TDT4127 Programming and Numerics

Torbjørn Ringholm, researcher, Department of Mathematical Sciences
Office: 1338 in Sentralbygg II (Gløshaugen)
Email: torbjorn.ringholm@ntnu.no
Course contents

• Learn basic programming
  – The programming language we will use is **Python**
  – Programming lectures are shared with TDT4110
  – **Note:** Programming lectures are given in Norwegian

• *Differences* from TDT4110
  – Instead of ICT theory you will learn about **numerics**.

• **General knowledge of numerical mathematics**
  – Knowledge of practical algorithms to approximate solutions to mathematical problems such as
    • Integration
    • Solving equations
    • Solving ordinary differential equations
Why do we need to know this?

• ICT is used everywhere and knowledge of ICT is a fundamental requirement within technical (and to some degree non-technical) jobs today.

• Examples: Digitalization of health services, music production, travel planning, education, all kinds of simulation, robotics, computer-aided design etc.

• It pays to know programming – it is much easier to get a job with programming skills.
Why do we need to know this?

- How does one calculate integrals like

\[ \int_{a}^{b} \tan(\cos(\sin(e^{x^5}))) \, dx? \]

**Answer:** By approximating them very closely.

- The same goes for solving equations like

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

*We solve them approximately, using algorithms*
What can we use this for?

Numerics is a centerpiece of all computation-based sciences.

The Rosetta landing (2016) would not have been possible without excruciatingly precise calculations.

Kilde: https://en.wikipedia.org/wiki/Rosetta_(spacecraft)
Goals for Programming and Numerics

Knowledge goals: The students will

• Have basic knowledge of procedural programming.
• Have basic knowledge of the process from problem to working program
• Have basic knowledge of numerical methods

Evaluation form:

• Digital exam 30.11.2018, at 09:00.
Goals for Programming and Numerics

Skill goals: The students will

• Be able to apply basic concepts in practical procedural programming.
• Be able to use relevant programming tools.
• Be able to analyze a problem, find a process and data structure to solve it, formulate a solution as pseudocode or a flow scheme and program and test a solution using Python.
• Be able to complete minor programming projects.
• Be able to perform numerical calculations using Python.
How will the teaching be organized?

- **Lectures:**
  - Numerics: 1 hour, only for TDT4127
  - Python: 2 hours, shared with TDT4110
  - Exercise lectures: 2 hours, shared with TDT4110

- **Mandatory exercises:**
  - Ordinary exercises: «homework»
    - 8 in total, approved by student assistant during lab hours
  - Auditorium exercises: «exam preparation»
    - 2 in total, done in auditoriums in the same way as exams (digital)
  - Requirements: 8 out of 10 approved exercises, of which at least 1 must be an auditorium exercise.
    - In addition: A mandatory project exercise equivalent to two exercises.

- **Your own effort in doing this is key!**
When/where will the teaching be?

• Lectures:
  – Numerics: Fridays 16:15-17:00 in R3
  – Python: Tuesdays 10:15-12:00 in R1
    • Can be VERY full. You should consider swapping to:
      – Fridays 08:15-10:00 in F1
      – Mondays 14:15-16:00 in R1
      – Wednesdays 08:15-10:00 in EL5
  – Exercise lectures: Fridays 14:15-16:00 i F1

• Exercise hours:
  – «Laboratory hours» all days 08:15-20:00 in the A-block of Realfagbygget (A2-121, A3-100, A3-103).
    • «Drop in»-hours, just show up and get in the queue
    • TDT4110 will be offered «personal» lab hours based on study programs
      – this does not apply to you due to your individualized schedules.
Tentative lecture plan

• This is a new course – not all is set in stone!

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

• Programming:
  – All foils, exercises and solutions thereof
  – You can also use older versions, see the course wiki

• Numerics:
  – Lectures, foils, exercises/solutions and additional notes.
  – Extra material: «Programming for Computations - Python»
More about the exercises

- **The exercises** is where you learn and get stuck the most
  - Important to do them yourselves and understand them
  - Regular exercises (8 in total) guided by stud.ass. 08:15-20:00.
    - If you’re stuck, drop by and ask!
    - For tougher questions, ask und.ass. or vit.ass.
    - You can choose certain exercises yourselves but **must** do the numerics exercises.
  - Online help and discussion at Piazza:
    - [https://piazza.com/ntnu.no/fall2018/tdt4110](https://piazza.com/ntnu.no/fall2018/tdt4110)
  - Auditorium exercises (2 stk) are done in auditorium at given times
    - Must have at least one such approved to take the final exam
    - Given on the same digital platform (Inspera) as the final exam

- **Work with and understand the exercises!**
Information channels

• Blackboard is used for messages and exercises
• All info about the course is on the course wiki
  – https://www.ntnu.no/wiki/display/itgk/Forside
• Online discussion and questions at Piazza
  – https://piazza.com/ntnu.no/fall2018/tdt4127/home
• Youtube videos on programming by Guttorm
  – https://www.youtube.com/channel/UCNwXyHIGGOWZLzTy0-hM63w/feed
Reference group

• Quality assurance for the course, especially important since it is new
  – Feedback on the curriculum and learning activities
  – Proposals for improvement in the short and long terms
• Requires little work (ca. 4 hours in total)
• Looks good on a CV
• One representative per program?
Important before next time

• Remember that the programming lectures are full!
  – Can be smart to change from the given lectures
• Attend the exercise lecture
  – Important information on exercises will be given
  – You will receive help installing Python
  – Bring your own computer!
  – More information will be given about e.g. user support
• Numerics lectures start in two weeks
  – Next week is intensive start-up lecture with Python
Questions?