

Kort om IN1900, MAT-IN1105, IN-KJM1900 og IN-GEO1900

Joakim Sundnes¹

Ole Christian Lingjærde¹

¹Department of Informatics, University of Oslo

Aug 22, 2023

0.1 Plan for 22 august

- Info om kursene IN1900, MAT-IN1105, IN-KJM1900, IN-GEO1900
 - 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

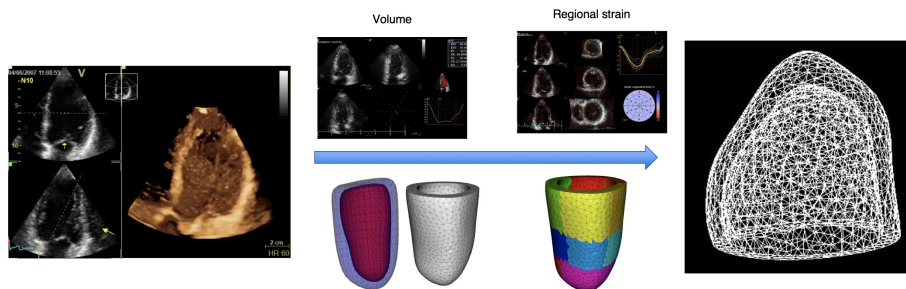
0.2 Opptak av forelesninger

- Alle forelesninger vil bli tatt opp og lagt ut på nett.
- Tilgang til opptak krever innlogging
- Eventuelle spørsmål fra salen vil komme med i opptaket, men kan redigeres bort hvis det er ønskelig.

0.3 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

0.4 Forskning; matematiske modeller av hjertet

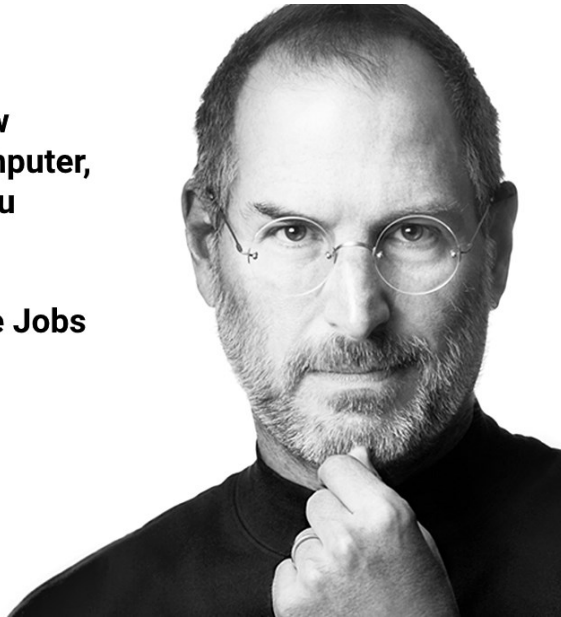


- Fysikk-baserte modeller for elektrofysiologi og mekanikk
- Systemer av differensiallikninger - umulig å løse analytisk
- Løses med ulike numeriske metoder - programmert i C++ og Python

0.5 Hvorfor lære programmering?

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

Steve Jobs



0.6 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/IN-GEO1900)

0.7 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, VS code, Spyder, iPython, etc.

0.8 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
 - ukeoppgaver og gruppeundervisning (obliger): uke 35-40
- Samme midtveiseksamen som IN1900
- Fullstendig sammenfallende undervisning med MAT-INF1100 etter det

0.9 IN-KJM1900 og IN-GEO1900 = IN1900 uke 1-6 (ca) + prosjekt

- Uke 34-40: Samme forelesninger og obliger som IN1900
 - forelesninger: uke 34-40
 - gruppeøvelser (obliger): uke 35-40
- Samme midtveiseksamen som IN1900
- Eget kjemi/geo-relatert programmeringsprosjekt etter det

0.10 All informasjon og alle beskjeder ligger på nettsidene

- IN1900: <http://www.uio.no/studier/emner/matnat/ifi/IN1900/h23>
- MAT-IN1105: <http://www.uio.no/studier/emner/matnat/math/MAT-IN1105/h23>
- IN-KJM1900: <http://www.uio.no/studier/emner/matnat/ifi/IN-KJM1900/h23>
- IN-GEO1900: <https://www.uio.no/studier/emner/matnat/ifi/IN-GEO1900/h23/>
- Se spesielt IN1900 undervisningsplan for info om hva som skjer hver uke

0.11 Undervisningen består av øvelser og forelesninger

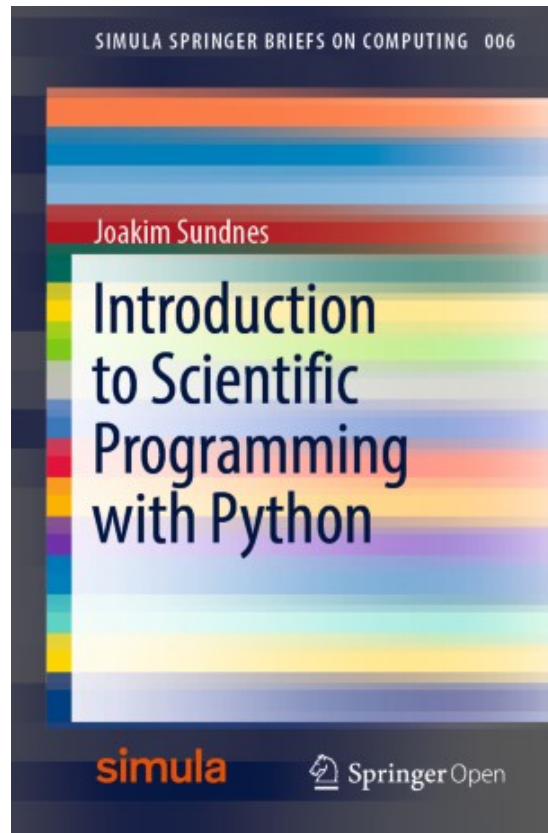
- Plenumsundervisning tirsdager 10.15-12 og torsdager 14.15-16
 - Oppgaveløsning (live-programmering)
 - Gjennomgang av nytt stoff (lysbilder, notebooks)
- 2 t oppgaveløsning på terminalstue i mindre grupper der du kan få individuell veiledning
- Anbefaling: Delta på *alle* undervisningstimene i starten av semesteret!
- Opptak av forelesningene blir lagt ut på emnesidene (etter ca 3-4 timer)

0.12 Om forelesningene

- Tidligere ca 50/50 splitt mellom lysbilder og live-programmering
- Undervisning basert på lysbilder blir lett kjedelig, men er nyttig for å introdusere nytt stoff.
- Fra 2019: Gradvis mer og mer live-programmering og mindre lysbilder. Fortsetter med dette i 2023.
- Noe introduksjon av nytt stoff blir rent selvstudium
 - Morsommere og mer lærerike forelesninger
 - Større krav til forberedelser!

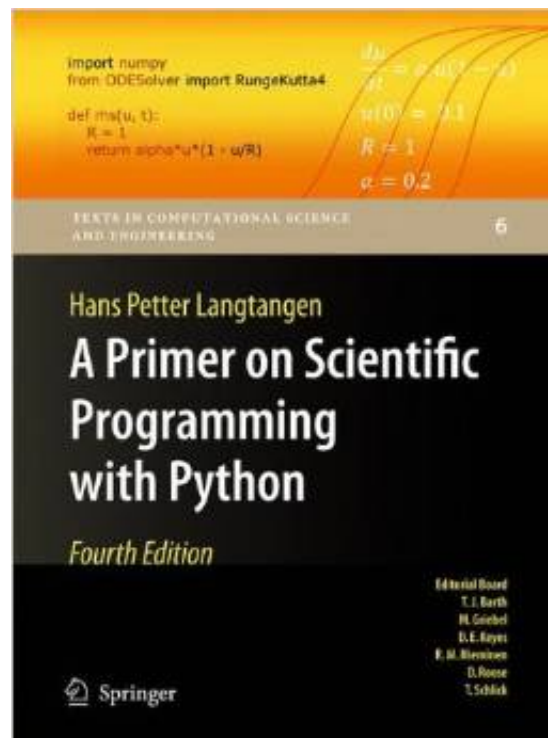
0.13 Undervisningsmaterieell (1)

- Samme lærebok siden høsten 2020
- Kortversjon av boka som har vært brukt tidligere
- Oppdatert til nyere versjoner av Python (≥ 3.6)
- Vi går gjennom ca ett kapittel pr uke, stort sett med ett til to nye sentrale begreper
- Kompendium for noen tema senere i semesteret
- Gratis tilgjengelig for alle, link på emnesiden



0.14 Undervisningsmaterieell (2)

- Lærebok brukt fram til 2019
- Nyttig tilleggsressurs, som har mye mer info og flere eksempler enn den nye boka
- Skrevet i Python 2, men forskjellene er små.
- Gratis tilgjengelig for UiO-studenter, må være pålogget UiO-nettverk for nedlasting



0.15 Python 2 vs Python 3

- Python 3 dominerer i dag
 - Python 3.11 er nyeste stabile versjon
- Lærebok, foiler, oppgaver i plenum etc er Python 3 (≥ 3.6)

- Gammel lærebok (Langtangen) er Python 2.7
- For IN1900 er forskjellen relativt liten (se mer info på emnesidene)

0.16 Nytt i år: færre obligatoriske oppgaver

- Fram til 2022:
 - Obligatoriske innleveringer hver uke
 - Jevn jobbing gjennom semesteret -> veldig gode resultater
- Fra 2023:
 - Ukentlige oppgaver frivillig, men anbefales sterkt!
 - Oppgaver kan leveres inn, for retting og tilbakemelding fra gruppelærere
 - To obligatoriske oppgaver, den første i uke 36
- Eksamen (uendret):
 - Midtveiseksamen 14. oktober - teller 25% av karakteren i IN1900/IN-KJM1900/IN-GEO1900, og 1/3 av karakteren i MAT-IN1105
 - Avsluttende eksamen 2. desember

0.17 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](#)

0.18 Mer om forkunnskaper og arbeidsinnsats



0.19 Anbefalt arbeidsform

- Minimum:
 - Les ukens kapittel i læreboka
 - 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, på egen hånd eller på gruppetime

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

0.20 Nettforum

For spørsmål om ukeoppgaver, eksempler fra forelesninger eller programmering generelt.

- Tidligere Piazza og Padlet, fra 2022 Astro Discourse
- Link legges ut på emnesiden, blir klart i løpet av første uke
- Gruppelærere, studenter og forelesere svarer aktivt
- Vanligvis mye raskere responstid enn spørsmål på mail

0.21 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)

0.22 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

0.23 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

0.24 Bruk av egen laptop, operativsystem, etc

- Man kan gjennomføre kurset uten egen laptop, alle oppgaver kan gjøres på terminalstuene
- Egen laptop anbefales; Python kjører på Windows, Mac og Linux
- Anbefalt arbeidsform; enten Spyder (IDE) eller teksteditor + terminalvindu (mer om dette senere)
- Se emnesidene for tips om programvare
- 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!

0.25 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). Sjekk denne jevnlig!

0.26 Bruk av kilder på web

- Ikke overraskende er Google et veldig nyttig verktøy også for å lære programmering, med noen ulemper:
 - Du må vite hva du skal søke etter, dvs du må kunne litt 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 en grei forklaring på hva som er galt
- **NB:** Tidligere IN1900-studenter (og andre) har lagt ut løsninger på oppgaver på web. Vær *veldig* forsiktig med å bruke disse

0.27 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 eller forelesningene 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.

0.28 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.

0.29 Tilbakemelding på undervisningen

- Still spørsmål underveis
- Send epost med synspunkter etter forelesning
- Hva fungerer bra, hva er mindre bra?
- Er noen tema spesielt vanskelige?
- Er info på web utdatert eller feil?
- Alle innspill blir vurdert, selv om ikke alle forslag lar seg gjennomføre

0.30 Viktig hjemmearbeid første uke

- Ta nettbasert [forkurs i informatikk](#)
- Les gjennom informasjon på semestersidene:
 - Sjekk ut alle lenker fra semestersiden
 - Bli kjent med undervisningsplanen
- Installer Python og VS code på egen maskin (se info på semestersiden)
- Eventuelt, finn en terminalstue, gjerne med Linux-maskiner:
 - Logg inn med UiO-brukernavn

- Åpne en editor VS Code og et terminalvindu (Terminal (Mac) eller Powershell (Windows))
- Skriv `python` i terminalvinduet og se hvilken versjon du kjører (øverste linje, sannsynligvis 3.7-3.11, kanskje 2.7). Avslutt Python ved å skrive `quit()` eller tast *Ctrl-D*.