

# The graphical user interface of OpenTURNS, a UQ software in simulation

Michaël Baudin <sup>1</sup>   Anne Dutfoy <sup>1</sup>   Anthony Geay <sup>1</sup>   Anne-Laure Popelin <sup>1</sup>   Aurélie Ladier <sup>2</sup>   Julien Schueller <sup>2</sup>   Thierry Yalamas <sup>2</sup>

<sup>1</sup>EDF R&D. 6, quai Watier, 78401, Chatou Cedex - France, michael.baudin@edf.fr

<sup>2</sup>Phimeca Engineering. 18/20 boulevard de Reuilly, 75012 Paris - France, yalamas@phimeca.com

15 June 2017, UNCECOMP 2017, Rhodes, Greece



# Contents

Introduction

Demo

Background

What's next ?

Extra slides

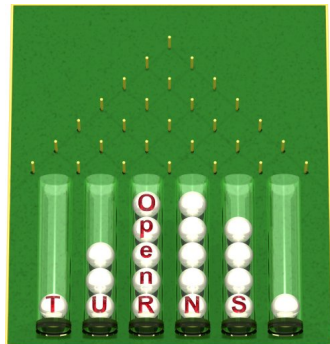
Demo backup

# OpenTURNS

- ▶ Uncertainty quantification, uncertainty propagation, sensitivity analysis and metamodeling
- ▶ Partners : EDF, Phiméca, Airbus, IMACS
- ▶ [www.openturns.org](http://www.openturns.org)
- ▶ Licence LGPL
- ▶ Linux, Windows

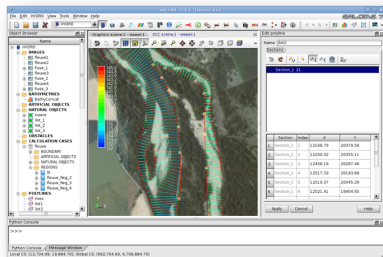
Programming:

- ▶ Python module
- ▶ C++ Library



# SALOME

- ▶ Integration platform for pre and post processing, and 2D/3D numerical simulation
- ▶ Features : geometry, mesh, distributed computing
- ▶ Visualization, data assimilation, uncertainty treatment
- ▶ Partners : EDF, CEA, Open Cascade
- ▶ Licence : LGPL
- ▶ Linux, Windows
- ▶ [www.salome-platform.org](http://www.salome-platform.org)



# The graphical user interface of OpenTURNS

- ▶ Main goal : provide a graphical interface of OpenTURNS in SALOME
- ▶ Features
  - ▶ Uncertainty quantification (distribution fitting), central tendency, sensitivity analysis, probability estimate, meta-modeling
  - ▶ Generic (not dedicated to a specific application)
  - ▶ GUI language : English, French
- ▶ Partners : EDF, Phiméca
- ▶ Licence : LGPL
- ▶ Schedule :
  - ▶ Since summer 2016, one EDF release per year
  - ▶ On the internet : 2018

# GUI : the demo

Demo time.

## GUI : outline

- ▶ From scratch : 3 inputs, 2 outputs, sum, central dispersion study with default parameters
- ▶ Open axialStressedBeam-python.xml : central dispersion with sample size 1000, Threshold  $P(G < 0)$  with  $CV=0.05$
- ▶ Import crue-4vars-analytique.py : S.A. with sample size 1000, sort by size

## UQ, the easy way

Main goal : make UQ easy to use

- ▶ classical user-friendly algorithms with a state-of-the-art implementation,
- ▶ default parameters of the algorithms whenever possible,
- ▶ an easy access to the HPC resources,
- ▶ an automated connection to the computer code.

Produce standard results :

- ▶ numerical results e.g. tables,
- ▶ classical graphics.



# Overview (1/2)

Inputs from the user :

- ▶ Physical model : symbolic, Python code or SALOME component
- ▶ Probabilistic model : joint probability distribution function of the input.

Then :

- ▶ Central dispersion: estimates the central dispersion of the output  $Y$  (e.g. mean).
- ▶ Threshold probability: estimates the probability that the output exceeds a given threshold  $S$ .
- ▶ Sensitivity analysis: estimates the importance of the inputs to the variability of the output.

## Overview (2/2)

Probabilistic modeling :

- ▶ Distribution fitting from a sample
- ▶ Dependence modeling (Gaussian copula)

Meta-modeling :

- ▶ Polynomial chaos (full or sparse)
- ▶ Kriging

# Fields

## Field example :

- ▶ Input : 4 independent random variables
- ▶ Output : height of the river Garonne on a 100 km segment
- ▶ Computer code : TELEM2D
- ▶ Quantity of interest : pointwise average over 70 000 random simulations

## Roadmap :

- ▶ Now : massive Python/OpenTURNS scripting
- ▶ 2017-2018 : in the gui



# The end

Thanks !

Questions ?

# Interactive uncertainty visualization with Paraview



# Methodology



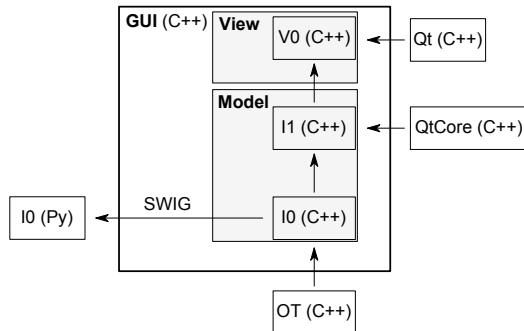
# Software architecture

Two entry points:

- ▶ interactive,
- ▶ Python.

Advantages of the Python programming of the GUI:

- ▶ unit tests,
- ▶ going beyond the GUI



# Symbolic physical model

The screenshot shows the OTGui application window. On the left is a tree view under 'Crue Analytique' containing 'physicalModel\_0', which is further divided into 'Deterministic study', 'Probabilistic study', and 'Designs of experiment'. The main area is split into 'Inputs' and 'Outputs' sections.

**Inputs Table:**

	Name	Description	Value
1	Q	Débit (m <sup>3</sup> /s)	0
2	Ks	Strickler (m <sup>1/3</sup> /s)	30
3	Zv	Côte aval (m)	50
4	Zm	Côte amont (m)	55
5	Hd	Hauteur de la digue (m)	8
6	Zb	Côte de la berge (m)	55,5
7	L	Longueur de la rivière (m)	5000
8	B	Largeur de la rivière (m)	300

Buttons: + Add, - Remove

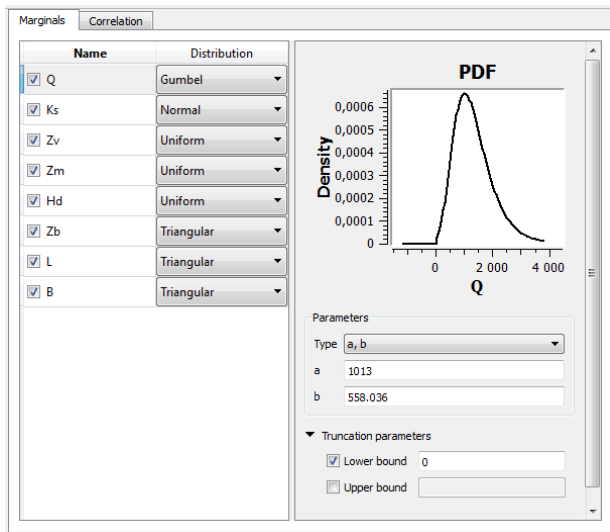
**Outputs Table:**

	Name	Description	Formula	Value
1	H	Surverse (m)	$(Q / (Ks * B * \sqrt{(Zm - Zv) / L}))^{(3.0 / 5.0)} + Zv - Zb - Hd$	-13,5

Buttons: + Add, - Remove, Evaluate



# Probabilistic model



## Limit state study : definition of the threshold

Definition of the failure event :

Output	Operator	Threshold
H ▼	< ▼	-10

# Limit state study : algorithm parameters



# Limit state study : summary

Summary

Histogram

Convergence graph

Output H

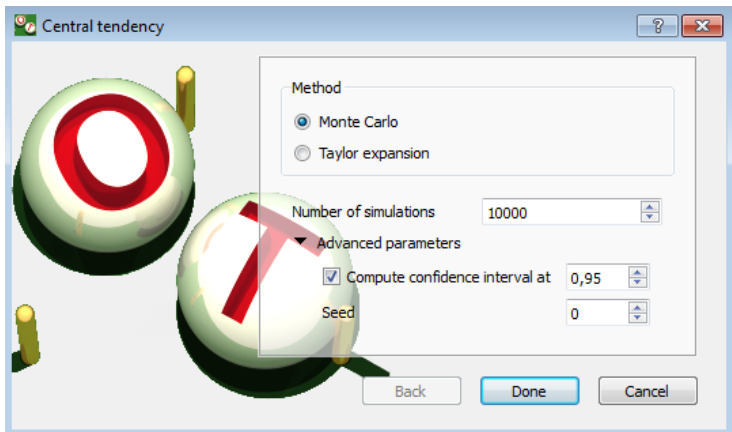
Number of simulations: 26

Estimate	Value	Confidence interval at 95%	
		Lower bound	Upper bound
Failure probability	0.807692	0.656203	0.959182
Coefficient of variation	0.0956949		

# Limit state study : histogram



## Central tendency : algorithm parameters



# Central tendency : summary results

Moments estimate

Estimate	Value	Confidence interval at 95%	
		Lower bound	Upper bound
Mean	-11.0178	-11.0417	-10.9938
Standard deviation	1.22309	1.20637	1.24028
Skewness	0.20005		
Kurtosis	3.01907		
First quartile	-11.8721		
Third quartile	-10.2129		

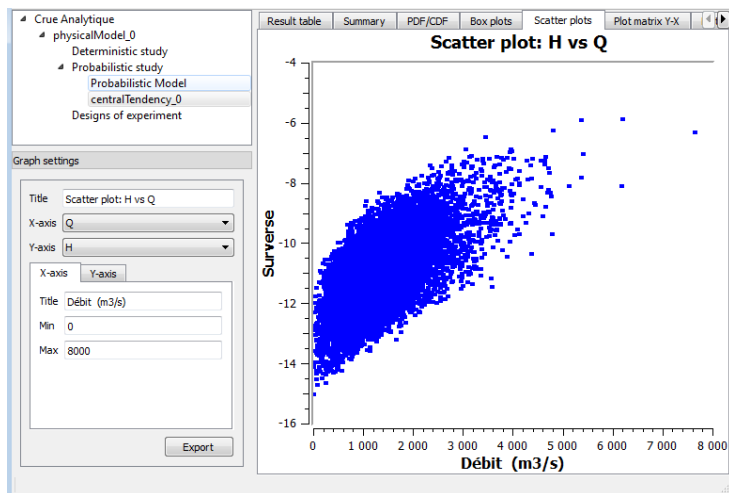
**Probability**   
**Quantile**

# Central tendency : summary results

Result table	Summary	PDF/CDF	Box plots	Scatter plots
Output <span>H ▾</span>				
Number of simulations: 10000				
Minimum and Maximum				
	Variable	Minimum	Maximum	
Output	H	-15.0155	-5.88758	
Inputs at extremum	Q	7.97827	6187.43	
	Ks	27.132	25.5926	
	Zv	49.1681	50.9071	
	Zm	54.5469	55.3994	
	Hd	8.76082	8.49391	
	Zb	55.5436	55.4935	
	L	4999.26	4997.37	
	B	303.187	300.871	



# Central tendency : scatter plots



# Sensitivity analysis : Sobol' indices

