Overview of calibration features in OpenTURNS

Michaël Baudin [1]

[1]EDF R&D. 6, quai Watier, 78401, Chatou Cedex - France, michael.baudin@edf.fr

June 19th 2024, Palaiseau, France

Introduction

Calibration is the step B' of the generic methodology[1].

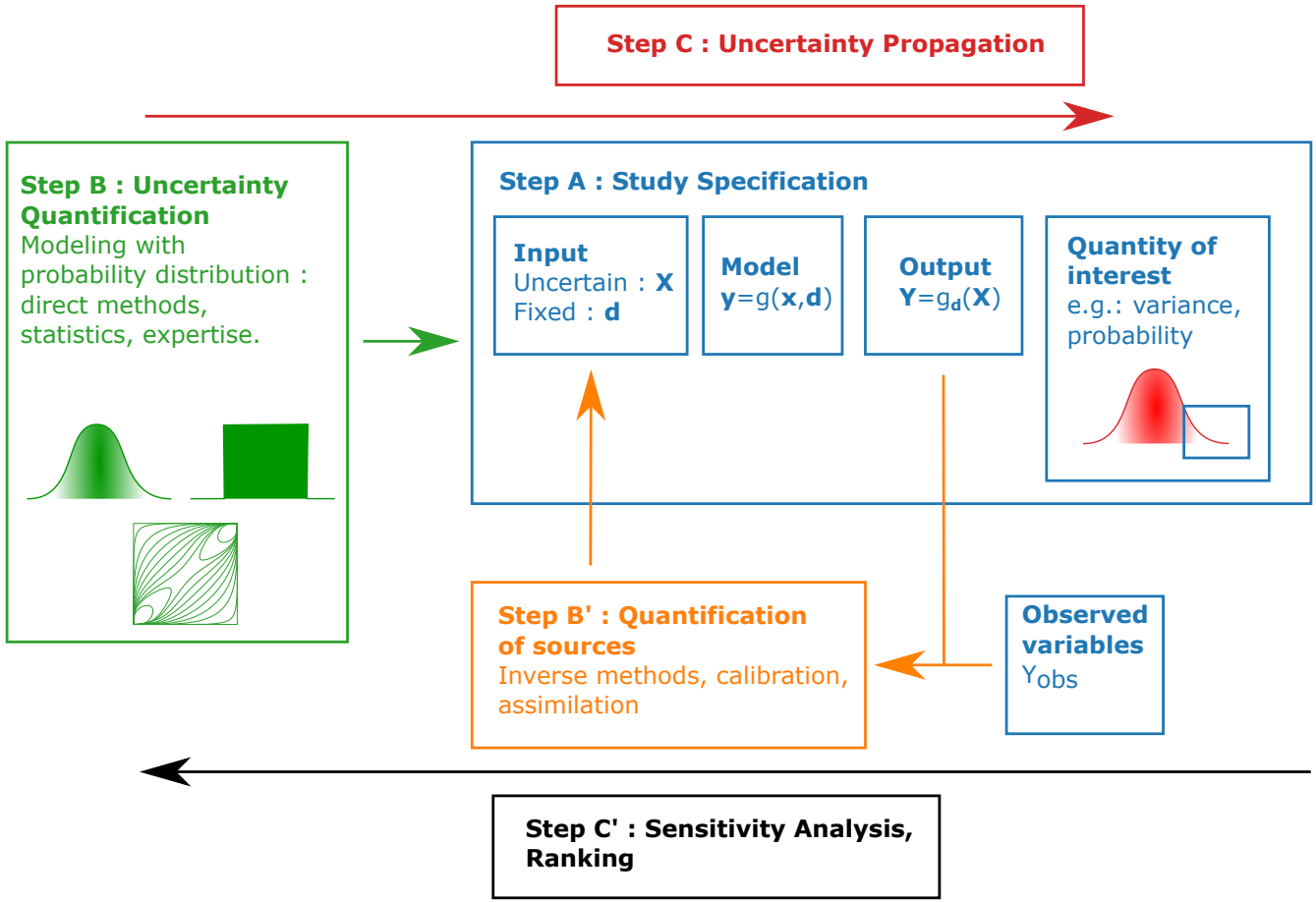

Figure: The step B' brings the observed predictions from the model closer to the observed outputs to calibrate the parameters.

[1] See[?, ?].

Introduction

We have:

- a dataset,
- a parametric model with unknown parameters.

We search for:

- parameter values,
- such that the predictions of the model are as close as possible to the data.

Since the dataset is random, we want the distribution of the parameters.
From there, we can compute confidence intervals of the parameters.

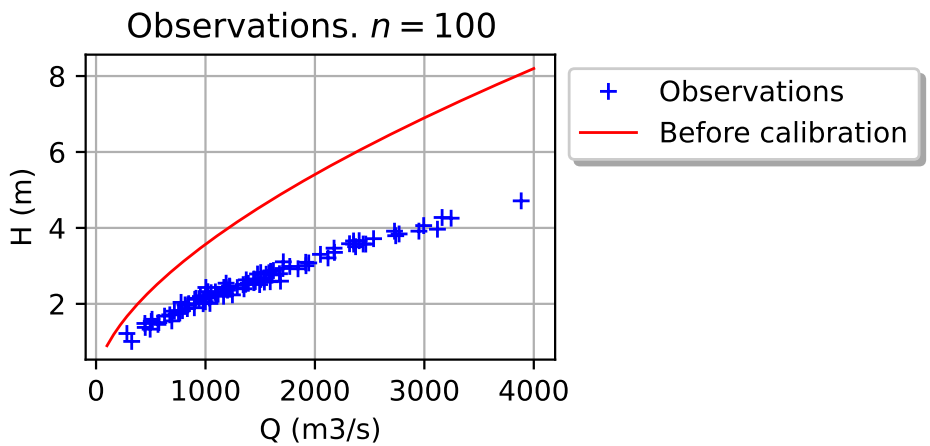

Figure: Observations compared to the predictions of a model.

Overview

In OpenTURNS, we have several calibration features:

- theory help pages
- API help pages
- examples.

There are two types of features :

- linear and non linear least squares, Gaussian linear and non linear calibration : `*Calibration` classes. These classes compute the **posterior distribution of the parameters**.
- Monte Carlo Markov Chain (MCMC) algorithms : `*MetropolisHastings`, etc. These classes **generate a sample from the posterior distribution of the parameters**.

The simplest example is Calibrate a parametric model: a quick-start guide to calibration
Here, we are going to review the Calibration of the flooding model

Conclusion

Other tools :

- Calibration methods are also available in Persalys : linear and non linear least squares, Gaussian linear and non linear calibration.

Perspectives:

- provide bounds to the optimization algorithms (return truncated normal distribution if necessary);
- unify the `ParametricFunction` in `*Calibration` and `*MetropolisHastings` classes (exchange the roles of x and θ);
- calibrate parametric functions with field output more easily;
- provide algorithms to automatically compute finite difference steps (not specific to calibration);
- provide the covariance matrix of the parameters as a diagonal matrix when possible;
- scale the parameters to calibrate (not specific to calibration);
- implement `CalibrationResult.isBayesian()` (see 2560);
- implement a `CalibrationResult` structure for M.-H. classes.

Références I