Benchmark of integrated solutions for uncertainty quantification

Persalys users' day

8th of November 2024

B. Kerleguer (baptiste.kerleguer@cea.fr)
CEA, DAM, DIF, F-91297 Arpajon, France

Context

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

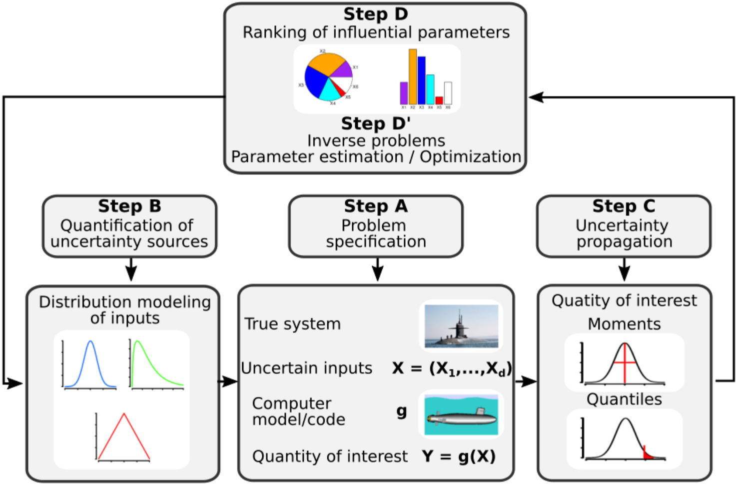
1. Training in these methods $\rightarrow$ internal DAM training.
2. Identifying "experts" $\rightarrow$ the role of my laboratory.
3. Using of a software for common problems $\rightarrow$ Choise of Persalys in 2021.

Question :

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

- This presentation takes the choice made in 2021 as a starting point
- In 2024, Actualization of the performance on 2 examples.

Uncertainty quantification schema



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

Software products

Performances

Outline

Problem

Software products

Performances

Finding the most suitable software for UQ for CEA DAM

An uncertainty quantification approach

Inputs / Outputs

- Inputs : 2 test cases.
  - A drone flying
  - Hydrogen's equation of state
- Outputs
  - Global sensity 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

- User Interface
- Methods available
- Software’s ergonomy

Important points

- Sensitivity indices computation time
- Surrogate models computation time and performance
- Optimization adaptive performances

Outline

Problem

Software products

Performances

Candidates

Name	Main Developers	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 ONERA Phymeca, FR			openturns.github.io/www/index.html
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/

Software products

Small presentation of the 3 softwares

The evaluated softwares

Dakota	Persalys	Lagun
Sandia National Lab 1997	Phymeca, EDF $\sim$ 2019	Safran, IFPEN 2019

Conclusion on Dakota

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

Persalys

Parameters

- V 16.1

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).

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

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 leaft
- Simple very fast simulator

1D drone ballistic trajectory

Technical details of the problem

- 4 code inputs + 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

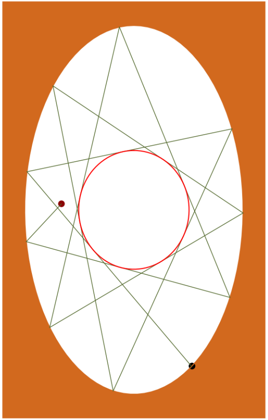
Elliptical billiard table

Technical details of the problem

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

Goals

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



Outline

Problem

Software products

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	2.210^{-16}	0.06253
Billard Y_2	0.34	-0.00178

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

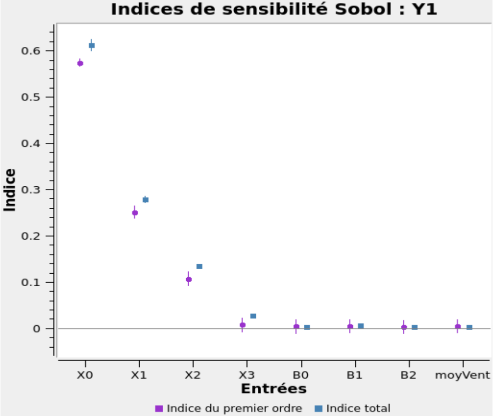
	Persalys	Lagun
Drone Y_1	0.94	0.96
Drone Y_2	0.99	0.99
Billard Y_0	-0.00037	0.014
Billard Y_2	0.92	0.97

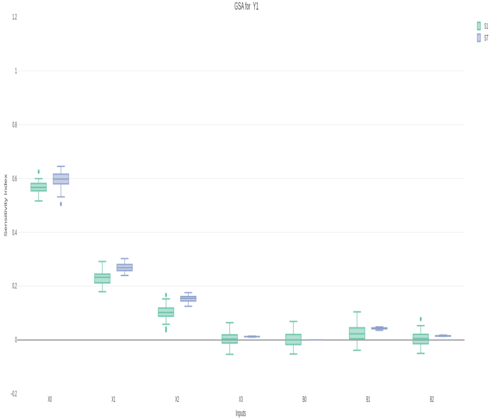
Persalys : Krigin Model with constant trend Matèrn $\frac{5}{2}$ kernel
Lagun : Kriging Model with constant trend (kernel seems to be an optimization between 4 options)

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

Global sensitivity analysis

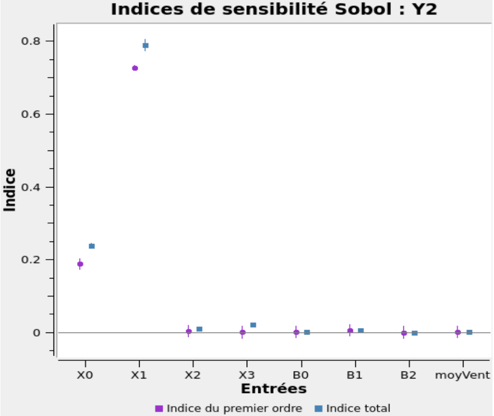
The GSA for both softwares solutions are based on Sobol indices on surrogates.
Differences will therefore appear in the results due to the metamodel. In this study we did not attempt to evaluate the computation of the indices.

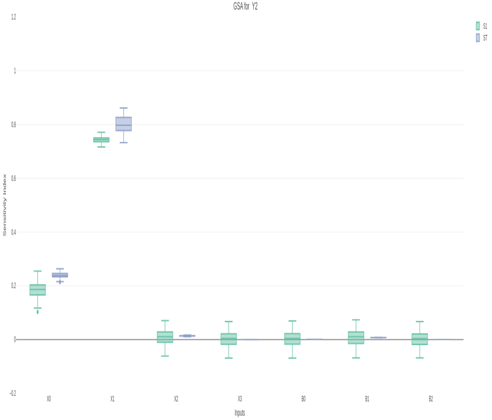




Global sensitivity analysis

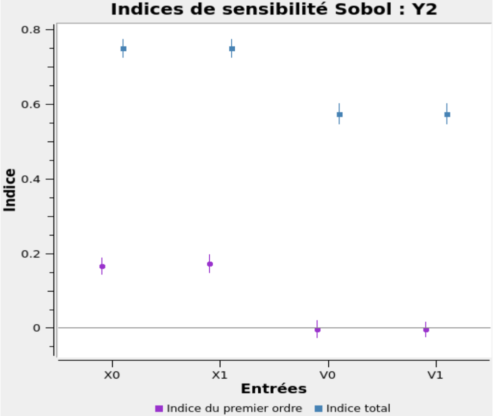
Drone

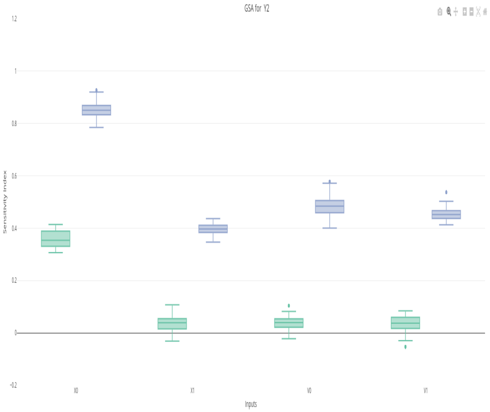




Global sensitivity analysis

Billard

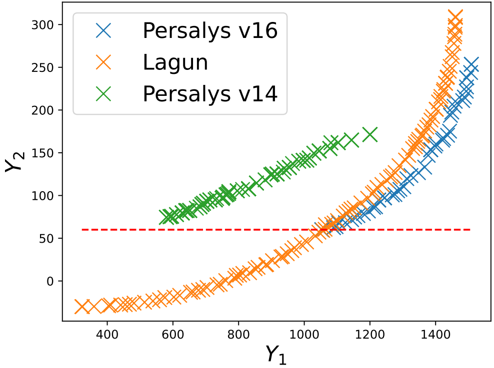




Optimization

Pareto front

- Maximize Y_1, Minimize Y_2
- Constraints : $Y_2 > 60$



Calibration

Drone

- 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%
B0	0.112615	[0.103109, 0.122121]
B1	0.0390973	[0.0279607, 0.0502339]
B2	9.79157	[9.77498, 9.80815]

β_0	β_1	β_2
0.095	0.03	9.8050

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]

A_1	A_2
3	2

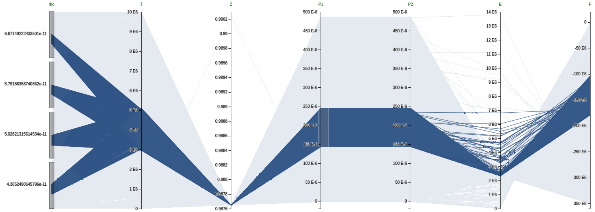
The little details that makes all the differences

LAGUN

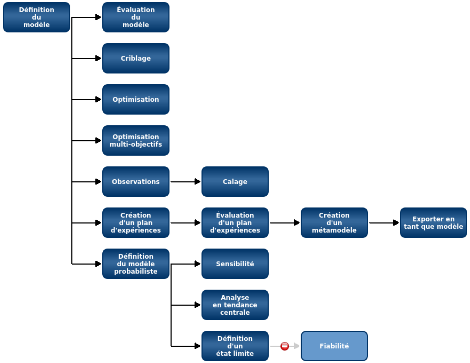
- Surrogate model combination

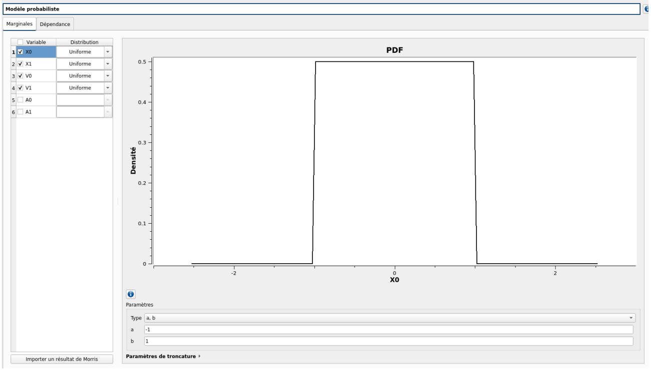
Type	Var	Trend	Z	P1	P2	E	F
BestQ2			**0.7789**	**0.986**	**0.986**	**1**	**1**
Lasso			0.1263	0.986	0.986	1	1
Acosso1			0.3731	0.9828	0.9828		
Acosso2	All						
Acosso2	Acosso1						
Kriging	All	Constant	0.7789	0.9796	0.9798	1	1

- 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 point 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?

Usage we envisage

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

Thank you for your attention