

Lecture 1: Introduction to the Course Modelling Complex Systems (Spring 2026)

Jordi-Lluís Figueras

Matematiska Institutionen, Uppsala Universitet

March 24, 2026

WELCOME!



TO THE MASTER COURSE IN
MODELLING COMPLEX SYSTEMS

$$+ \Delta X = f(X, Y)$$

$$+ - = P(t)$$



Lecture goals

- Present the course structure, contents, and evaluation.
- Give a first overview of complex systems and of the four thematic blocks.

Course philosophy

This course combines:

- mathematical modeling,
- and computational simulation in Python.

Students are expected to:

- develop models from first principles,
- implement and simulate systems,
- critically validate computational outputs.

Course organization

- Duration: March 24 – May 29, 2026.
- Structure: 19 lectures, 5 labs, and final presentations.
- The course progresses from modeling ideas to computational implementation.
- Labs may sometimes precede full theoretical coverage and should be understood as exploratory learning.

Entry requirements

- 120 credits, including 40 credits in mathematics.
- One of the courses Scientific Computing KF and Introduction to Scientific Computing.
- English 6 or equivalent.

Learning outcomes, content, and assessment

- Students should be able to implement simulation methods, use approximations, and explain how complex systems are measured and studied computationally.
- The syllabus highlights numerical methods, software for simulations, random processes, diffusion approximations, chaos, cellular automata, epidemic models, and interacting particle systems.
- Assessment is based on computer laboratory and project work.

Syllabus page: <https://www.uu.se/en/study/syllabus?query=ccebcfca-b4e2-11f0-bd83-743a3e5145ea>

Global course structure

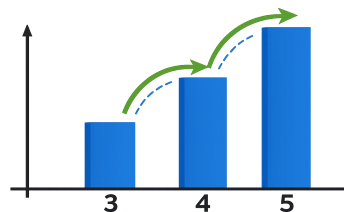
Block	Topic	# Lectures	Lab
0	Intro to course, Complex Systems, and Generative AI tools	1	Lab 1
1	Dynamical systems	5	Lab 2
2	Cellular automata	3	Lab 3
3	Networks	4	Lab 4
4	Particle systems	6	Lab 5

Evaluation

Component	Points
4 projects	60
Presentation	10
Total	70

Grades:

- 3: at least 40 points
- 4: at least 50 points
- 5: at least 60 points



Labs, projects, and group work

- The course has 5 labs and 4 graded projects, with the labs supporting the project work.
- Each lab-project follows material introduced in about three lectures.
- Work should be done in groups of about 5 students (minimum 4, maximum 6).
- Only one assignment per group should be submitted.
- Work should be distributed so that everyone contributes in coding, writing, and modelling.
- All code should be written in Python.
- All text should be written in LaTeX and submitted in pdf.

Submission, grading, and practical information

- Lab-projects are submitted as assignments in **Studium**.
- Deadline for Project 1: 14 days after the lab.
- Deadline for the other projects: 10 days after the corresponding lab.
- Each project gives at most 15 points, for a total of 60 project points.
- Each team presents one project; 10 points are reserved for the presentation.
- One goal of the oral presentation is to show complete understanding of the submitted work.
- If this understanding is not demonstrated, the points from the written project will not be added.
- If necessary, retake lab assignments are available on demand.
- The Studium webpage has the full and updated practical information for the course.

Use of AI in this course

Why talk explicitly about generative AI?

- Generative AI is not a course topic in itself.
- It is an instrument that can assist modeling, coding, and exposition.
- Their use must be critical and documented.
- In scientific work, convenience never replaces correctness.
- Uppsala University also provides guidance for students on reflective and responsible use of generative AI.

Reference: Uppsala University,

<https://www.uu.se/en/students/it-for-students/generative-ai-and-studies>



Generative AI usage policy

Generative AI tools are:

- allowed and encouraged,
- treated as assistive tools, not authoritative sources.

All assignments should include the most relevant

- prompts used,
- generated outputs,
- corrections and validation steps.

Core principle for this course

Principle

Generative AI output is considered an unverified conjecture until validated.

Consequences:

- do not trust output blindly,
- check mathematics and code independently,
- keep a reproducible record of what was generated and how it was corrected.

What is an author?

At what point does AI assistance stop being assistance and start replacing authorship?



Source of image: Wikimedia Commons, Astrid Lindgren
(cropped).jpg.

What is an author?

Following the discussion in Figueras, *Authorship, Plagiarism, and Generative AI in Mathematical Research* (March 16, 2026, usingGenAI.pdf), authorship should be based on all of the following:

- ① substantial contribution to the conception, design, analysis, or interpretation,
- ② drafting the work or critically reviewing its intellectual content,
- ③ final approval of the version to be published,
- ④ accountability for the accuracy and integrity of the whole work.

Authorship and generative AI use

A useful question for this course is:

Guiding question

Am I retaining the authorship of the work if I do this?

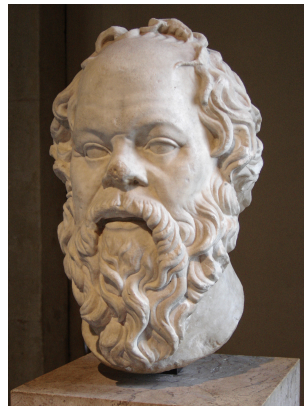
In particular:

- using generative AI for feedback, reformulation, or debugging can be legitimate,
- delegating originality-bearing text or code without critical contribution weakens authorship,
- writing a short prompt alone does not make one the author of the output.

Capabilities of generative AI in scientific computing

Useful tasks include:

- brainstorming modeling approaches (Socratic dialogue),
- drafting Python code,
- explaining algorithms,
- suggesting tests or alternative formulations,
- improving documentation and reproducibility.



Socrates, Louvre Museum

Source: Wikimedia Commons

Limitations of generative AI

Typical failure modes include:

- incorrect mathematical statements,
- fabricated references or unsupported claims,
- code that runs but does not solve the intended problem,
- hidden assumptions,
- poor treatment of edge cases and limiting regimes.

Prompt engineering for modeling

Better prompts usually specify:

- the system to be modeled,
- the assumptions and variables,
- the desired output format,
- the level of mathematical detail,
- the requested implementation language.

A good prompt reduces ambiguity; it does not eliminate the need for verification.

Code generation in Python

Generative AI tools can help produce:

- simulation prototypes,
- plotting scripts,
- parameter sweeps,
- simple numerical experiments.

But every script must still be checked for:

- correctness,
- numerical stability,
- consistency with the model assumptions.

If you don't understand how a script works, you cannot use it.



Programming image from Wikimedia Commons

File: ComputerProgrammer.jpg

When using generated mathematics or code, always test:

- sanity checks,
- invariants,
- limiting cases,
- consistency with definitions and units,
- reproducibility of the workflow.

Reproducibility and documentation

Good practice requires keeping track of:

- prompts,
- generated outputs,
- manual corrections,
- validation procedures,
- final code and conclusions.

This is part of scientific methodology, not an optional extra.

Objective: learn controlled use of generative AI.

Main tasks:

- prompt engineering for simulations,
- debugging generative-AI-generated code,
- critique and validation of models,
- comparison of workflows with and without generative AI support.

If you are interested you can start playing with it on your own. There are several free generative AI you could use: for example, *DeepSeek*.

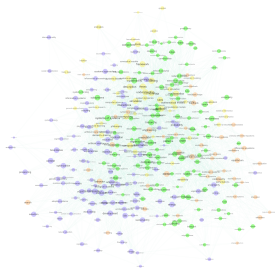
Complex Systems

Complex systems: definition and examples

A **complex system** [Wikipedia] is a system made of many interacting parts whose collective behavior is difficult to infer from the parts alone.

Typical features include nonlinearity, feedback, adaptation, and emergence across scales.

Examples include the climate, the brain, ecosystems, cities, and power grids.

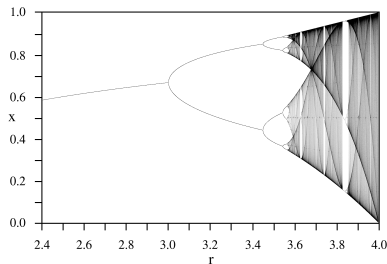


Source of image: Wikimedia Commons, Complex systems topic network (Sayama 2024).png.

Dynamical systems

A **dynamical system** [Wikipedia] is a system whose state evolves over time according to a fixed rule.

In this block we will study iterated maps, fixed points, stability, bifurcations, and chaos.

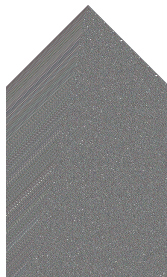


Source of image: Wikimedia Commons, LogisticMap_BifurcationDiagram.png.

Cellular automata

A **cellular automaton** [Wikipedia] is a discrete model made of a regular grid of cells, each in one of finitely many states, updated synchronously by a local rule.

This viewpoint lets us study locality, pattern formation, emergence, and universality.

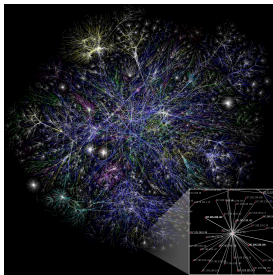


Source of image: Wikimedia Commons, Rule 30CAscaled.png.

Networks

Network science [Wikipedia] studies complex networks in which entities are represented by nodes and their interactions by links.

In this course we will use this language for graph structure, degree distributions, random and small-world models, and processes such as diffusion and epidemics.

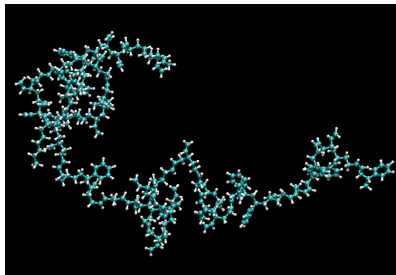


Source of image: Wikimedia Commons, Internet map 1024.jpg.

Particle systems

An **interacting particle system** [Wikipedia] is a stochastic process describing many components that evolve jointly through interaction.

In this course this will motivate particle-based modeling, interaction potentials, collective behavior, swarming, and flocking.



Source of image: Wikimedia Commons, Styrene-butadiene chain2.png.

Take-away message

- This course is about modeling complex systems with mathematics and computation.
- Generative AI can be a useful assistant, but never a substitute for understanding.
- The central standards are modeling assumptions, validation, and computational rigor.

References used in these slides

- J.-L. Figueras, *Authorship, Plagiarism, and Generative AI in Mathematical Research*, March 16, 2026.
Local file: biblio/usingGenAI.pdf.
- Wikipedia pages: *Complex system*, *Dynamical system*, *Cellular automaton*, *Network science*, *Interacting particle system*.
- Wikimedia Commons images: Complex systems topic network (Sayama 2024).png, LogisticMap_BifurcationDiagram.png, Rule 30CAscaled.png, Internet map 1024.jpg, Styrene-butadiene chain2.png, Socrates_Louvre.jpg, ComputerProgrammer.jpg.
- If a picture is not acknowledged, it was generated with generative AI.

Acknowledgment

These slides were prepared with the assistance of **OpenAI Codex / ChatGPT 5.4** as a drafting and editing tool.

All mathematical, pedagogical, and course-specific content remains under the responsibility of the lecturer.