

# TDT4127 Programmering og Numerikk

Torbjørn Ringholm, forsker, Institutt for matematiske fag

Kontor: 1338 i Sentralbygg II (Gløshaugen)

Epost: [torbjorn.ringholm@ntnu.no](mailto:torbjorn.ringholm@ntnu.no)

# Innhold i emnet

- Lære grunnleggende programmering
  - Programmeringsspråket vi bruker er **Python**
  - Programmeringsundervisning er delt med TDT4110
- *Forskjell fra TDT4110:*
  - I stedet for *IT-teori* skal dere lære om **numerikk**.
- Generell kunnskap om numerisk matematikk
  - Kunnskap om praktiske algoritmer for å beregne løsninger på matematiske problemer:
    - Integrasjon
    - Løsning av vanlige ligninger
    - Løsning av ordinære differensialligninger

# Hvorfor trenger vi å kunne dette?

- IKT brukes «overalt» og kunnskap om IKT er en grunnleggende forutsetning innenfor tekniske og *ikke-tekniske* yrker i dag.
- Eksempler: Digitalisering av helsetjenester, musikkproduksjon, reiseplanlegging, utdanning, alle slags simuleringer, robotikk osv.
- Skal man opp og fram her i verden, lønner det seg å kunne programmere.

# Hvorfor trenger vi å kunne dette?

- Hvordan kalkulerer man egentlig integraler som

$$\int_a^b \tan(\cos(\sin(e^{x^5}))) dx?$$

*Svar: Ved å tilnærme dem veldig nøyaktig.*

- Det samme går for å løse ligninger som

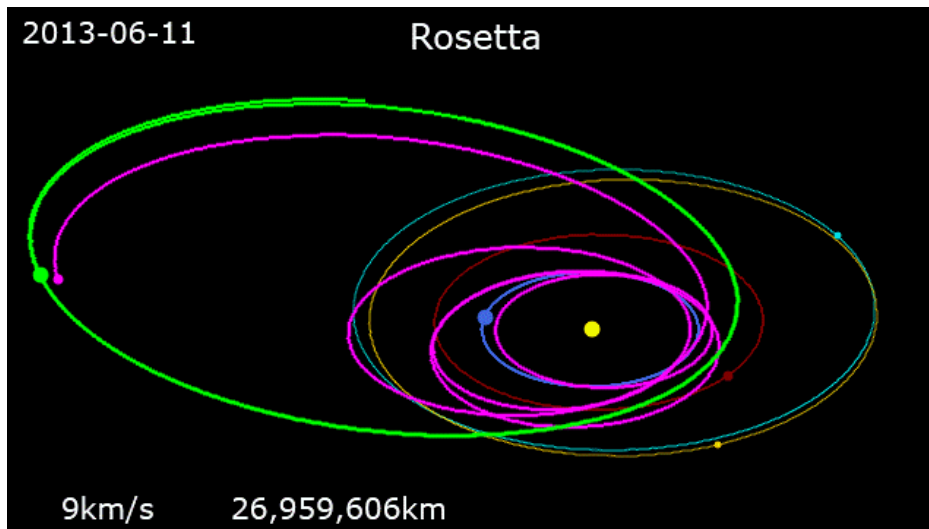
$$\log(\cos(x^2)) = \frac{e^{x^3}}{1 + \sqrt{x}}$$

*Vi løser dem tilnærmet, ved å bruke algoritmer.*

# Hva kan vi bruke dette til?

Numerikk er en grunnleggende del i alle beregningskrevende fagfelt.

Rosettalandingen (2016) ville vært umulig uten pinlig nøyaktige beregninger.



Kilde: [https://en.wikipedia.org/wiki/Rosetta\\_\(spacecraft\)](https://en.wikipedia.org/wiki/Rosetta_(spacecraft))

# Mål for Programmering og Numerikk

**Kunnskapsmål:** studentene skal

- Ha *grunnleggende kunnskap* om grunnelementene i *prosedyreorientert programmering*
- Ha *grunnleggende kunnskap* om prosessen fra *problem* til *fungerende program*
- Ha *grunnleggende kunnskap* innen *numeriske metoder*

Vurderingsform:

- *Digital* eksamen **30.11.2018**, kl. 09:00.

# Mål for Programmering og Numerikk

**Ferdighetsmål:** studentene skal

- Kunne anvende grunnelementene i *praktisk, prosedyreorientert programmering*.
- Kunne bruke relevante *programmeringsverktøy*.
- Kunne *analysere* et problem, finne *prosess* og *datastrukturer* som løser problemet, formulere en *løsning* som *pseudokode* eller *flytskjema*, og *programmere* og *teste* en løsning i **Python**.
- Kunne gjennomføre mindre *programmeringsprosjekter*.
- Kunne utføre *numeriske beregninger* i **Python**.

# Hvordan vil undervisningen foregå?

- Forelesninger:
  - Numerikk: 1 time, spesielt for TDT4127
  - Python: 2 timer, delt med TDT4110
  - Øvingsforelesninger: 2 timer, delt med TDT4110
- Obligatoriske øvinger:
  - Ordinære øvinger: «hjemmelekse»
    - 8 stk, godkjennes av stud.ass. på datasal i lab-timer
  - Auditorieøvinger: «eksamensforberedende»
    - 2 stk, gjennomføres i auditorium som en eksamen (digital)
  - Krav: 8 av 10 godkjente øvinger, hvorav minst 1 auditorieøving.
    - I tillegg: En obligatorisk prosjektoppgave tilsvarende 2 øvinger.
- Egen innsats er avgjørende!



# Når/hvor vil undervisningen foregå?

- Forelesninger:
  - Numerikk: fredager 16:15-17:00 i R3
  - Python: tirsdager 10:15-12:00 i R1
    - Kan bli **VELDIG** fullt. Anbefaler å vurdere å gå i:
      - Fredager 08:15-10:00 i F1
      - Mandager 14:15-16:00 i R1
      - Onsdager 08:15-10:00 i EL5
  - Øvingsforelesninger: fredager 14:15-16:00 i F1
- Øvingstimer:
  - Laboratorietimer alle dager 08:15-20:00 i A-blokken på Realfagbygget (A2-121, A3-100, A3-103).
    - «Drop in»-timer, bare still opp og meld dere i køen
    - TDT4110 vil få tilbud om faste labtider per studieprogram, dette gjelder ikke dere pga individuelle timeplaner.

# Tentativ forelesningsplan

- Nytt fag – ikke alt er satt i stein!

Uke	Programmering	Numerikk
34		Introduksjon
35	Introduksjon til programmering	Introduksjon til programmering
36	Program, variable og input	Oppfriskning av matematikk
37	Betingelser og logiske uttrykk	Numerisk integrasjon
38	Løkker (for- og while-løkker)	Numerisk integrasjon
39	Definisjon og bruk av funksjoner	Numerisk ligningsløsning (1D)
40	Funksjoner med retur og moduler	Numerisk ligningsløsning (1D)
41	Lister og tupler	Løsning av lineære systemer
42	Mer om strenger	Løsning av lineære systemer
43	Filer og unntak	Numerisk ligningsløsning (flerdim)
44	Oppslagslister og mengder	Numerisk løsning av differensialligninger
45	Rekursjon og algoritmer	Numerisk løsning av differensialligninger
46	Lage større systemer i Python	Numerisk integrasjon med adaptiv Simpson
47	Repetisjon	Repetisjon

# Pensum

- Programmering:
  - Alle foiler, øvinger og løsningsforslag er pensum
  - Bok: «Starting out with Python», 4th ed, Tony Gaddis, Pearson
  - Andre utgaver kan også brukes, se fagets nettside.
- Numerikk:
  - Forelesninger, foiler, øvinger og eventuelle notater.
  - Ekstramateriale: «Programming for Computations - Python»  
<https://link.springer.com/book/10.1007%2F978-3-319-32428-9>

# Mer om øvingene

- **Øvingene** er der man lærer mest og står mest fast
  - Viktig å gjøre dem selv og forstå dem selv
  - Laboratorieøvinger (8 stk) veiledes av stud.ass. 08:15-20:00.
    - Sitter du fast, stikk innom og spør!
    - Har du vanskeligere spørsmål, spør und.ass./vit.ass.
    - Kan velge visse oppgaver selv, men **må** gjøre numerikkoppgavene.
  - Online hjelp og diskusjon på Piazza:
    - <https://piazza.com/ntnu.no/fall2018/tdt4110>
  - Auditorieøvinger (2 stk) gjøres i auditorium (gitt tidspunkt)
    - Må ha minst én aud.øving godkjent
    - Foregår på samme plattform (Inspira) som eksamen
- **Jobb med og forstå øvingsoppgavene!**

# Informasjonskanaler og lignende

- Blackboard brukes til beskjeder og øvinger
- All info om faget ligger på fag-wiki
  - <https://www.ntnu.no/wiki/display/itgk/Forside>
- Online diskusjon og spørsmål på Piazza
  - <https://piazza.com/ntnu.no/fall2018/tdt4127/home>
- Youtube-videoer om programmering av Guttorm
  - <https://www.youtube.com/channel/UCNwXyHIGGOWZLzTy0-hM63w/feed>

# Referansegruppe

- Kvalitetssikring av faget, spesielt viktig siden det er nytt
  - Tilbakemelding på pensum og læringsaktiviteter
  - Forbedringsforslag på kort og lang sikt
- Krever lite arbeid (ca. 4 timer totalt)
- Tar seg pent ut på CV
- En representant per program?

# Viktig til neste gang

- Husk at programmeringsparallellene er ganske fulle!
  - Kan være lurt å bytte fra den oppsatte parallellen
- Still opp på øvingsforelesning
  - Det blir gitt viktig informasjon om øvinger
  - Det blir gitt hjelp til å installere Python
  - **Ta med egen datamaskin!**
  - Det blir nærmere informasjon om brukerstøtte og praktiske ting
- Numerikkforelesninger starter opp om to uker
  - Neste uke er det intensiv kom-i-gang-forelesning med Python

Spørsmål?