UKACM & GACM AUTUMN SCHOOL 2025

Open-Source Codes for High-Performance Computing

Hands-on Session 4C & QUEENS

30.09.2025 and 01.10.2025

QUEENS

Sebastian Brandstäter[1], Georg Hammerl[2], Matthias Mayr[1,3],
Gil Robalo Rei[4], Ingo Scheider[2], Christoph P. Schmidt[4]

[1] Institute for Mathematics and Computer-Based Simulation | University of the Bundeswehr Munich
[2] Institute for Material Systems Modeling | Helmholtz-Zentrum Hereon
[3] Data Science & Computing Lab | University of the Bundeswehr Munich
[4] Institute for Computational Mechanics | Technical University of Munich

Update tutorial material

Virtual machine:

Pull the latest version from GitHub

1. Open a terminal in the virtual machine
2. Go to the folder with the tutorial material
3. Run:
   git pull --rebase origin

Docker container:

Pull the latest version from GitHub

1. Open a terminal on your machine
2. Go to the folder with the tutorial material
3. Run:
   git pull --rebase origin
4. (Re-)build the QUEENS Docker container.
   Follow the instructions here:
   https://github.com/mayrmt/UKACM_GACM_Tutorial_4C_QUEENS/blob/main/PREPARATION.md#queens-docker-container

See also the instructions on GitHub:
https://github.com/mayrmt/UKACM_GACM_Tutorial_4C_QUEENS/blob/main/PREPARATION.md

UKACM & GACM AUTUMN SCHOOL 2025
Open-Source Codes for High-Performance Computing
Hands-on Session 4C & QUEENS

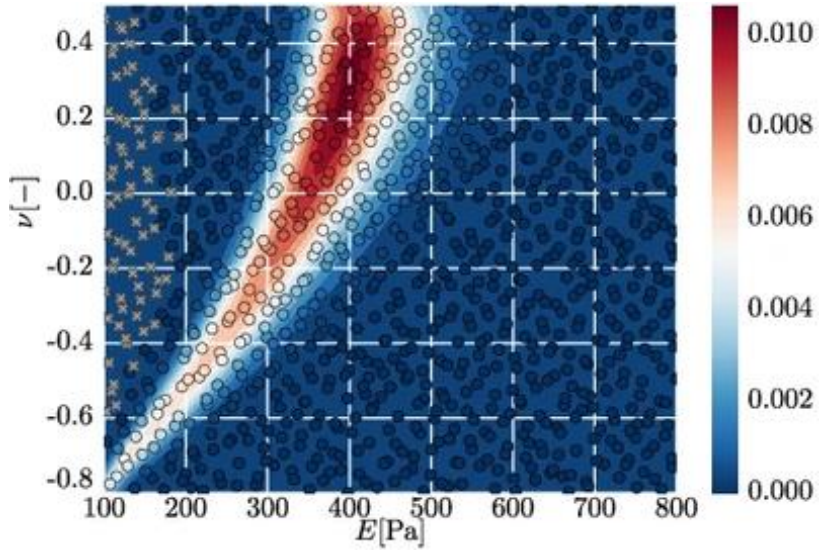
QUEENS

QUANTIFICATION OF UNCERTAIN EFFECTS IN ENGINEERING SYSTEMS

Sebastian Brandstäter[1], Maximilian Dinkel[2], Lea Häusel[2], Jonas Nitzler[2], Gil Robalo Rei[2]
[1] Institute for Mathematics and Computer-Based Simulation | University of the Bundeswehr Munich
[2] Institute for Computational Mechanics | Technical University of Munich

QUEENS

is a Python framework for **solver-independent multi-query analyses** of large-scale **computational models**.

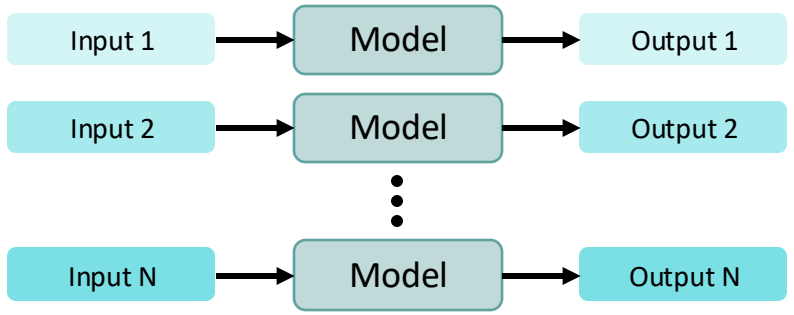


https://github.com/queens-py/queens

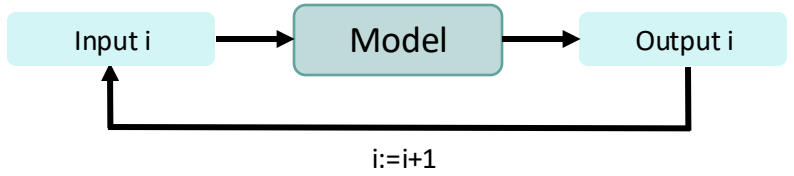
Multi-Query Analysis

Examples of multi-query scenarios:

- **Open loop:** parameter studies, uncertainty quantification



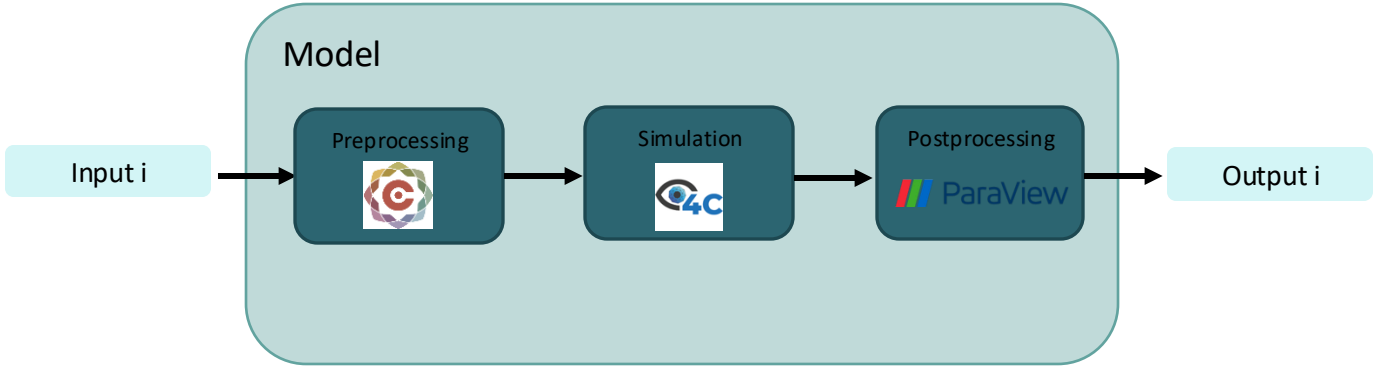
- **Closed loop:** optimization, (Bayesian) inverse problems



➢Evaluation of the same model at **many input locations**

Automation of model evaluations

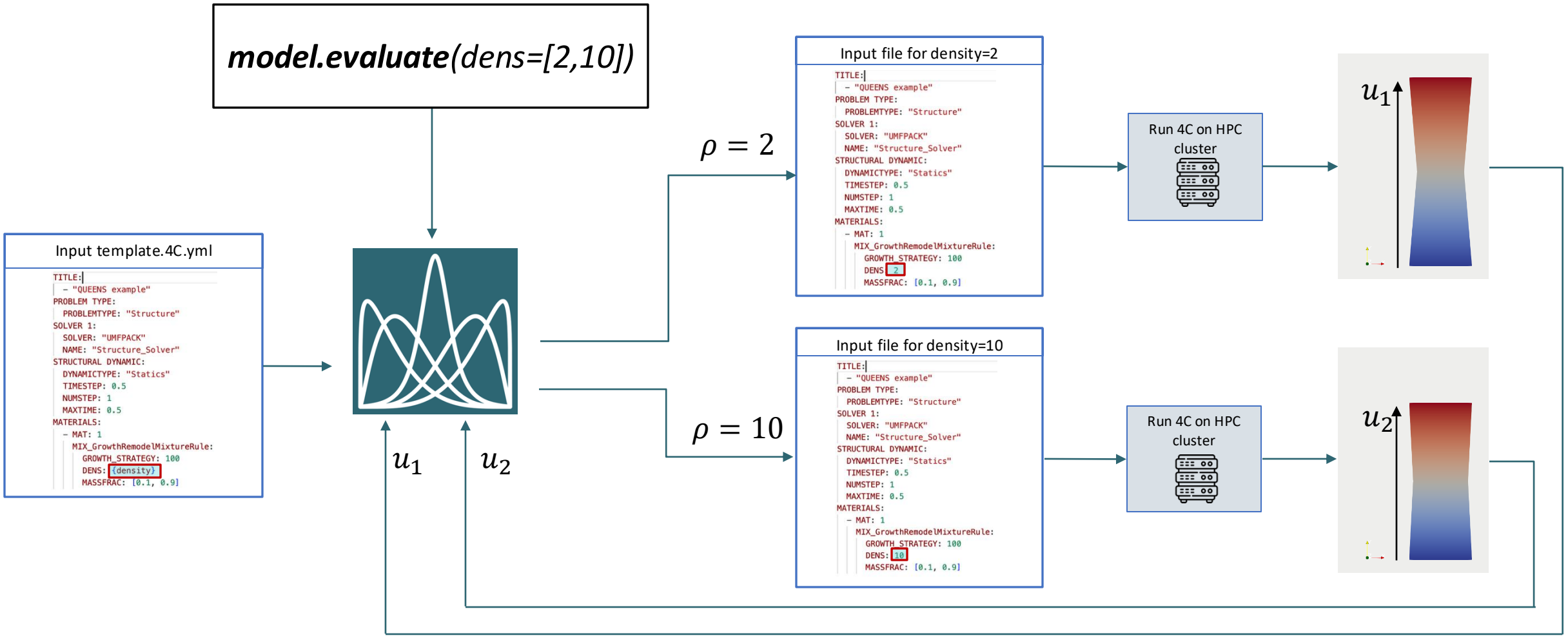
Each model evaluation involves numerous time-consuming manual sub-steps, e.g.,



➢QUEENS offers an abstract framework to **automate all of these sub-steps.**

Run 4C with a single line of code

Essentially, everything you need is:



Available analysis methods

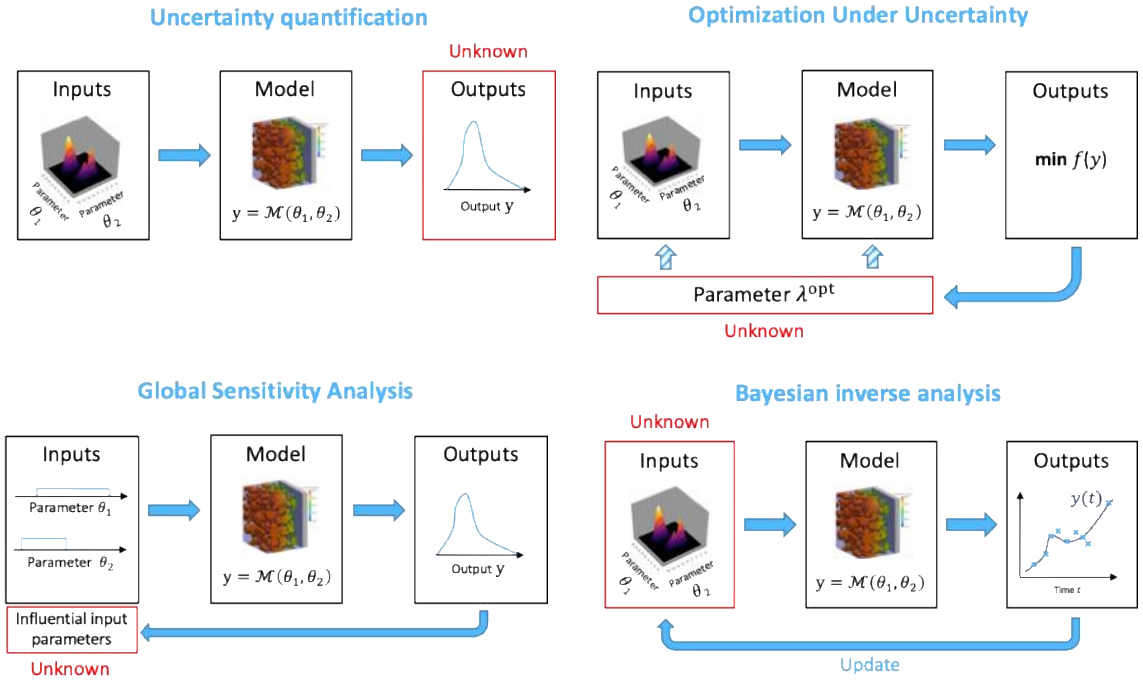
QUEENS offers a large collection of cutting-edge algorithms for deterministic and **probabilistic analyses**:

Methodological complexity

- parameter studies and identification
- sensitivity analysis
- surrogate modelling

⋮

- (multi-fidelity) uncertainty quantification
- Bayesian inverse analysis



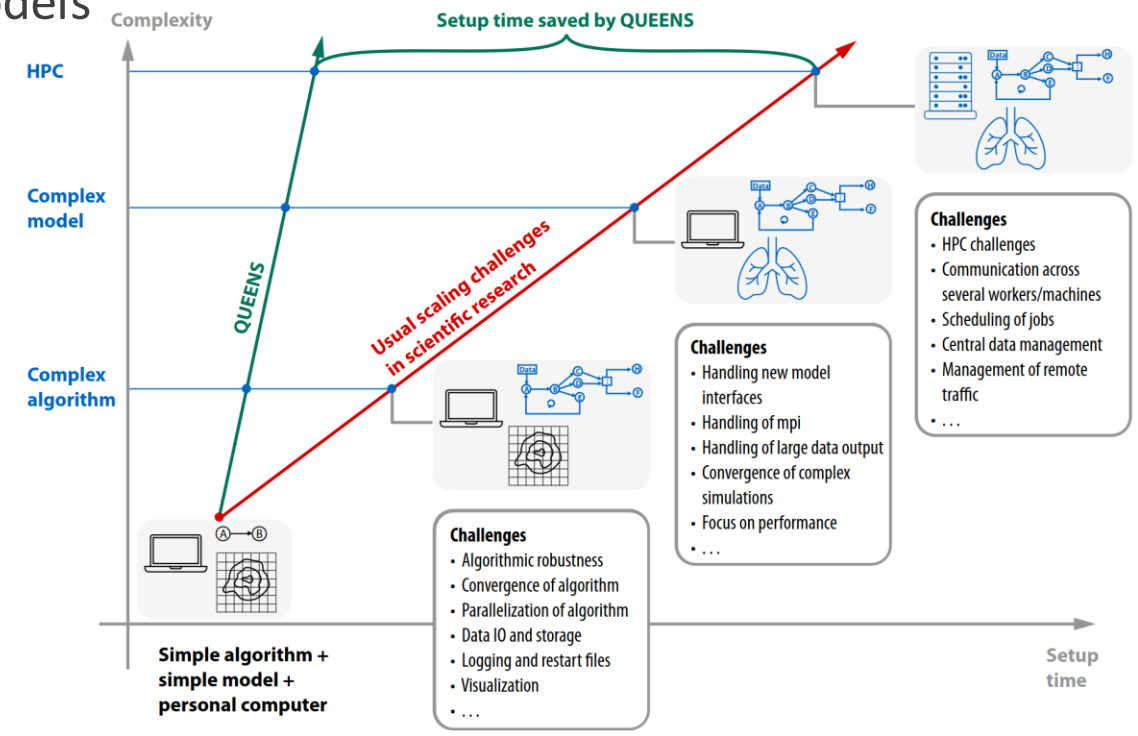
➢**A large variety of methods that are likely to be relevant to your research**

Large-scale computational models

QUEENS is designed for the analysis of computational models in which a single evaluation requires **significant computational resources.**

It provides a modular architecture for:

- parallel queries of large-scale computational models
- robust data, resource, and error management
- easy switching between analysis types
- smooth scaling from laptop to HPC cluster



Solver-independence



However, QUEENS is developed as a solver-independent platform.
It has also been interfaced with other solvers

- any Python package
- OpenFOAM
- Fenics
- deal.II-based codes
- and many others …






Community

Maintainer team

Sebastian Brandstäter

Maximilian Dinkel

Lea Häusel

Jonas Nitzler

Gil Robalo Rei

Contributors

Daniel Wolff

Silvia Hervás Raluy

Regina Bühler

Bishr Maradni

der Bundeswehr Universität München

You are welcome to join!
Connect with us on GitHub.

https://github.com/queens-py/queens

Workflow example

Experiment

Model

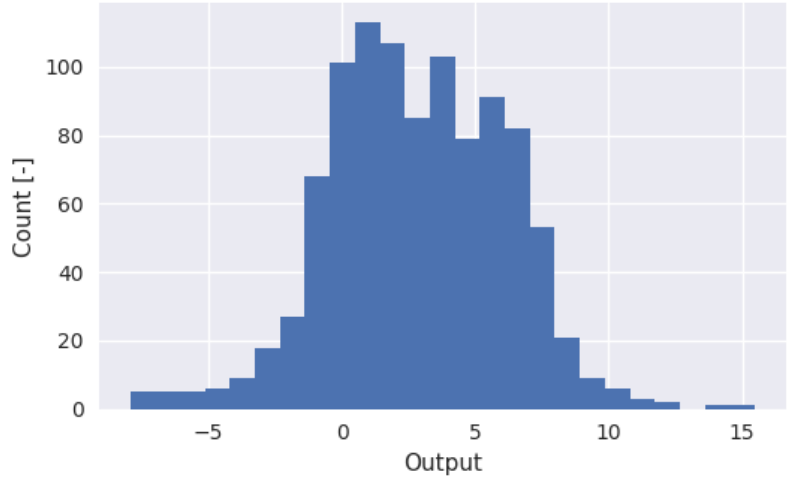
Parameters

Evaluation

Compute

Multi-query algortihm

```python
from queens.distributions import Beta, Normal, Uniform
from queens.drivers import Function
from queens.global_settings import GlobalSettings
from queens.iterators import MonteCarlo
from queens.main import run_iterator
from queens.models import Simulation
from queens.parameters import Parameters
from queens.schedulers import Local

if __name__ == "__main__":
    # Set up the global settings
    global_settings = GlobalSettings(experiment_name="monte_carlo_uq", output_dir=".")

    with global_settings:
        # Set up the uncertain parameters
        x1 = Uniform(lower_bound=-3.14, upper_bound=3.14)
        x2 = Normal(mean=0.0, covariance=1.0)
        x3 = Beta(lower_bound=-3.14, upper_bound=3.14, a=2.0, b=5.0)
        parameters = Parameters(x1=x1, x2=x2, x3=x3)

        # Set up the model
        driver = Function(parameters=parameters, function="ishigami90")
        scheduler = Local(
            experiment_name=global_settings.experiment_name, num_jobs=2, num_procs=4
        )
        model = Simulation(scheduler=scheduler, driver=driver)

        # Set up the algorithm
        iterator = MonteCarlo(
            model=model,
            parameters=parameters,
            global_settings=global_settings,
            seed=42,
            num_samples=1000,
            result_description={"write_results": True, "plot_results": True},
        )

        # Start QUEENS run
        run_iterator(iterator, global_settings=global_settings)
```



Theory and background

Since some of you might not be familiar with probabilistic approaches…

We will first cover some theory and background to:

- Motivate the use of probabilistic approaches
- Explain the key concepts behind the hands-on examples

Don’t worry—we’ll keep the math to a minimum.

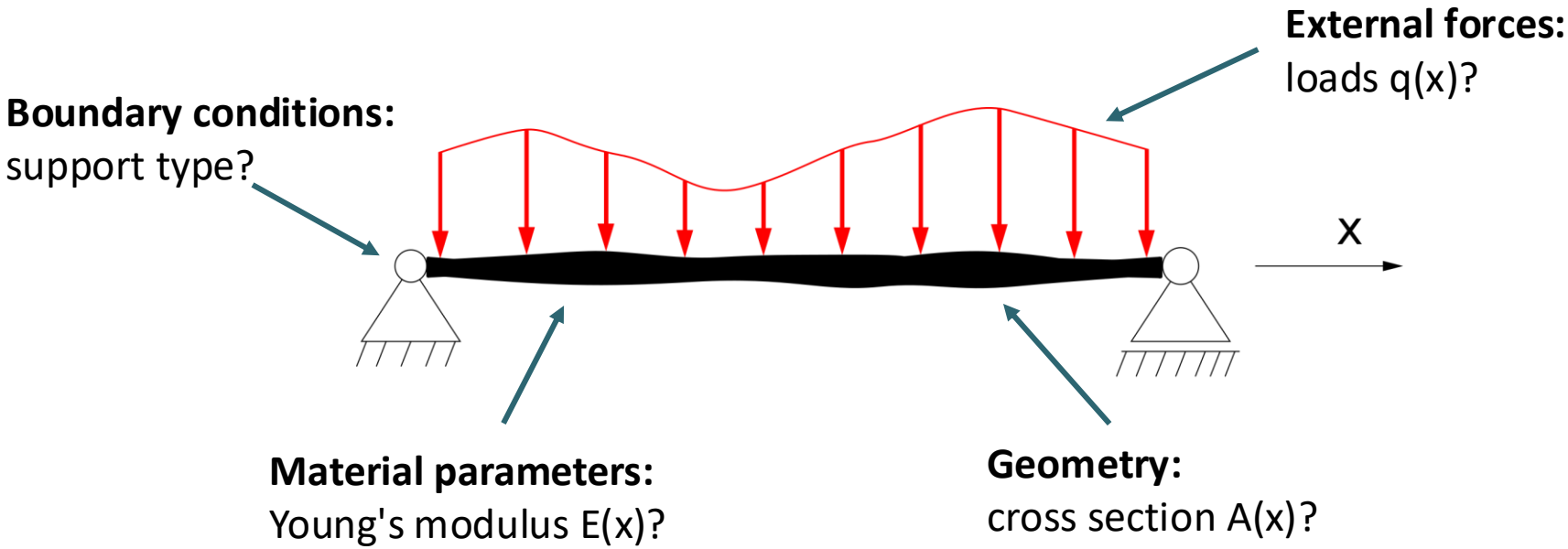
You’ll revisit these concepts during the tutorial, and most importantly, you’ll be able to follow the exercises even if some of the theory feels challenging.

What are uncertainties?

Uncertainty is the **lack of certainty** about a quantity because it has **more than one possible state**

- Can arise from **inherently random effects**

- Can arise from **lack of information/knowledge**



Why should I care about uncertainties?

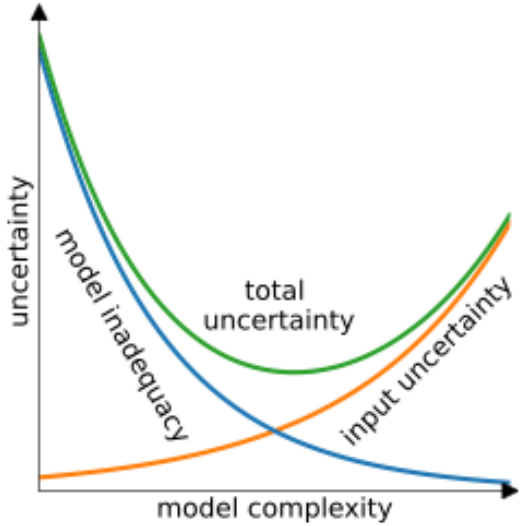
To make **meaningful predictions**, uncertainty needs to be incorporated:

Every model is only an **approximation of reality**

- Uncertainty through **model**

Even if a model is perfect, numerically **precise results can be meaningless** if model parameters are not known precisely

- Uncertainty through **parameters**



With uncertainties, we can **tell what we know**, but also **what we do not know.**

How to handle uncertainties?

How engineers used to handle uncertainties:

- Redo analysis or measurements,
  analyse mean and standard deviation
- Propagation of significant figures
- Propagation of tolerances
- Post process data (filters, interpolation, ...)
- Use of safety factors
- Playing around with input parameters

Probabilistic approach:

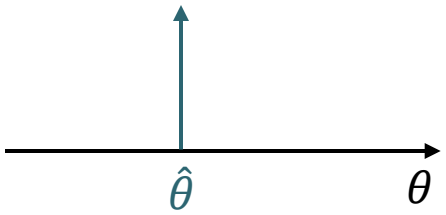
- Use **probability theory** to describe uncertainty
- Use **random variables** to model uncertain parameters

} Systematic approach

What are random variables?

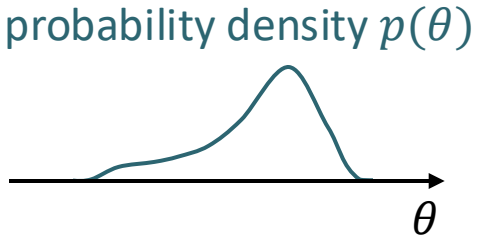
For example, θ is a model parameter:

Deterministic:



- θ assumes a **single value.**

Random variables:



- θ can assume **many values**, but with varying probability.

What are random fields?

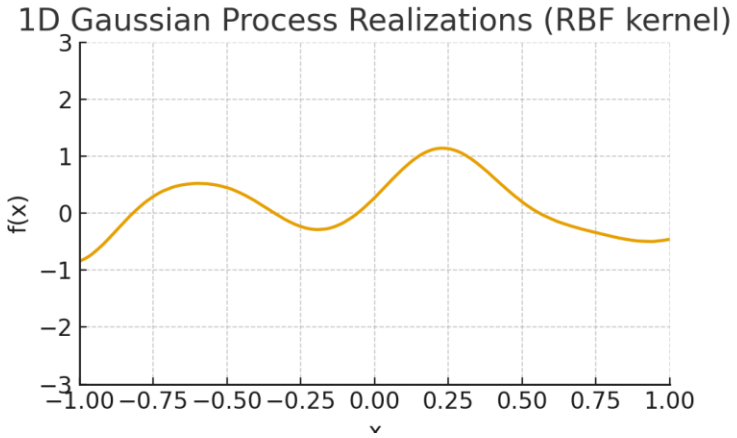
Random variables:

- A random variable is a single random number.
  θ can assume **many values**, but with varying probability.

$$\theta = -0.694$$

Random field:

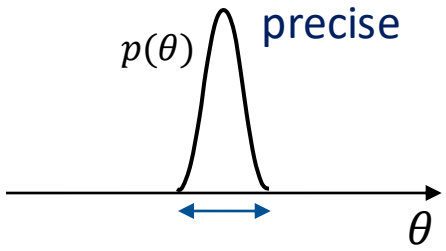
- A random field is the extension to functions: instead of a random number, you get a whole random function.
  Each realization of a random field is just one specific function.

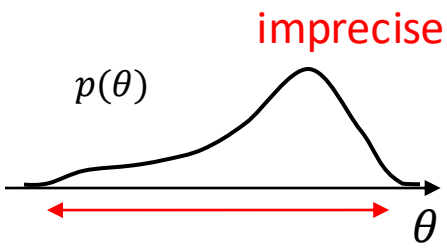


Advantages of probabilistic approaches

Information Quality

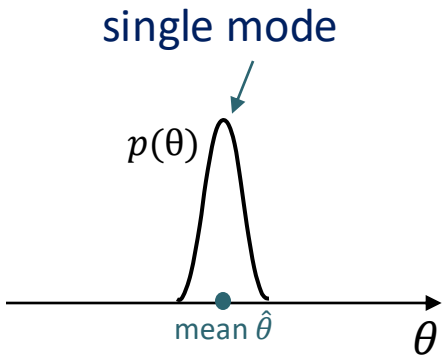
How precise can we measure?

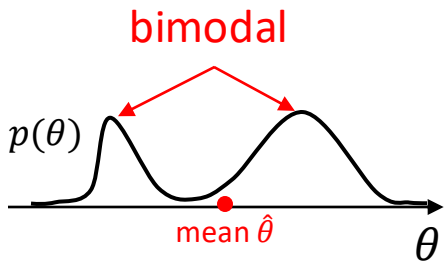




System Preferences

Are there multiple preferred configurations?

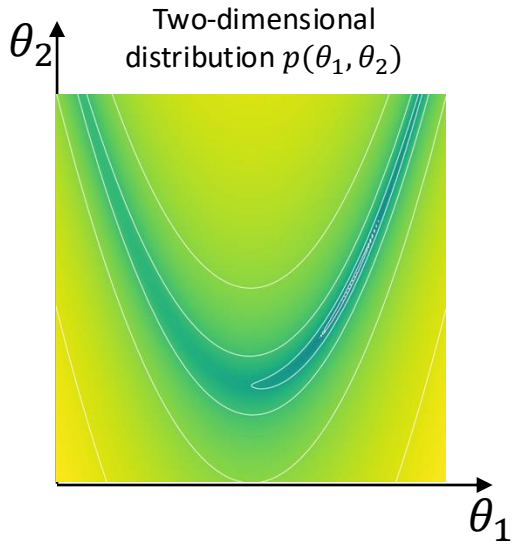




Interactions

Is there a correlation of parameters?

nonlinear interactions between θ_1 and θ_2



And many more…

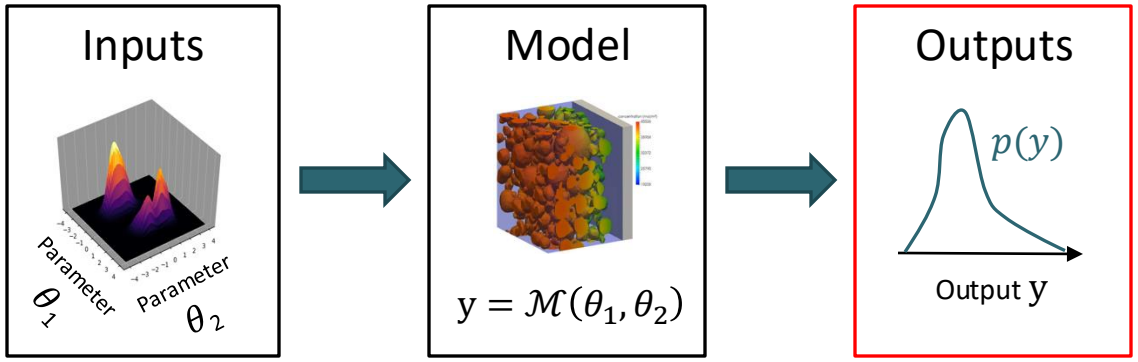
How to quantify uncertainties?

Uncertainty Propagation

Forward Uncertainty Quantification (UQ)

Quantification of uncertainties in the output

Forward UQ

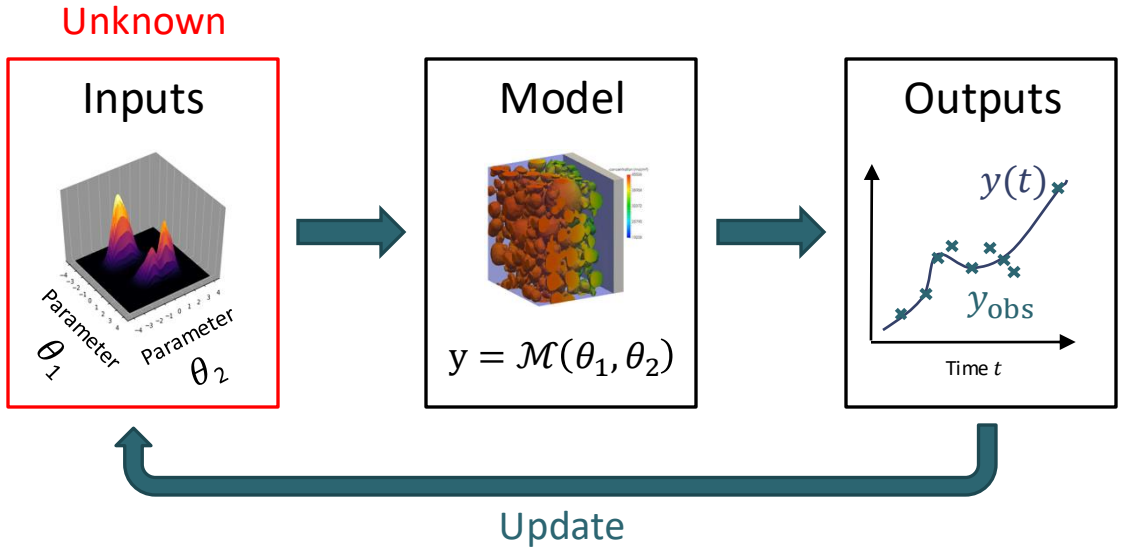


Inverse Analysis (IA)

Backward Uncertainty Quantification (UQ)

Quantification of uncertainties in the input

Backward UQ



Hands-on Sessions on ZOOM

09:30 - 11:00 QUEENS 1: From Grid studies to deterministic optimisation

11:00 - 11:15 Break

11:15 - 12:15 QUEENS 2: Uncertainty propagation and quantification

12:15 - 13:15 Lunch break

13:15 - 14:45 4C & QUEENS 1: Simulation analytics - Orchestrating 4C simulations with QUEENS

14:45 - 15:00 Break

15:00 - 16:30 4C & QUEENS 2: Quantifying uncertainty due to heterogeneous material fields

16:30 - 17:00 State-of-the-art research with QUEENS

Link: https://unibw.zoom-x.de/j/64722868182?pwd=V4bEWtP43aJy9NOx2TfkdPbwMuebqY.1
Meeting-ID: 647 2286 8182
Passcode: 409537