

"Benchmark" of integrated solutions for uncertainty quantification

Quantifying uncertainty makes it easier

14th of June 2024

B. Kerleguer (baptiste.kerleguer@cea.fr)

CEA, DAM, DIF, F-91297 ARPAJON, FRANCE



Problématique

The strategy for encouraging the use of uncertainty quantification at the DAM is based on three pillars:

- 1 Training in these methods → [internal DAM training](#).
- 2 Identifying "[experts](#)" → the role of my laboratory.
- 3 Highlight of a software for "simple" problems → Choise of [Persalys](#) in 2021.

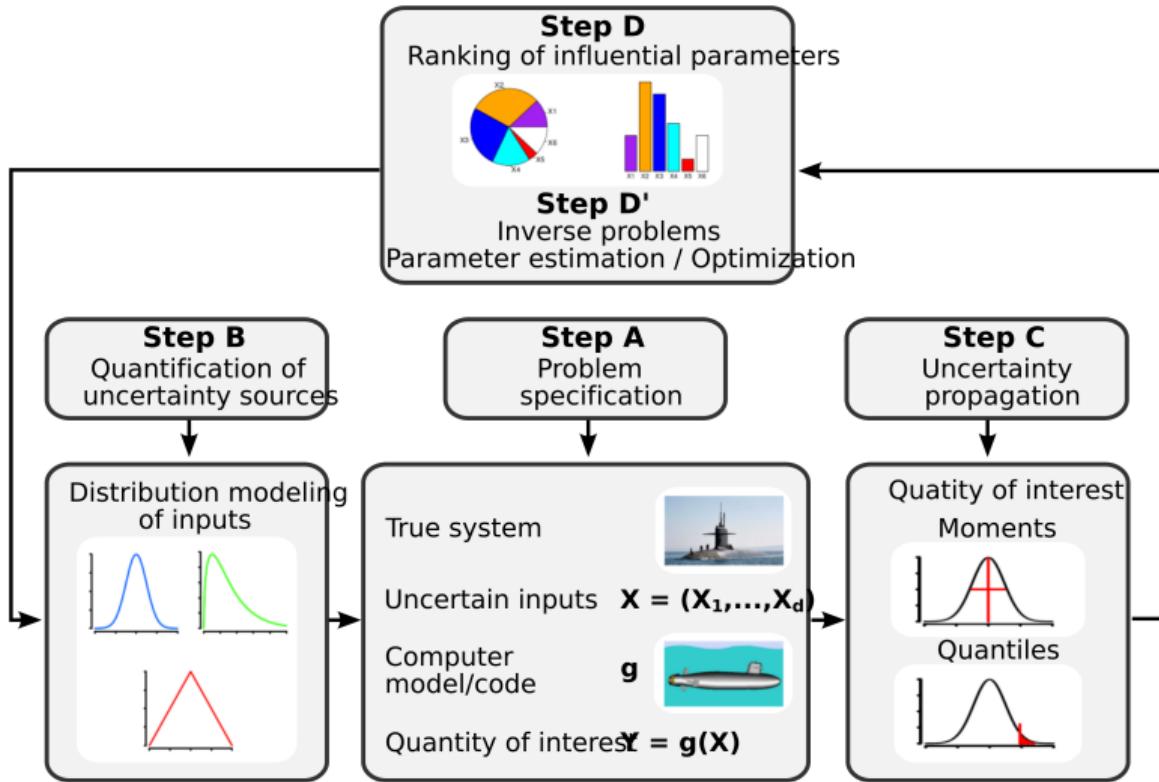
Question :

What software tools can be made available to DAM units for simple uncertainty quantification studies?

- This presentation takes the choice made in 2021 as a starting point and questions this choice on the basis of [3 new examples](#).



Uncertainty quantification method





The user

Needs

- Design of multiphysics systems.
- Interface with complex computer codes.

Knowledge of the use

- The basics of statistics.
- The basics of the ABCD method.
- A very advanced understanding of the problem to be solved.

Outline

Problem

Softwares

Performances



Outline

Problem

Softwares

Performances



Finding the best software for uncertainty quantification

An uncertainty quantification approach



Inputs / Outputs

- Inputs : 2 test cases.
 - a drone flying
 - Hydrogen's equation of state
- Outputs
 - Global sensitivity analysis
 - Surrogate models (not just linear regression)
 - Optimisation (option for multi-objective optimization)

Conditions to be evaluated

- Available on Linux and Windows, with documentation.
- Free and able to run the computation on the user's computer (to protect the user's data).
- (CEA condition) Installation should be possible offline (in less than 2 hours).



How the softwares are evaluated

Criteria or multi-objective

- User Interface
- Methods available
- Software's ergonomy

Important things we have assessed without drawing any conclusions

- Sensitivity indices computation
- Surrogate models computation
- Optimization performances

Outline

Problem

Softwares

Performances





Candidats

Name	Main Developpers	HCI	Based on	Reference
Lagun	Safran IFPEN, FR	Point and click		https://gitlab.com/drti/lagun
UQpy	SURG (M. Shields), US			uppyproject.readthedoc.io/en/latest/
UQLab	ETH Zurich, CH			www.uqlab.com/
PyApprox	Sandia NL, US			sandialabs.github.io/pyapprox/index.html
Cossan	Univ. of Liverpool, UK	Point and click		www.cossan.co.uk/software/open-cossan-e/
OpenTURNS	Airbus EDF IMACS			openturns.github.io/www/index.html
	ONERA Phymeca, FR			
Persalys	EDF Phymeca FR	Point and click		persalys.fr/
UQTk	Sandia NL, US			www.sandia.gov/uqtoolkit/
URANIE	CEA, FR			sourceforge.net/projets/uranie/
SmartUQ	SmartUQ, US	Point and click		www.smartuq.com
DAKOTA	Sandia NL, US	Point and		dakota.sandia.gov/

Softwares

Small presentation of the 3 softwares



The evaluated softwares

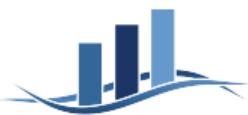
Dakota



DAKOTA

Explore and predict with confidence.

Persalys



Lagun

LAGUN



Sandia National Lab
1997



Phymeca, EDF
~ 2019



Safran, IFPEN
2019



Parameters

- V 14.1 (Not the most recent, but the most recent at the time of the study)

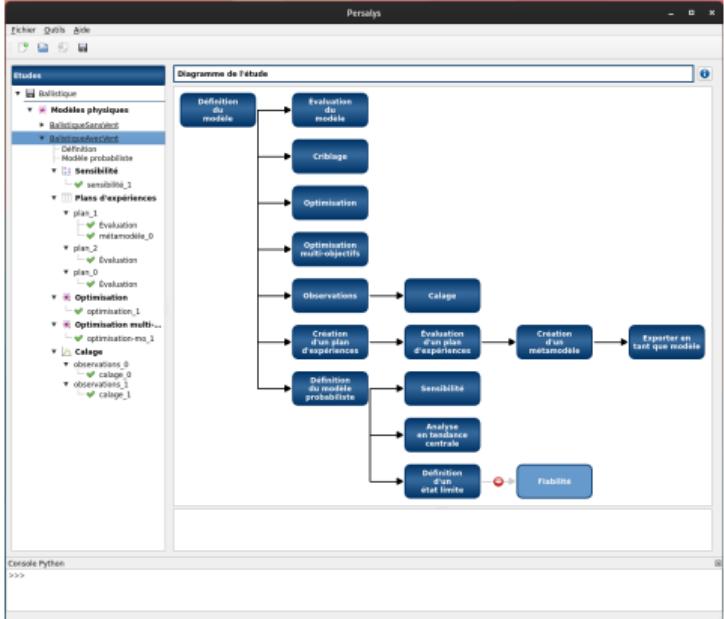
Features

- Design of Experiment
- Surrogate model
- Global sensitivity analysis
- Optimization (simple and multi-objective)
- Calibration

Problem

- Relatively slow for the RHEL installation that I use on a daily basis (no impact on my laptop opensource red hat).





Let see



LAGUN

Parameters

- V 1.0.0

Features

- Design of Experiment
- Surrogate model
- Global sensitivity analysis
- Optimization (simple and multi-objective)

Problem

- I was not able to use the simulator mode



LAGUN



Computer Code Exploration Platform LAGUN (V 3.0) | [About](#) | [Project](#) | [DOE](#) | [Problem Definition](#) | [Surrogate Model](#) | [Explore](#) | [Optimize](#) | [More](#) | [Help](#)

[Create](#)

LAGUN

LAGUN is a R/Shiny platform providing a user-friendly interface to methods and algorithms dedicated to the exploration and the analysis of datasets. Guided workflows are provided to help non-expert users to apply safely the proposed methodologies.

These tools are commonly used in the material uncertainty community (UMM) for example but are also widely applicable to experimental problems. The main functionalities are the following:

1. Optimal design of experiments: This tool allows to find the best parameters or the system which will generate the closest numerical simulations, settings of the experiments... so you can benefit for a better spatial repetition of the experiments.
2. Visual exploration tools: Once the numerical simulations and input/output responses are available, you can use it to perform insightful visual analyses and identify the main trends and the most influential parameters.
3. Going further with a surrogate model: And even more, you can use the database to write a script to infer a predictive relationship between the input/outputs and the outputs. This estimated relationship, the surrogate model, can help push forward the analysis with its ability to predict the responses for any new combination of the inputs. In particular it can be extensively used for uncertainty quantification, sensitivity analysis, deterministic optimizations, optimization under uncertainty (product and reliability based) or more intensive graphical studies.

For a general introduction, please see the following tutorial: [Introduction](#)

History

The first version of LAGUN was initiated at Institut Fréch (under a different name), the corporate research center of Eiffel. Its goal was to give an easy access to methods and algorithms to all civil engineers with a user-friendly interface. A collaboration was later launched with OpenTURNS in 2011 to share algorithms and developments in a common platform now named LAGUN ("Assistance" in French language).

The platform is organized in tabs, each one of them corresponding to a step above.
Click to expand the panels below to learn more through tutorials and test cases (plots >> details)

General Layout & Color Code

Prepare DOE

Problem Definition

Surrogate Model

Explore

Optimize

More

[CEA INSTITUTE](#)

Let see





Why I will not use DAKOTA

DAKOTA

Explore and predict with confidence.

- Too hard to install
- Graph are very poor
- Old school for french engineers

The screenshot displays two windows of the DAKOTA software:

- Parameter Study Window:** Shows a grid of 20 parameter sets. Each set consists of four input values (0.0, 0.25, 0.5, 0.75) and four output values (0.0000, 0.2500e-9, 0.5000, 0.7500e-9). The columns are labeled "Input_0000", "Input_0025", "Input_0050", and "Input_0075". The rows are labeled "Output_0000", "Output_0025", "Output_0050", and "Output_0075".
- Code Editor Window:** Displays a C++ code snippet for a parameter study. It includes sections for `#include`, `introduction`, `model`, `variables`, `interface`, and `frequency`. The `variables` section defines parameters like `constant_and_design` and `user_bounds`. The `interface` section specifies `oturns` drivers and `direct` access. The `frequency` section specifies `functions = 2`.



Test Cases

How we plan to use the softwares



1D drone ballistic trajectory

Newton's laws for drone ballistic

- Drone without engine with drag and leaf
- Simple very fast simulator



Target



1D drone ballistic trajectory

Technical details of the problem

- 4 inputs of the code + 3 parameters + 1 random variable
- Code in and
- Maximum 80 code's calls

Goals

- Global sensitivity analysis
- Calibration of the code
- Optimization of the parameters for maximum distance



Equation of State Hydrogen

Compute the quantum two-body problem

- Equation between ρ , T and Z , P , E , F of hydrogen.
- Very large dataset (90000 points)

Goals

- Global sensitivity analysis and data analysis
- Surrogate model

Reason of this example

- Test the limits of softwares
- Tested software on a well-known example

This example did not work for more than 1000 points





Equation of State Hydrogen

Compute the quantum two-body problem

- Equation between ρ , T and Z , P , E , F of hydrogen.
- Very large dataset (90000 points)

Goals

- Global sensitivity analysis and data analysis
- Surrogate model

Reason of this example

- Test the limits of softwares
- Tested software on a well-known example

This example did not work for more than 1000 points



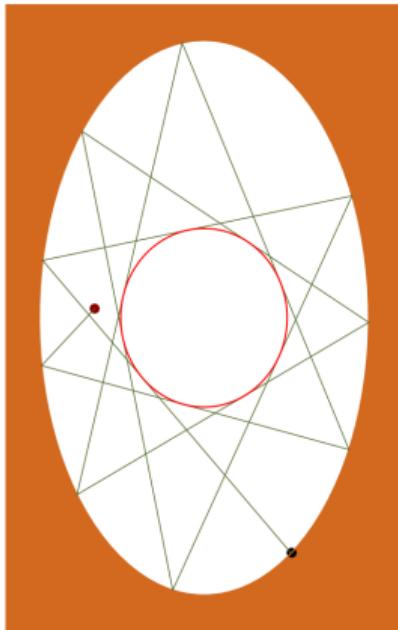
Elliptical billiard table

Technical details of the problem

- 4 inputs of the code + 2 parameters
- Code in
- Maximum 1000 code's calls

Goals

- Global sensitivity analysis
- Calibration of the code
- Optimization 2 inputs for surface optimisation



Outline

Problem

Softwares

Performances





Surrogate models- Linear regression

R^2 for the same training set

	Persalys	Lagun
Drone Y_1	0.80	0.76
Drone Y_2	0.93	0.92
Billard Y_0	$2.2 \cdot 10^{-16}$??
Billard Y_2	0.34	??

Persalys : Linear Model, order 1 without interactions

Lagun : Lasso Model

- The best linear model, order 2 with interactions, gives much better results (ex drone Y_1 : 0.95 and Y_2 : 0.99)



Surrogate models- Kriging

Q^2 for the same training set

	Persalys	Lagun
Drone Y_1	0.94	0.96
Drone Y_2	0.99	0.99
Billard Y_0	-0.00037	??
Billard Y_2	0.92	??

Persalys : Krgin Model with constant trend Matérn $\frac{5}{2}$ kernel

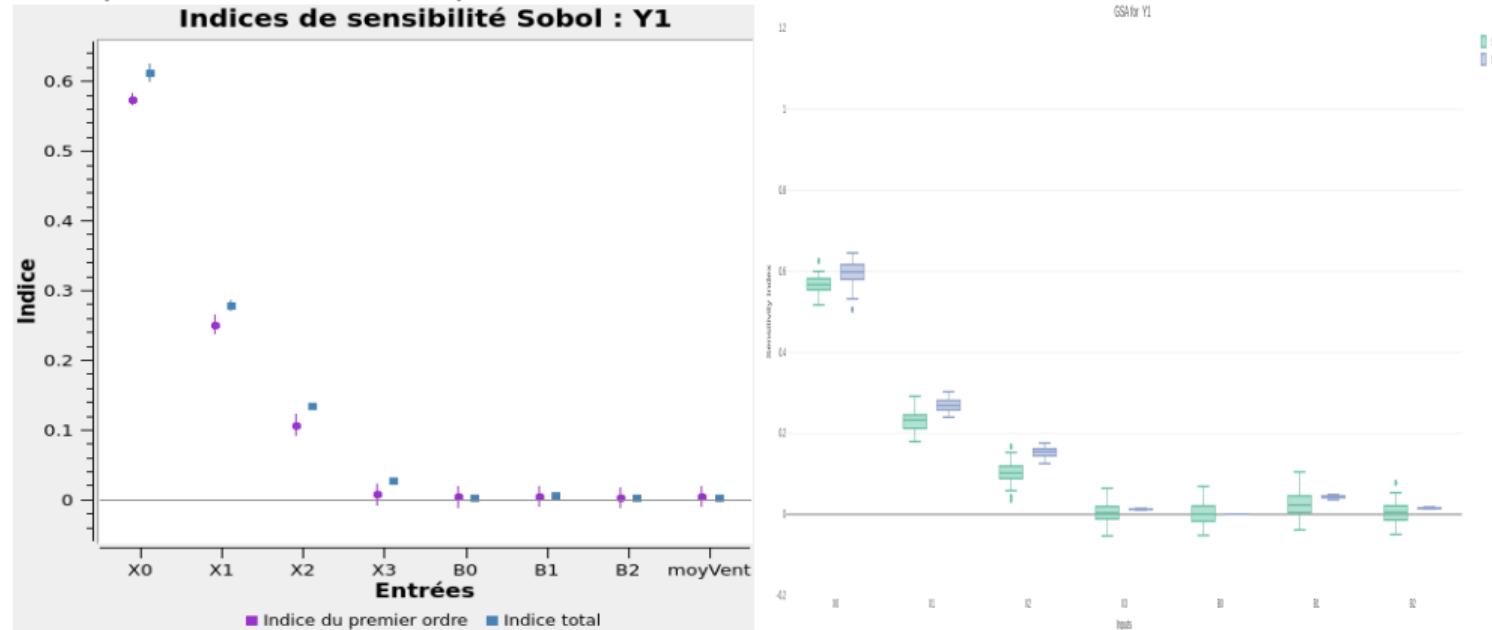
Lagun : Kriging Model with constant trend (kernel seems to be an optimization between 4 options)

- For Persalys I have test all possible model and takes the one that gives the best performances in Q^2 .



Global sensitivity analysis

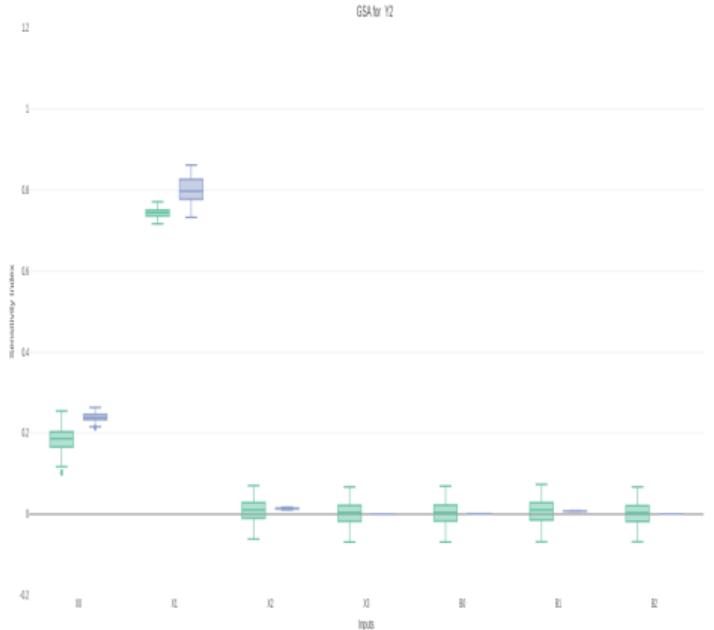
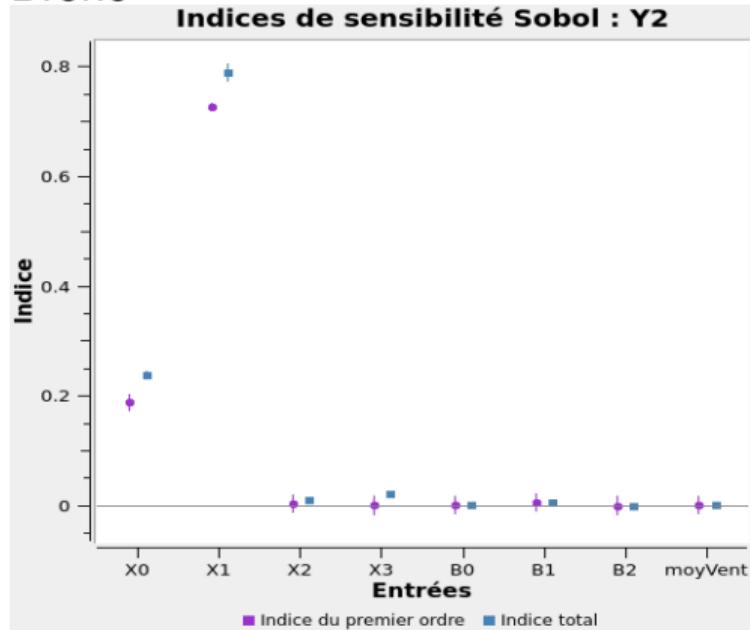
The GSA for both softwares solutions are based on Sobol indices on surrogates. Differences will therefore appear in the results due to the metamodel. In this study we did not attempt to evaluate the computation of the indices.





Global sensitivity analysis

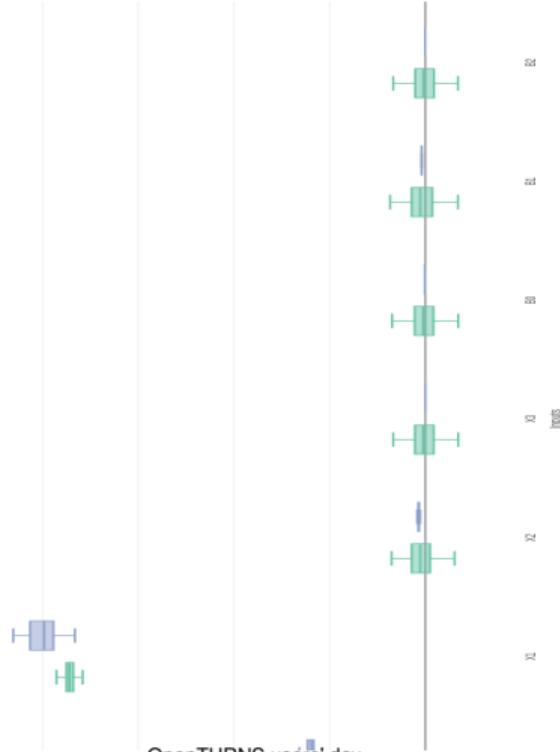
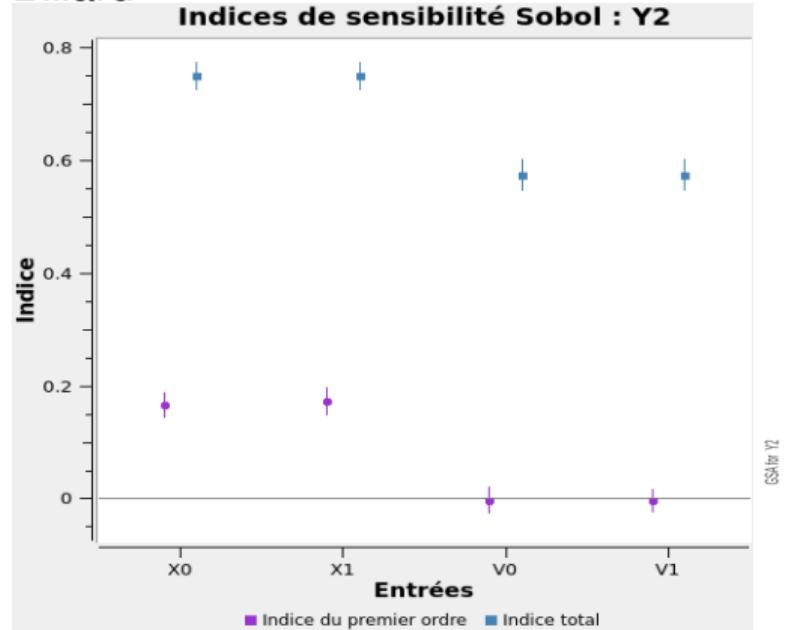
Drone





Global sensitivity analysis

Billard

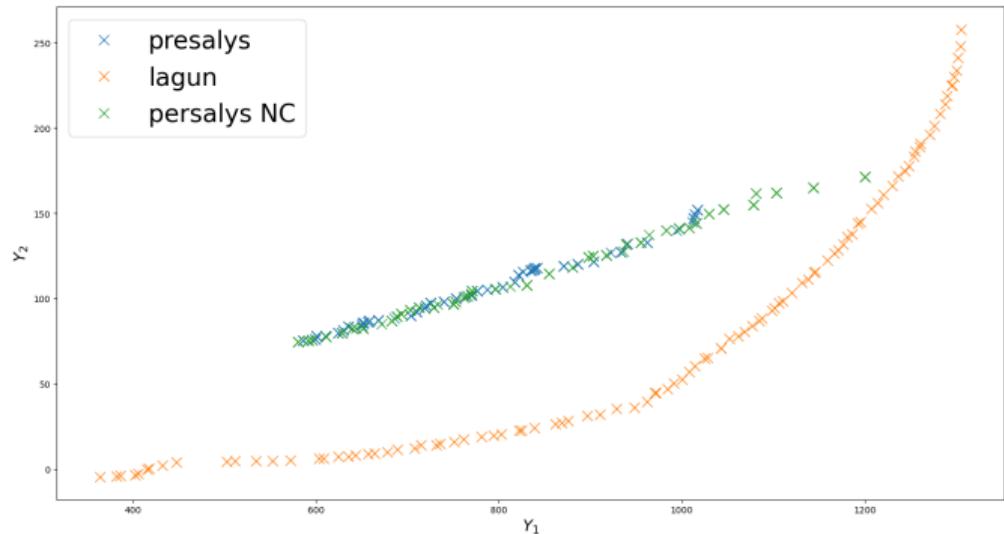




Optimization

The problem

- Maximize Y_1 , Minimize Y_2
- Constraints : $Y_2 > 60$ and $Y_2 < 152.4$ (This constraint is not available on Lagun)





Calibration



Drone

- Easy to set experimental data
- 4 different choices for calibration algorithm : linear and non-linear least squares and linear and non-linear Gaussian (3D-Var).
- Results :

Entrée	Valeur	Intervalle de confiance à 95%
B0	0.112615	[0.103109, 0.122121]
B1	0.0390973	[0.0279607, 0.0502339]
B2	9.79157	[9.77498, 9.80815]

$$\begin{array}{ccc} \beta_0 & \beta_1 & \beta_2 \\ 0.095 & 0.03 & 9.8050 \end{array}$$



Calibration



Billard

- Easy to set experimental data
- 4 differents choices for calibration algorithm : linear and non-linear least squares and linear and non-linear Gaussian (3D-Var).
- Results :

Entrée	Valeur	Intervalle de confiance à 95%
A0	2.97425	[2.97363, 2.97478]
A1	2.0008	[2.00077, 2.00083]

$$\begin{matrix} A_1 \\ 3 \end{matrix} \quad \begin{matrix} A_2 \\ 2 \end{matrix}$$



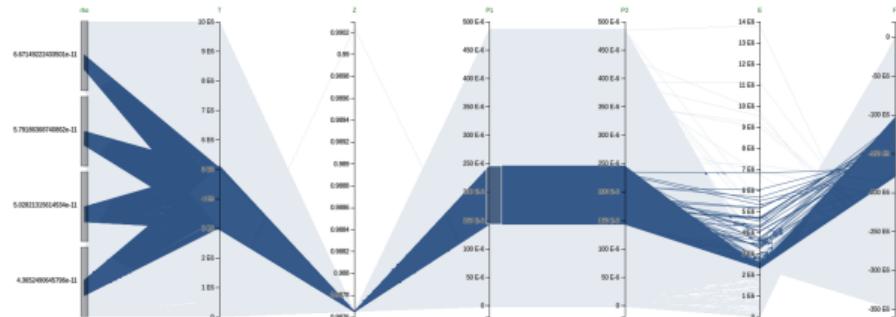
The little details that makes all the differences

LAGUN

■ Surrogate model combination

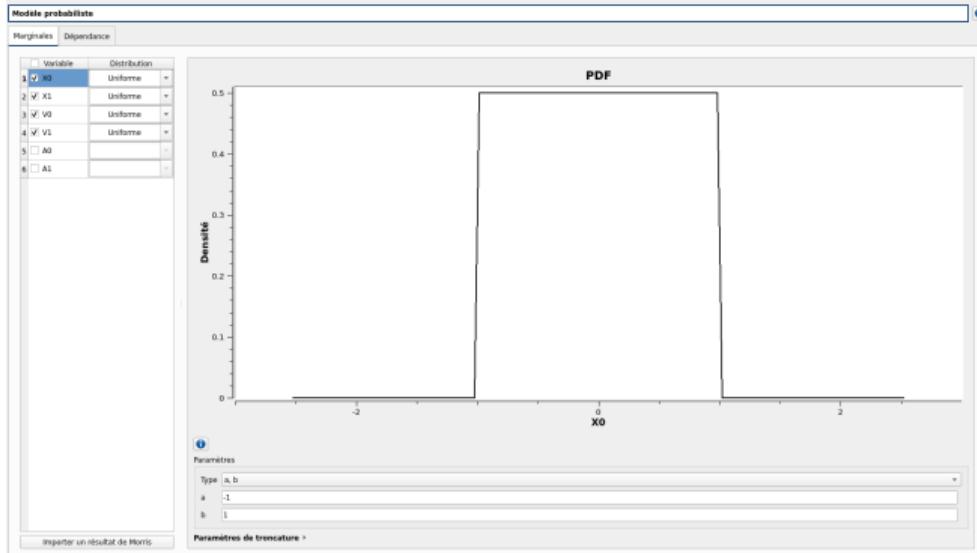
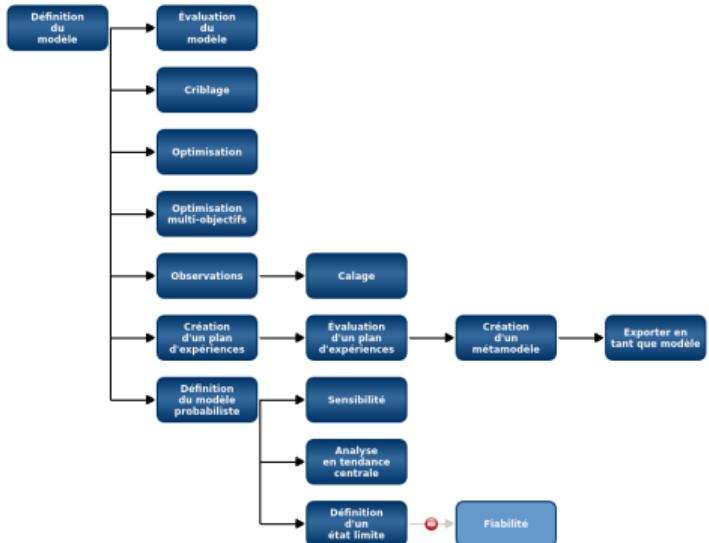
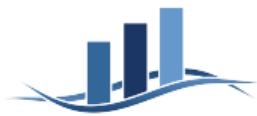
Type	Var	Trend	Z	P1	P2	E	F
BestQ2			0.7789	0.986	0.986	1	1
Lasso			0.7784	0.986	0.986	1	1
Acossol1			0.9791	0.9828	0.9828		
Acossol2	All						
Acossol2	Acossol1						
Kriging	All	Constant	0.7789	0.9796	0.9798	1	1

■ Parallel plot





The little details that makes all the differences





Conclusion

Powerful tools for speed and efficiency

- Complete all the steps in the ABCD method faster than we can in R/python
- Very, Very fast compared to simulations for most of the industrial cases.
- Advanced methods that are easy to use

There are still a few points that raise questions

- Tools vulnerable to defects in the basic software bricks. For example, a package required by Lagun is archived in CRAN because it no longer works on the new version of R?
- Some parts (Kriging Lagun and GSA all) are still black boxes for me, although I wanted to find out what was going on.

My use

- I use them to save time in my engineering studies.
- I give them to physicists and engineers to facilitate their UQ.



Thank you for your attention