Numerical Methods for Chemical Engineers

Study guide for 6E5X0

Dr.ir. Ivo Roghair, Prof.dr.ir. Martin van Sint Annaland

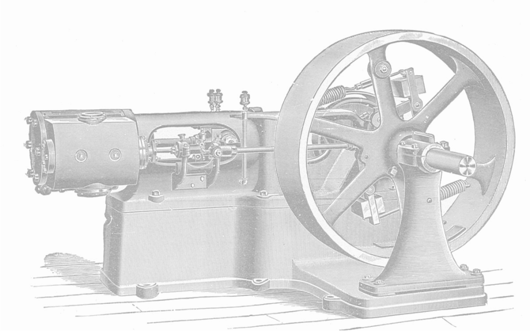
Chemical Process Intensification group
Eindhoven University of Technology

Numerical Methods (6BER03), 2024-2025

Numerical Methods

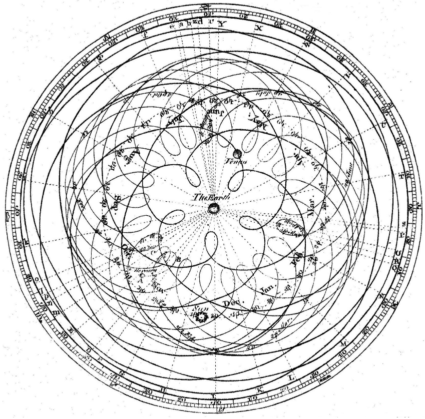
"Simulation and mathematical modeling will power the twenty-first century the way steam powered the nineteenth."

— W.H. Press[★]



[★] Author of Numerical recipes, in "The Nature of Mathematical Modeling" by Neil Gershenfeld

Ptolemy and the almagest



∼150 AD. Development of numerical approximations to describe the motions of the heavenly bodies with accuracy matching reality sufficiently.

Numerical Methods

- Numerical analysis is concerned with obtaining approximate solutions to problems while maintaining reasonable bounds of error...
- ...because it is often impossible to obtain exact answers ...
- Numerical analysis makes use of algorithms to approximate solutions

Used in many fields... Including Chemical Engineering

- Description of reactors and separators (dynamic and steady state)
- Computational fluid dynamics
- Thermodynamic equations of state
- Optimizing process performance
- Design and synthesis of processes
- Regression of data, e.g. isotherms, kinetics, ...

Course Schedule

Lecture	Topic	Teacher
1	Programming and algorithms (1)	IR
2	Programming and algorithms (2)	IR
3	Numerical errors	IR
4	Linear eqns: direct methods	IR
5	Linear eqns: iterative methods	IR
6	Interpolation + integration	IR
7	Non-linear equations (1)	MSA
8	Non-linear equations (2)	MSA
9	ODEs (1)	MSA
10	ODEs (2)	MSA
11	PDEs	MSA
12	Regression and Optimization	IR

Course Objectives

- Gain experience with programming basics and algorithm design (Python, Excel)
- Understand the background of important numerical algorithms used in CEC
- Recognize and solve (systems of) linear, non-linear and differential equations numerically
- Learn about numerical integration, interpolation, optimization
- Apply your skills to solve realistic assignments and small sample problems

Prerequisites

The following courses should give you enough background knowledge to follow this course comfortably:

- Calculus
- Linear Algebra
- Some basic Python experience
  - We will shortly cover some aspects on Python programming in the first lectures. Detailed documents and courses are provided on Canvas, for your own reference.

You will definitely need a laptop with Python and Excel installed!

Course Materials

- Lecture slides (+ lecture recordings?)
- Python scripts
- Additional articles
- There are some useful books for those seeking more in-depth knowledge and alternative methods, not mandatory:
  - Numerical recipes, W.H. Press et al.
  - Numerical methods for chemical engineering, K.J. Beers
  - Numerical methods for chemical engineers, A. Constantinides
  - Python Crash Course, 3rd Edition, Eric Matthes

Look on Canvas for the slides, exercises, scripts, assignments and additional documentation on Python.

Assessment

4 assignments

- Each 20% of the final result
- Done in groups of 2 persons, see Canvas → People → NM Groups
- Short report (template provided, Overleaf and Canvas)
- Assignment 1–3 graded through peer review

Final exam

- Practical and theoretical questions, covering *all topics*
- Exam taken on your own computer
- You can use the slides and modules documentation
- Sample exam will be released before Christmas
- Grade of the final exam needs to be at least a 5.0

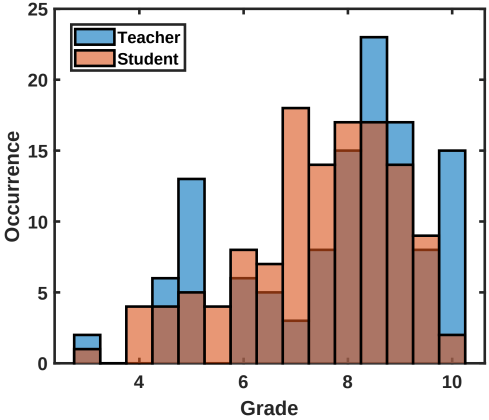
Assignment grading: peer assessment & feedback (1)

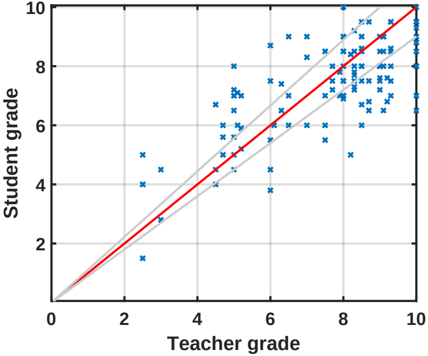
- Assignments are graded through supervised peer-assessment. After the deadline, each *person* will grade 2 other assignments. Rubrics are available to maintain a consistent assessment among different groups. Criteria are:
    - Functionality
    - Code style
    - Visualisation
    - Analysis
- The review should be done within 3 days using rubrics, additional feedback should be supplied to establish its validity. We will assess the quality of the feedback, and grade it by a multiplier (0.8-1.2).
- You can challenge one or more reviews by submitting a rebuttal;
- The final grade will be the averaged grade from the remaining peer-assessments (group), multiplied by the peer-review quality (individual), with a max. of 10.
- When statistics are poor (≤ 2 reviews), the assignment will be graded by the instructor, which discards all remaining peer-reviews.

Assignment grading: peer assessment & feedback (2)

- Along with the rubrics, you will give each other specific comments for improvement: What are you impressed with; why did you score a certain criteria low; how to improve the code or visuals, etc. Give at least 3 tips and 3 tops.
- Grades for an assignment are released only when proper assessment and feedback have been given.
- Rebuttals are turned in through an additional assignment. A rebuttal should convince us and provide evidence and in-depth argumentation why a particular review is flawed. We will evaluate the rebuttals and discard a peer-review if it is indeed disproportionate.
- We are getting help from student assistants to make the process go smoothly. I will show a possible solution after the deadline.
- Grading with rubrics: don't be afraid to use the full spectrum.

Peer assessment in the past





Peer assessment summary

Complete document can be read through Canvas, here's the summary:

- The assignment and report template are released along with the grading rubrics
- Canvas automatically performs a plagiarism check
- The lecturer will give a short overview of how the assignment could have been solved (point of reference)
- Students have 3 days for double-blind peer assessment
- Students have the opportunity to challenge their reviewers (rebuttal)
- Lecturers and TAs will check the review quality, as well as the reports that have very low or very high marks or large deviations among reviews coarsely. I will provide a full correction when no suitable peer-reviews have been done.
- If a student fails to produce a good, timely peer-review, their grade for the assignment will be marked NA.

Assignment handout and deadlines

Hand-in your assignments via Canvas

- Deadlines are given on Canvas as well
- Deliver the report in PDF format
- Send along the scripts + necessities in a .zip
- Use your student ID instead of your name for identification purposes
- Be aware that a .docx stores the original author name as metadata.

Course Philosophy

- This is a hard course! It will take a lot of hours, especially if you have no coding experience.
- We are here to help you. To have nice discussions, to show alternative ways, etc. Clarify language subtleties, or suggestions. Not to give away the answers.
- We encourage research and independent learning. It comes down to paying attention, repetition of the concepts, and practice practice practice!
- We try to make the lectures interactive, working on examples and creating scripts as we go. It is advised that you work along with us to get the most out of this course!
- It is ok to discuss the general approach to solving the problem, or to get a hint, or several hints, if you get stuck while solving a problem, but work out the details of the solution in your own group.
- It is **not ok** to take someone else's solution and simply copy their scripts, answers, etc.
- Take regular breaks! Better 6 times per hour a short break than 1 time per hour a long break. Use keyboard shortcuts, or a mouse if you must. Set up screen brightness to a pleasant value. Stretch.

Contact information

Ivo Roghair

- E-mail: i.roghair@tue.nl
- Office: STW 0.37

Martin van Sint Annaland

- E-mail: m.v.sintannaland@tue.nl
- Office: STW 0.39

Some Acknowledgements

Some Real Acknowledgements

For inspiration, examples and exercises:

- To Roel Verstappen of Groningen University
- To Johan Hult of Cambridge University
- To Edwin Zondervan of University of Twente
- To edX: MITx course "Introduction to Computer Science and Programming Using Python"