mer om dette senere)

## Python 2 vs Python 3

- Python 3 dominerer i dag
  - Python 3.7 og 3.6 er de vanligste
  - Python 2 (2.7) brukes fortsatt noe
- Foiler, oppgaver i plenum etc kommer til å være Python 3
- Eksempler i læreboka er Python 2.7

- For IN1900 er forskjellen veldig liten (se beskrivelse på emnesidene)
- Det finnes programmer for å konvertere mellom Python 2 og 3 (se læreboka)
- Mer info på emnesidene

## Det kreves innlevering av 3-5 obligatoriske oppgaver hver uke

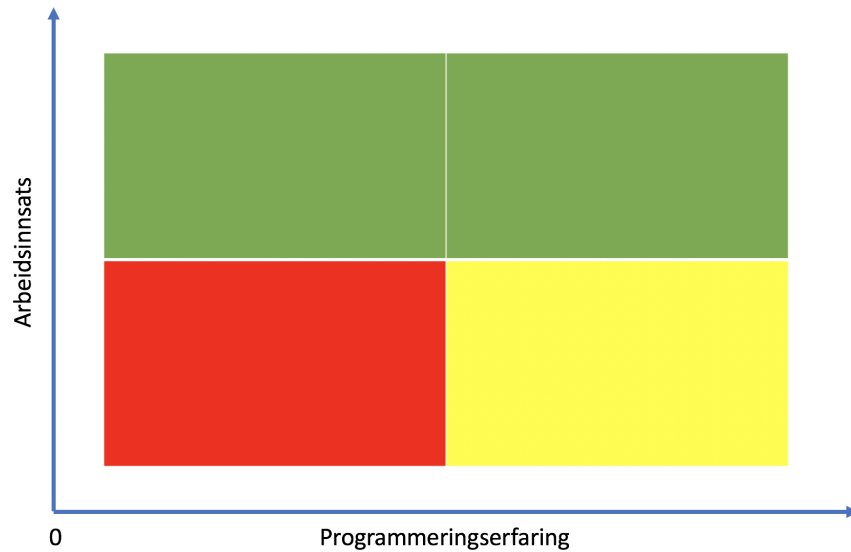
- Ukentlige innleveringsoppgaver:
  - 3-5 obligatoriske oppgaver hver uke (vurderes til bestått eller ikke bestått)
  - IN1900: Ca 2/3 av oppgavene bestått før eksamen
  - Krav MAT-IN1105: Ca 2/3 av oppgavene i uke 35-39, men det blir gitt ekstraoppgaver etter midtveiseksamen
  - IN-KJM1900: Ca 2/3 av oppgavene i uke 35-41, pluss prosjektoppgave
- Eksamen:
  - Midtveiseksamen 9. oktober - teller 25% av karakteren
  - Avsluttende eksamen 09. desember - teller 75% av karakteren

## Kursene krever ingen forkunnskaper om programmering

- Vi starter med **veldig** enkle programmer, men går relativt fort fram
- For mange er programmeringen en helt ny måte å bruke en datamaskin på
- En del nye verktøy; editor, terminalvindu, etc
- Verktøy må læres underveis; kom tidlig i gang, spør om hjelp ved behov
- Har du aldri programmert før? Kom tidlig i gang, prioriter de første ukene.
- Har du programmert Python tidligere? Følg med på hva som skjer i undervisningen likevel

[Python bakgrunnssjekk](#)

## Mer om forkunnskaper og arbeidsinnsats



### Anbefalt arbeidsform

- Før hver forelesning må du ha lest ukens forelesningsnotater, gjerne også tilhørende kapittel i læreboken
- Minimum:
  - Les ukens forelesningsnotater
  - Les om ukens læringsmål i undervisningsplanen, sørg for at du har fått med deg disse
  - Forsikre deg om at du har kontroll på de sentrale begrepene.
- Før hver oppgaveløsning i plenum må du selv ha forsøkt å løse oppgavene
- Etterarbeid oppgavene når du har sett løsning i plenum
- Nå er du klar for ukens obliger: Du kan gjøre dem på terminalstue under veiledning

### Merk:

- Hvis du kan stoffet er det vanligvis mulig å gjøre alle ukens obliger i løpet av en gruppetime (2 timer)



- Å gå rett på oppgavene uten forberedelser ender ofte i mange timers prøving og feiling

## **Piazza**

Har du spørsmål om ukeoppgaver, eksempler fra forelesninger eller programmering generelt?

- Post dem på piazza!
- Alle blir invitert til et eget IN1900-forum (link fra emnesiden)
- Gruppelærere, studenter og forelesere svarer aktivt
- Gjennomsnittlig responstid høsten 2018; ca 25 minutter

## **Du må lære programmering ved å programmere mye**

- Du kan ikke lese deg til programmering
- De fleste synes programmering er krevende i begynnelsen - så blir det utrolig gøy!
- Oppskrift på suksess: vær godt forberedt til undervisningen - det gir deg mest fritid og mest læring
- Forventet arbeid er 13 timer med IN1900 hver uke (6 t undervisning, 7 t selvstudium)

## **Hvor mye matematikk må jeg kunne på forhånd?**

- Nesten alle eksemplene i IN1900 handler om bruk av matematikk
- Vi bygger (i prinsippet) på R2 fra vgs
- Men matematikken i IN1900 er stort sett *numerisk* matematikk (MAT-INF1100)
- Vi håper at IN1900 skal belyse matematikk fra en ny vinkel og hjelpe deg til å forstå matematikk bedre samtidig som du lærer å programmere

## Alt undervisningsmateriale er på engelsk

- Muntlig undervisning foregår på norsk
- Alt skriftlig materiale er på engelsk
- Hvorfor?
  - Du finner mye informasjon om programmering på nettet og i bøker - nesten all denne informasjonen er på engelsk og da må du kunne de engelske uttrykkene
  - Mesteparten av undervisningsmaterialet på UiO er på engelsk
  - I jobbsammenheng kan du regne med at alt skriftlig foregår på engelsk
  - Boken og undervisningsmaterialet brukes ved mange utenlandske universiteter

## Bruk av egen laptop, operativsystem, etc

- Du kan fint gjennomføre kurset uten egen laptop, alle oppgaver kan gjøres på terminalstuene
- Du kan også bruke egen laptop; Python kjører på Windows, Mac og Linux
- Anbefalt arbeidsform; teksteditor + terminalvindu
- Se emnesidene for tips om programvare
- Laptophjelpen tilbyr hjelp til installering i starten av semesteret
  - 1. etasje i Ole Johan Dahls Hus (se info på <http://ifi.uio.no>)
- Hvis du har problemer med bruk av egen laptop, bruk terminalstuene til å komme i gang med programmering og oppgaver. Løs problemene senere.
- Spør om hjelp på gruppetimene

Nettbasert forkurs i informatikk anbefales sterkt!

## Viktig info om UiO-brukernavn

- Du trenger UiO-brukernavn og passord for å:
  - Logge inn på maskiner på terminalstuene
  - Logge inn på UiO-maskiner hjemmefra
  - Levere oppgaver i [devilry](#)
- Brukernavn og passord aktiveres når du har betalt semesteravgiften!
- Din UiO-bruker er også en mail-adresse ([brukernavn@student.matnat.uio.no](mailto:brukernavn@student.matnat.uio.no)). Sjekk denne jevnlig!

## Bruk av kilder på web

- Ikke overraskende er google et meget nyttig verktøy også for å lære programmering, med noen ulemper:
  - Du må vite hva du skal søke etter, dvs du må kunne en del Python for å søke effektivt
  - Mye informasjon er veldig teknisk og lite egnet for nybegynnere; krever litt trening å lese
- Tips: kryptiske feilmeldinger fra Python kan limes rett inn i google søkefelt. Gir ofte (men ikke alltid) en grei forklaring på hva som er galt
- NB: Tidligere INF1100-studenter (og andre) har lagt ut løsninger på oppgaver på web. Vær *meget* forsiktig med å bruke disse

## Mer om løsningsforslag på nett

Hva er anbefalt bruk av løsningsforslag dere finner på web?

- Unngå disse sidene og gjør oppgavene selv?
  - Ja, anbefalt!
- Prøve å gjøre oppgavene selv, se på løsningsforslag hvis man står fast?
  - Ok, men det er viktig at man prøver å løse oppgavene selv først.
- Se på løsningsforslag for å få ideer til hvordan oppgaven skal løses?
  - Lovlig, men man lærer ikke mye av det. Se heller etter eksempler i læreboka som likner på oppgaven du skal løse.

- Kopiere eller skrive av løsningsforslag?
  - Dette regnes som fusk, og man risikerer utestengelse fra UiO! Det hjelper ikke å gjøre små forandringer i programmet for at det skal se mindre kopiert ut.

## **Samarbeid om obligatoriske oppgaver**

- Samarbeid er bra, men pass på at alle bidrar, og at alle forstår de løsningene man kommer fram til
- Alle skriver sin egen kode, og leverer sin egen innlevering.
- Skriv en kommentar i innleveringen om hvem du har samarbeidet med:
  - Ved samarbeid blir koden gjerne veldig lik
  - IFI har verktøy som finner like innleveringer, og flagger dette som mulig avskrift/fusk
  - Veldig enkelt å rydde opp i hvis vi vet hvem som har samarbeidet med hvem. Tar fort mye tid ellers.

## **Tilbakemelding på undervisningen**

- Still spørsmål underveis (ikke lett i Sophus Lie...)
- Kom med synspunkter etter forelesning eller send epost
- Hva fungerer bra, hva er mindre bra?
- Er noen tema spesielt vanskelige?
- Er info på web utdatert eller feil?
- Alle innspill blir lest og hørt, selv om ikke alle forslag lar seg gjennomføre
- Nytt for 2019: evalueringsgruppe av 4-5 studenter (frivillig)

## Viktig hjemmearbeid første uke

- Ta nettbasert forkurs i informatikk
- Les gjennom informasjon på semestersidene:
  - Sjekk ut alle lenker fra hovedsiden
  - Bli kjent med undervisningsplanen
- Finn en terminalstue med Linux-maskiner:
  - Logg inn med UiO-brukernavn
  - Åpne en editor og et terminalvindu
  - Skriv `python` i terminalvinduet og se hvilken versjon du kjører (øverste linje, sannsynligvis 3.7, 3.6 eller 2.7). Avslutt Python ved å skrive `quit()` eller tast *Ctrl-D*.
- Test eventuelt installasjon på egen maskin (se info på web)