Software for Uncertainty Quantification (MS91)

8:35-9:00 - OpenTURNS for Uncertainty Quantification

Michael Baudin, Anne Dutfoy, Anne-Laure Popelin, *EDF, France*

9:05-9:30 - Promethee Environment for Computer Code Inversion

Yann Richet, Gregory Caplin, *IRSN, France*

9:35-10:00 - ‘Mystic’: Highly constrained Non-convex Optimization and Uncertainty Quantification

Michael McKerns, *California Institute of Technology, USA*

10:05-10:30 - Uranie: the Uncertainty and Optimization Platform

Fabrice Gaudier, Jean-Marc Martinez, Gilles Arnaud, *CEA, France*

OpenTURNS for Uncertainty Quantification

Thursday, April 7
MS91 : Software for Uncertainty Quantification
SIAM Conference on Uncertainty Quantification

Michael Baudin, Anne Dutfoy, Anne-Laure Popelin

Industrial Risk Management, EDF R&D, Chatou, France

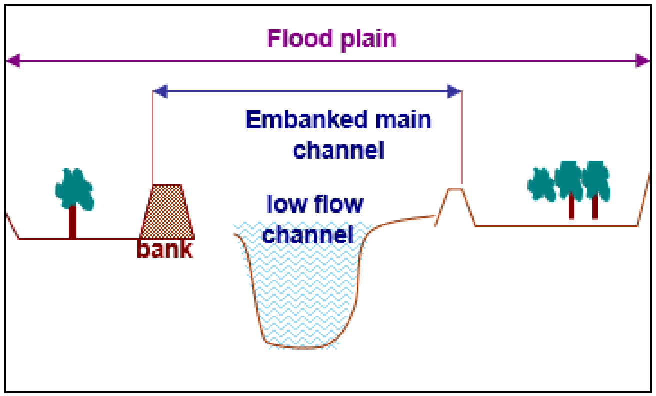
michael.baudin@edf.fr, anne.dutfoy@edf.fr, anne-laure.popelin@edf.fr

CHANGER L'ÉNERGIE ENSEMBLE

Some EDF applications

- Safety and reliability of structures :
  - Margins assessment
  - Safety compliance
- Calibration/validation
  - quality of our studies
  - validity domain of numerical codes (« VVUQ »)
- Robust Optimization
  - Performance of new systems
  - Optimization of maintenance policy

Cooling tower, Civaux (Vienne)



Inverse calibration of Strickler coefficient in a flood model

Optimization of buildings' energy performance

Outline

- Global framework of uncertainty treatment
- OpenTURNS overview
  - A simple exercise
  - External modules
- New features and perspectives
  - Optimized LHS
  - EGO
  - GUI
  - Visualization in uncertainty study

Outline

- Global framework of uncertainty treatment

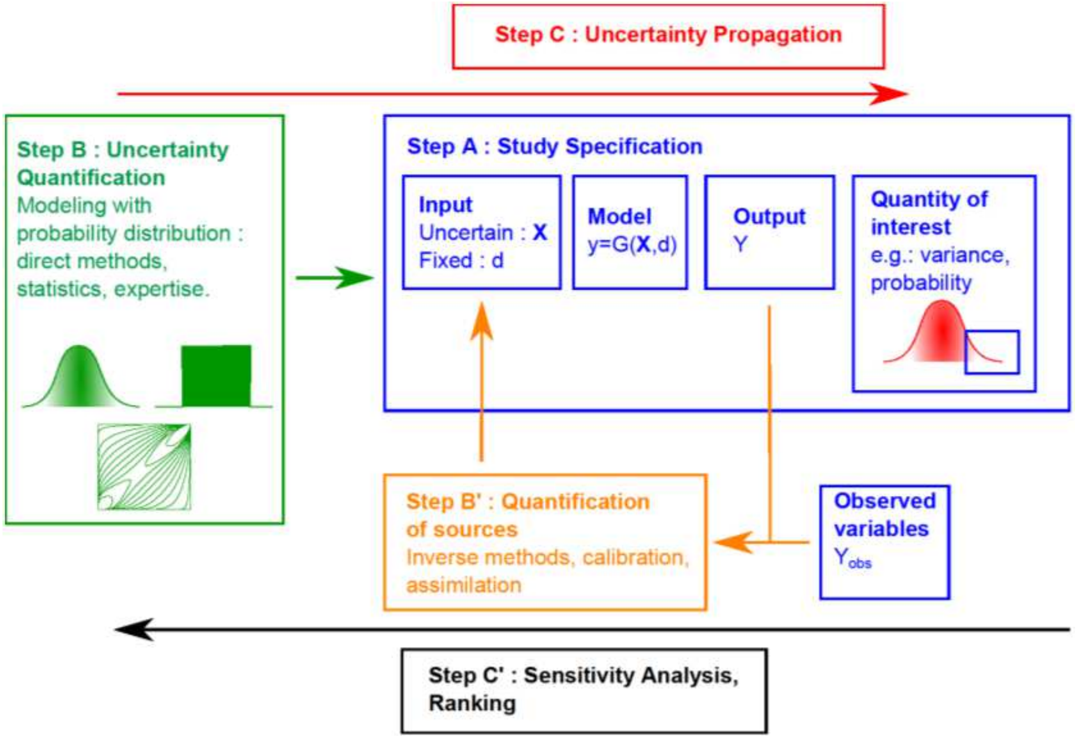
- OpenTURNS overview
  - A simple exercise
  - External modules

- New features and perspectives
  - Optimized LHS
  - EGO
  - GUI
  - Visualization in uncertainty study

Uncertainty methodology

OpenTURNS uses the « Global Methodology of Treatment of Uncertainties » :

- Step A : Study specification : uncertain input variables, model, variable of interest, quantity of interest (e.g. central dispersion)
- Step B : Uncertainty quantification : defines the joint distribution of the input (e.g. μ in the Normal distr.)
- Step C : Uncertainty propagation : estimates the quantity of interest (e.g. the mean)
- Step C’ : Sensitivity analysis : ranks the input variables from the variable with highest impact to the lowest



Outline

- Global framework of uncertainty treatment
- OpenTURNS overview
  - A simple exercise
  - External modules
- New features and perspectives
  - Optimized LHS
  - EGO
  - GUI
  - Visualization in uncertainty study

OpenTURNS : www.openturns.org

Features

- Steps : A, B, C, C'
- Stochastic processes
- **Meta-models** : polynomial chaos, kriging, support vector machine
- **Sensitivity analysis** : Morris, Sobol'
- **Threshold probability** : FORM/SORM, Subset Sampling, Adaptive Directional Sampling

Computer code G :

- Multi-thread evaluation of an analytical formula (exact derivatives)
- Distributed and multi-thread evaluation of a Python function (with finite differences)
- Evaluation by SALOME

Context :

- Since 2005
- Four partners :
- Documentation : for Python users, for developers
- Programing interface : Python module, C++ library
- Licence : LGPL
- Linux, Windows
- 1 technical committee / month
- 1 board committee / month

OpenTURNS : a simple exercise

- Four independent input variables

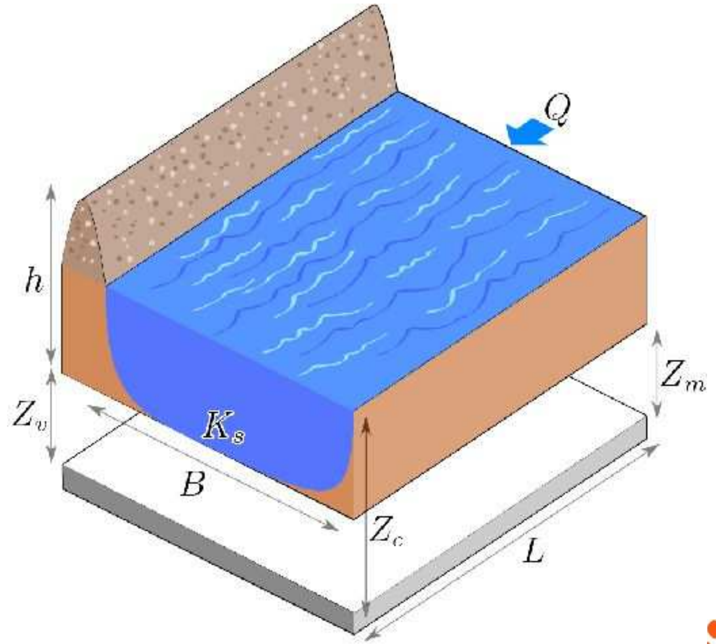
Variable	Distribution
Q : max. annual discharge (m^3/s)	Gumbel (mode=1013, scale=558)
Ks : Manning-Strickler coefficient $(m^{1/3}/s)$	Normal(mean=30, st.dev.=7.5)
Zv : downstream level of the riverbed (m)	Uniform(49,51)
Zm : upstream level of the riverbed (m)	Uniform(54,55)

- Output : overflow S

$$H = \left(\frac{Q}{300K_s\sqrt{\frac{Z_m - Z_v}{5000}}}\right)^{0.6}$$

$$S = Z_v + H - 58.5$$

- Quantity of interest : $\mathrm{P}(S > 0)$



OpenTURNS : a simple exercise in Python

Here is a simple exercise in Python, which is used in the OpenTURNS training.

We first import the openturns module.

```
from openturns import *
```

We then define the G Python function which takes the input X as argument and returns the output S.

```
# 1. Define G
def functionCrue(X) :
    Q, Ks, Zv, Zm = X
    alpha = (Zm - Zv)/5.0e3
    H = (Q/(300.0*Ks*sqrt(alpha)))**0.6
    S = [H + Zv - 58.5]
    return S
```

We finally use the PythonFunction class which converts the Python function into a function callable by OpenTURNS.

```
myWrapper = PythonFunction(4,1,functionCrue)
```

The next step is to define each input variable and its marginal distribution.

```
# 2. Define input variables
Q = Gumbel(1./558., 1013.)
Q = TruncatedDistribution(Q, 0, inf)
Ks = Normal(30.0, 7.5)
Ks = TruncatedDistribution(Ks, 0, inf)
Zv = Uniform(49.0, 51.0)
Zm = Uniform(54.0, 56.0)
```

We create the input random vector by combining the marginal distributions with an independent copula and create the output random vector.

```
# 3. Create the output Y=G(X)
inputX = ComposedDistribution([Q, Ks, Zv, Zm])
inputvector = RandomVector(inputX)
outputvector = RandomVector(myWrapper, inputvector)
```

OpenTURNS : a simple exercise in Python

We define the event by combining the output random vector with the zero threshold and a comparison operator.

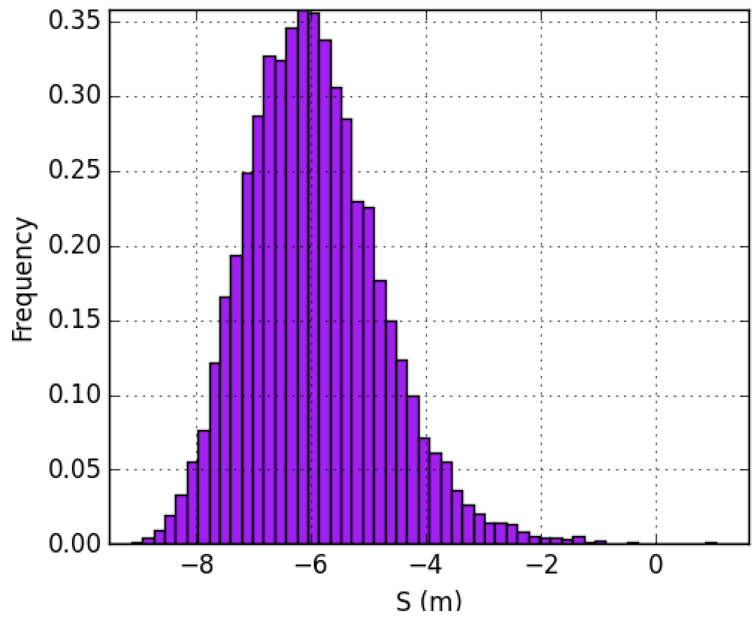
```
# 4. Estimate the probability
eventF = Event(outputvector, GreaterOrEqual(), 0)
```

We use a simple Monte-Carlo estimate.

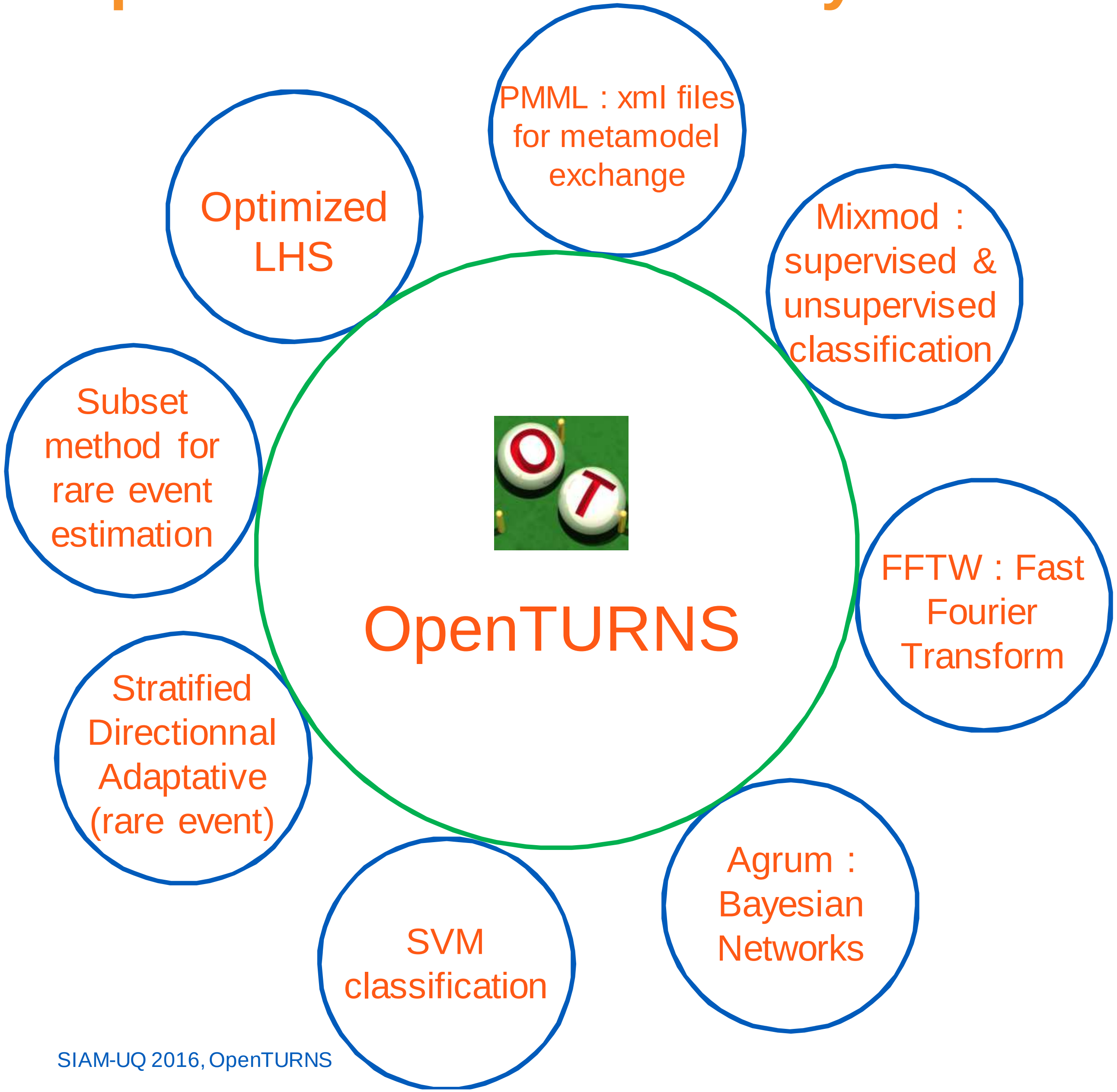
```
algoProb = MonteCarlo(eventF)
algoProb.setMaximumOuterSampling(1000000)
algoProb.setMaximumCoefficientOfVariation(0.1)
algoProb.run()
resultAlgo = algoProb.getResult()
neval = myWrapper.getEvaluationCallsNumber()
pf = resultAlgo.getProbabilityEstimate()
```

In the Python console :

```
Number of function calls = 153501
Failure Probability = 6.514616e-04
```



OpenTURNS : modularity



How to contribute to OpenTURNS?

doc.openturns.org/sphinx/contribute

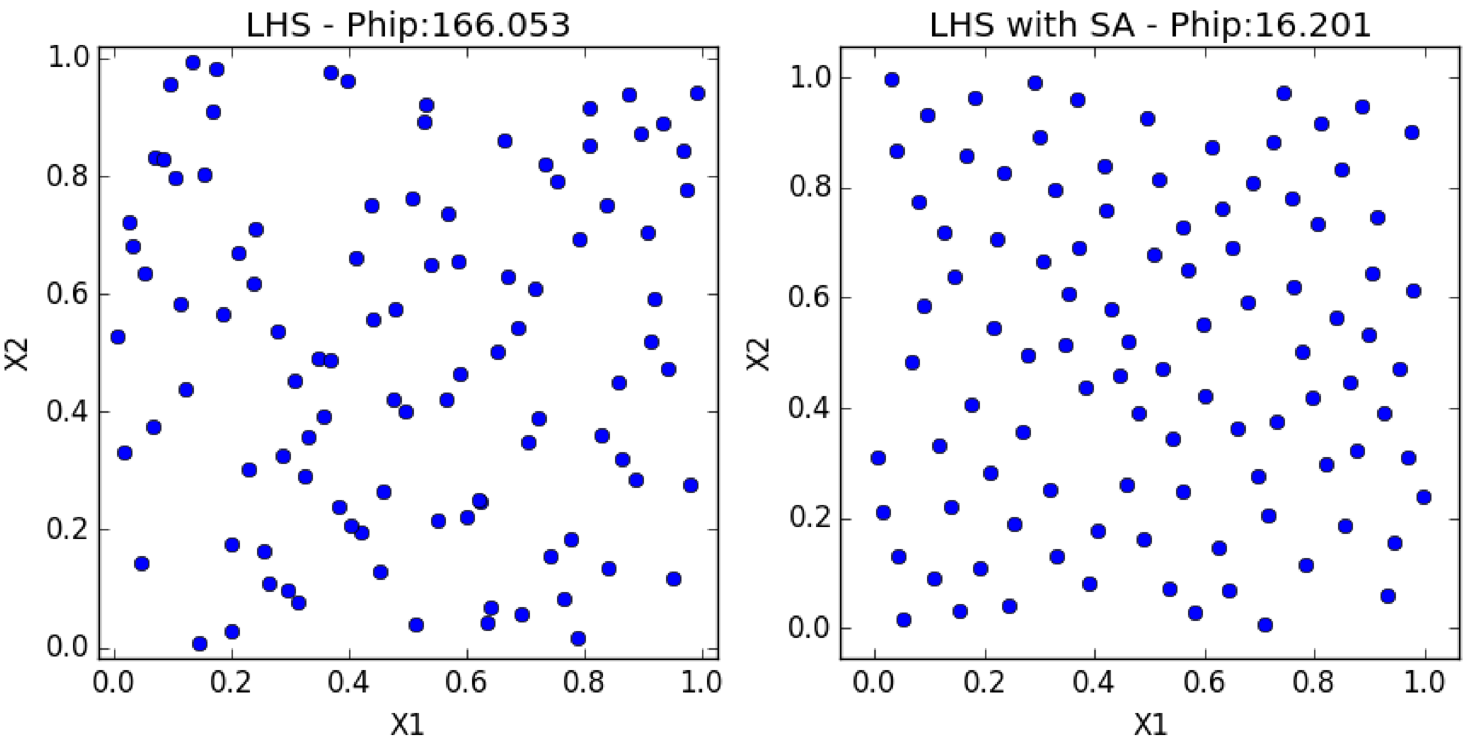
- Report bugs on the bugtracker
- Suggest a new feature on github
- Your developpment can be included in the core or as a module

Outline

- Global framework of uncertainty treatment
- OpenTURNS overview
  - A simple exercise
  - External modules
- New features and perspectives
  - Optimized LHS
  - EGO
  - GUI
  - Visualization in uncertainty study

The Optimized LHS module

- Goal
  - Create space-filling, optimized Latin Hypercube Sampling designs
- Algorithm :
  - Several space-fillling criteria : mindist or phi-p
  - Optimization by simulated annealing (or Monte-Carlo)



Two LHS with 100 points
left : classical, right : optimized with simulated annealing

Ongoing work : Efficient Global Optimization (EGO)

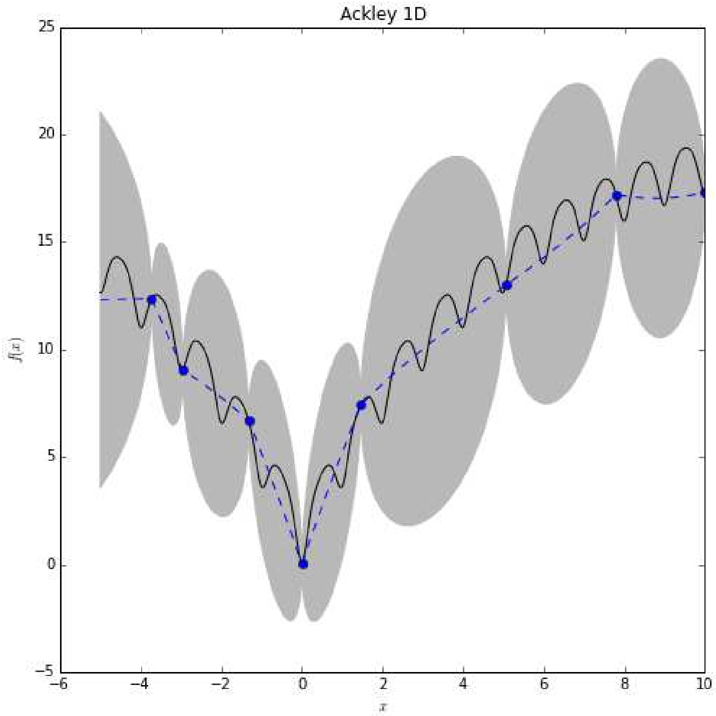
Objectives

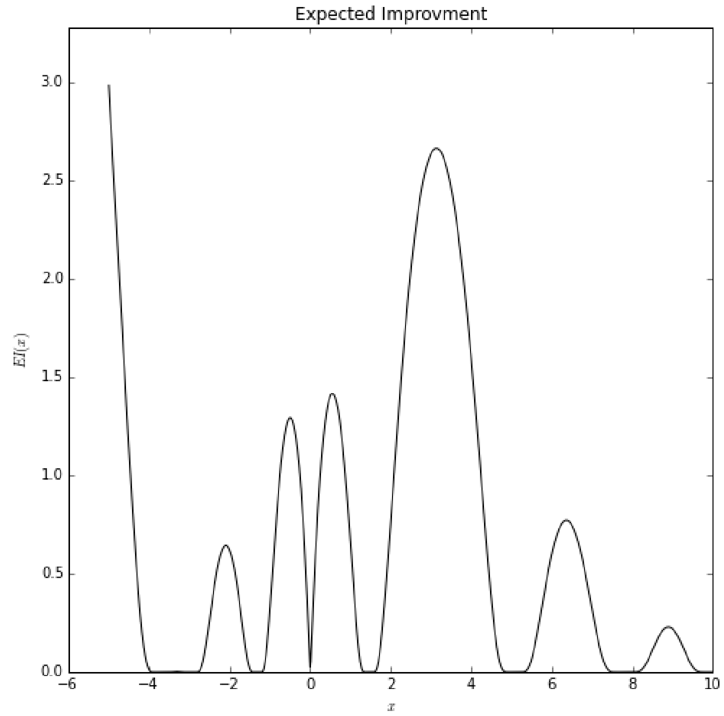
- Optimize a nonlinear, blackbox, costly objective function.
- Get a global optimum by exploring the whole input space.
- Use a kriging meta-model.

Algorithm

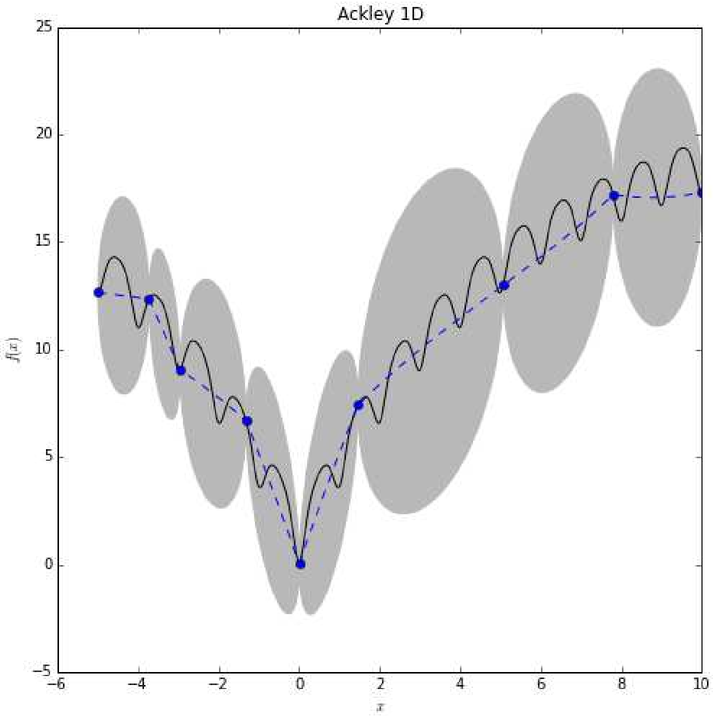
- Create an initial design of experiment D (e.g. optimized LHS)
- Evaluate the G function on the design
- Create the kriging meta-model
- Loop over the iterations :
  - Find a new point X minimizing the expected improvement
  - Add this point to the design D, evaluate Y=G(X)
  - Update the kriging meta-model
- Output : the best point X so far

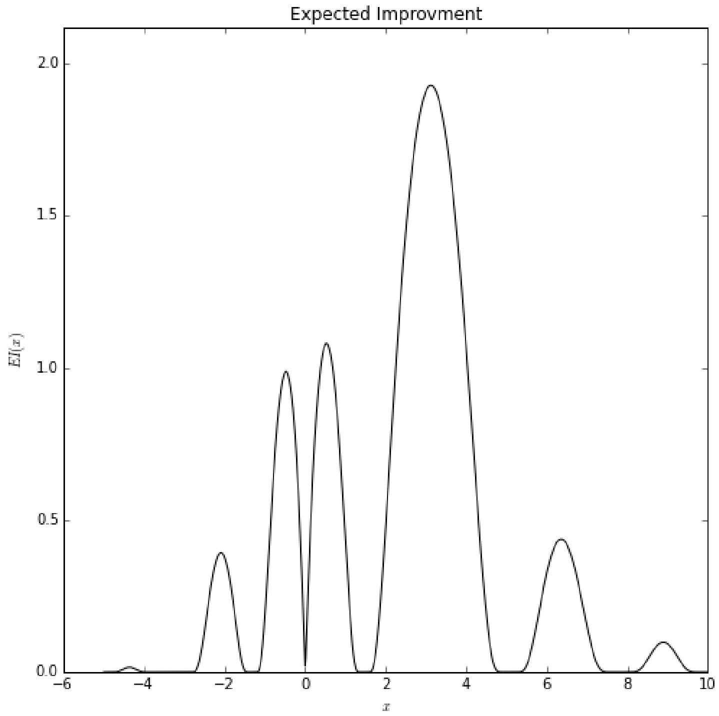
Efficient Global Optimization (EGO)



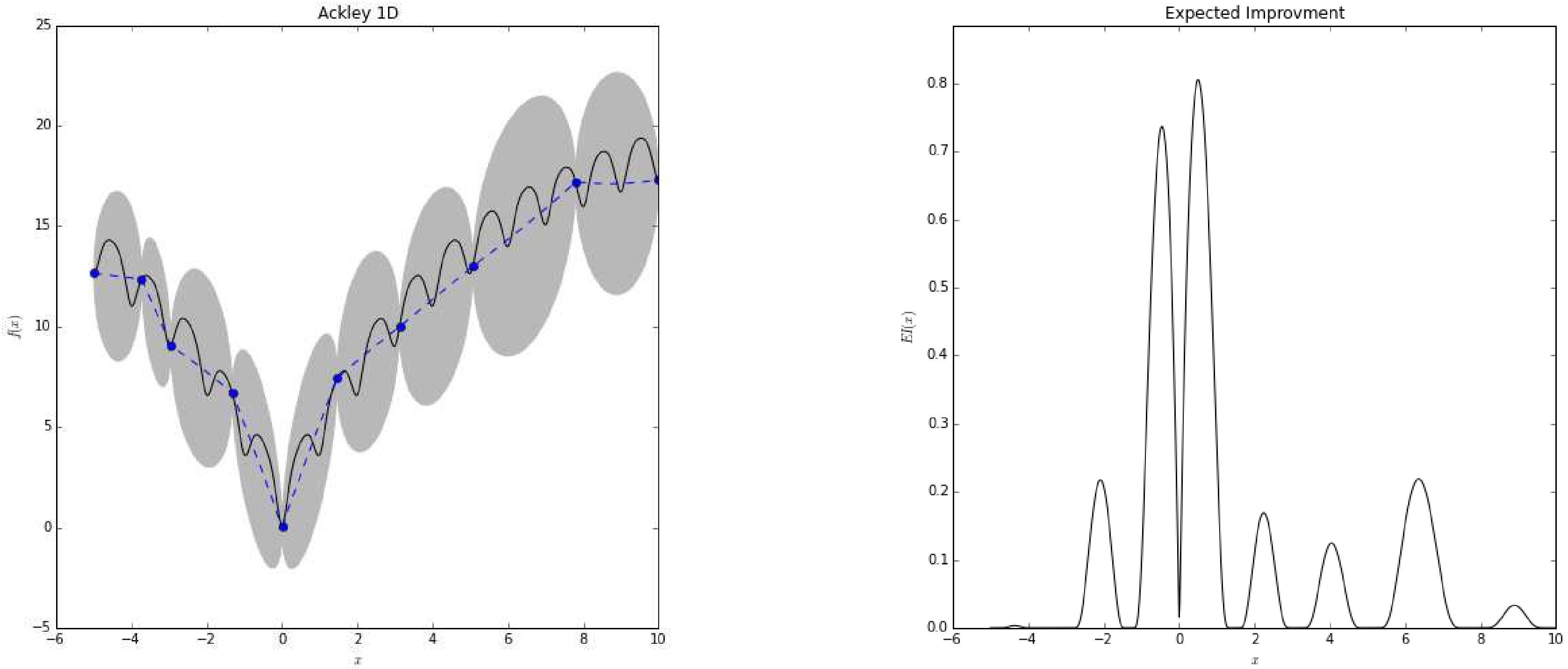


Efficient Global Optimization (EGO)

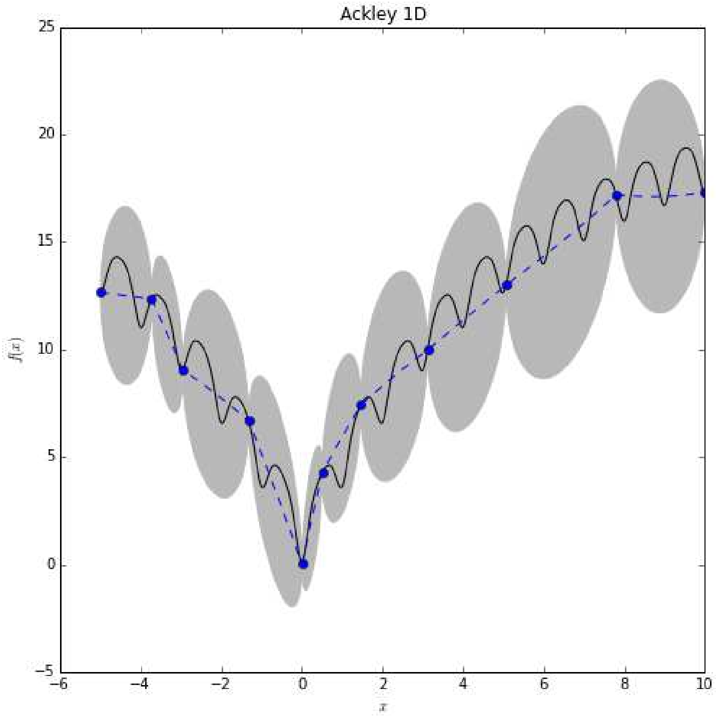


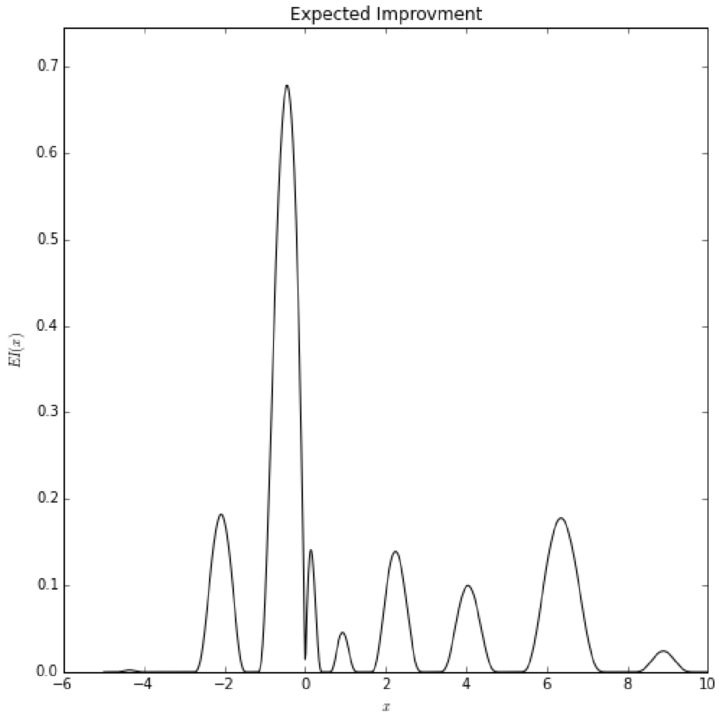


Efficient Global Optimization (EGO)



Efficient Global Optimization (EGO)





- In OpenTURNS v1.8 (summer 2016)

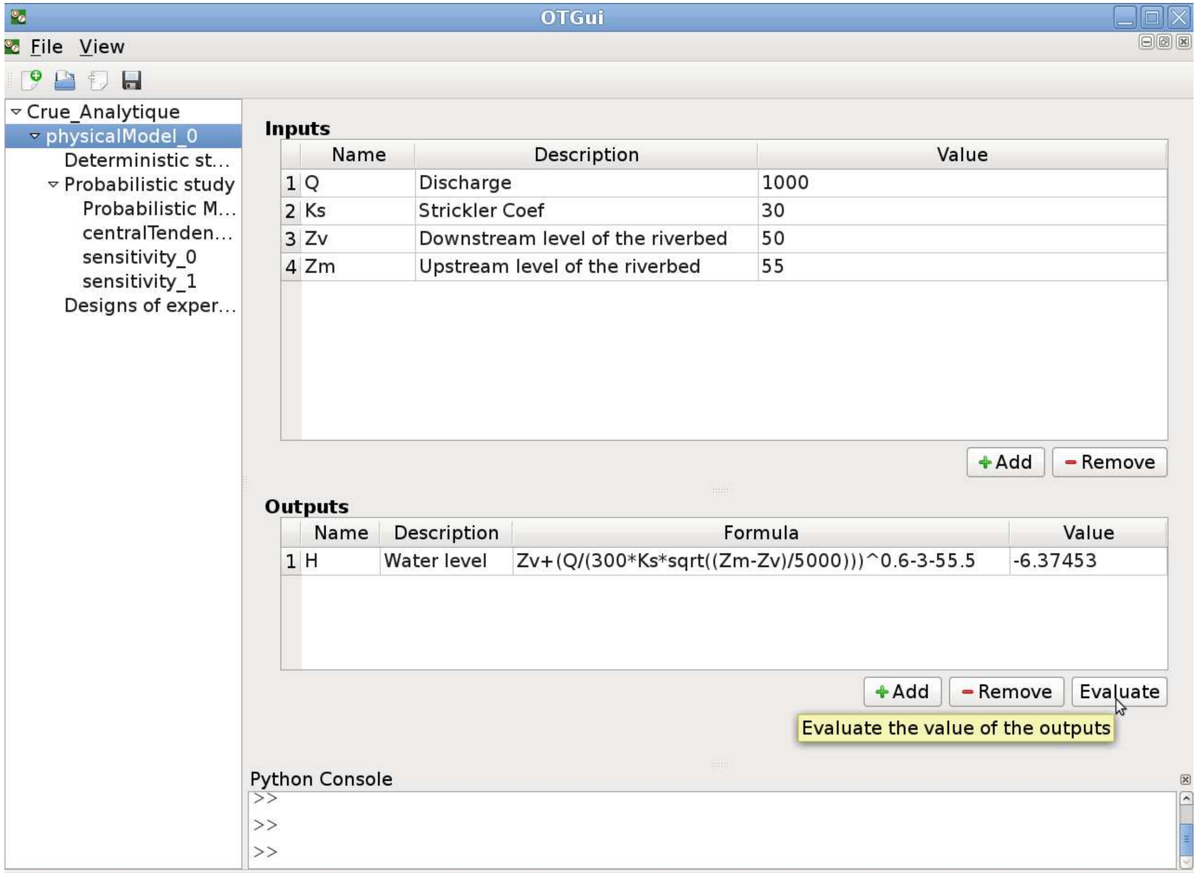
OpenTURNS Graphical User Interface

- GUI features
  - Generic (not dedicated to a specific application)
  - Access to the main functions of A, B, C, C' steps
- Schedule :
  - Release : Summer 2016
  - Then : one release each year
- Development : with our partner Phimeca (50% / 50%)
- Licence : LGPL

OpenTURNS Graphical User Interface

Definition of the physical model

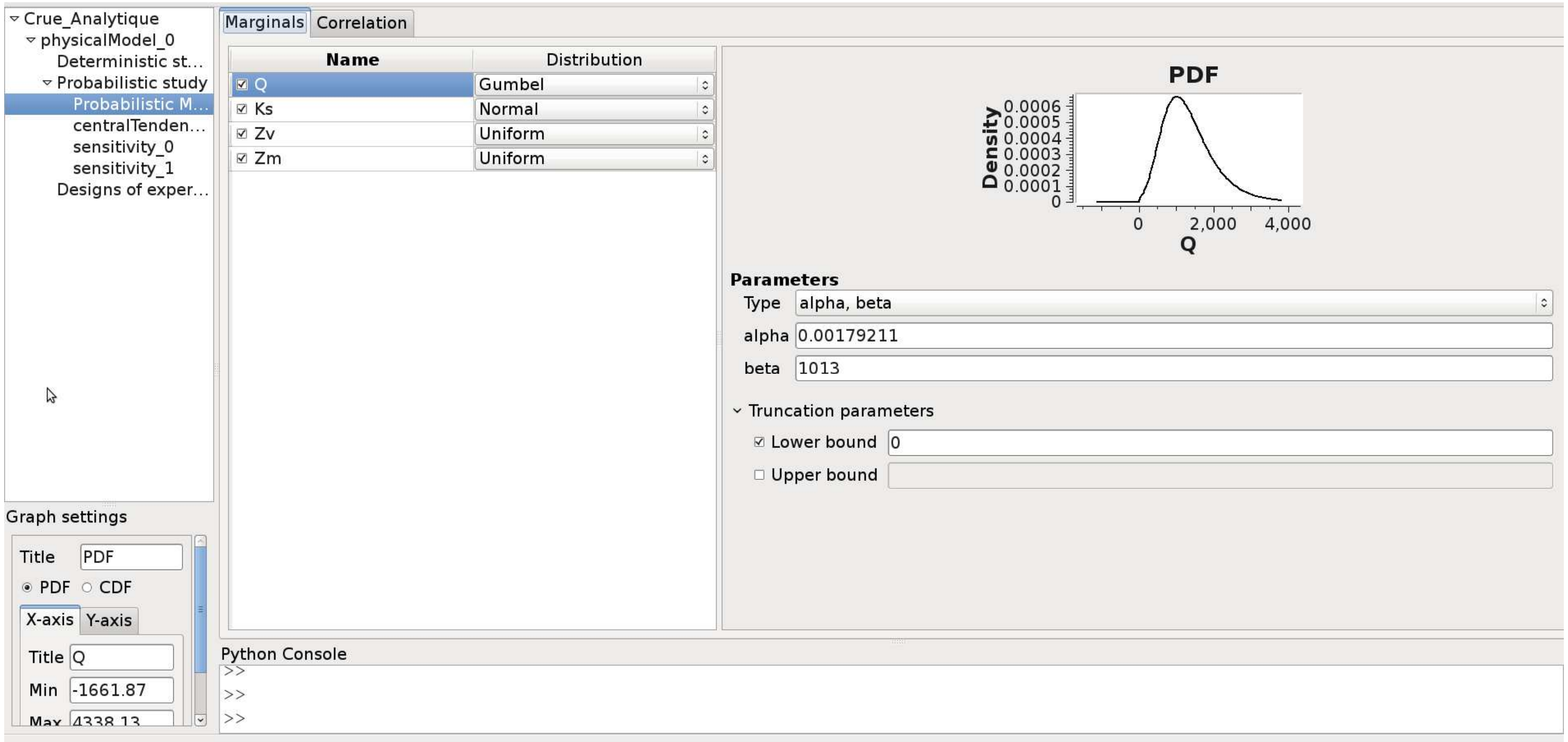
- Can be :
  - An analytical formula
  - A Python Function
  - An external model given in a dedicated xml file



OpenTURNS Graphical User Interface

Uncertainty Quantification step : Define the joint distribution :

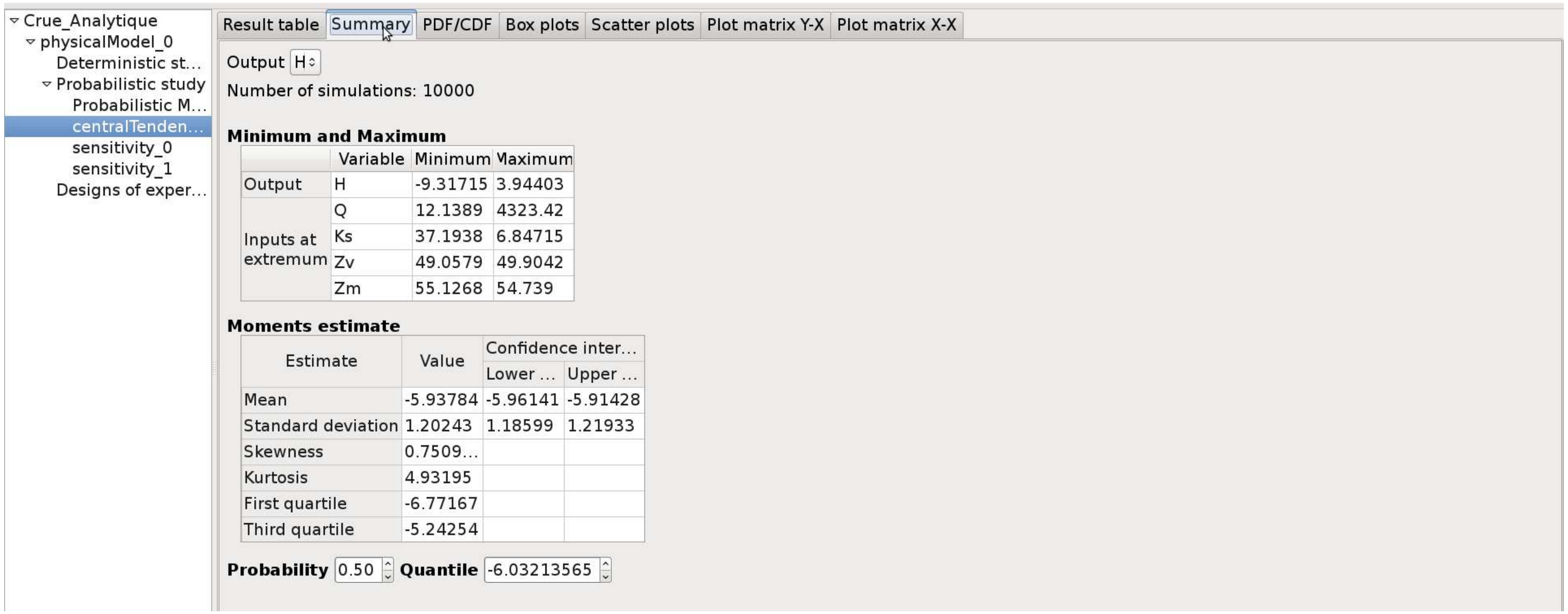
- marginals and their parameters
- correlation (Gaussian copula in the GUI)



OpenTURNS Graphical User Interface

Uncertainty propagation results : central dispersion (by Monte Carlo Sampling here)

- analysis of the summary statistics
- other tabs : graphical results



Minimum and Maximum

	Variable	Minimum	Maximum
Output	H	-9.31715	3.94403
Inputs at extremum	Q	12.1389	4323.42
	Ks	37.1938	6.84715
	Zv	49.0579	49.9042
	Zm	55.1268	54.739

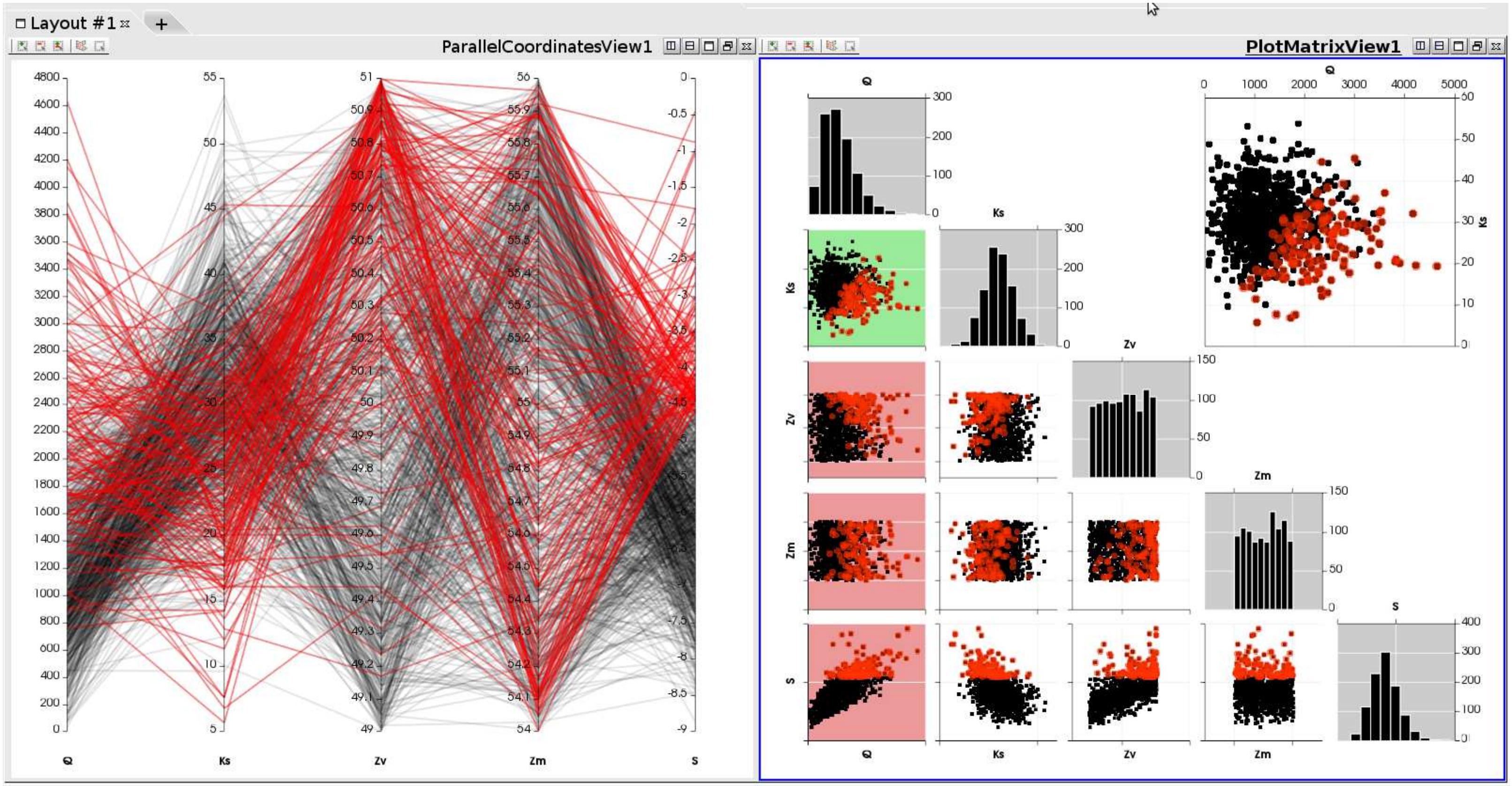
Moments estimate

Estimate	Value	Confidence inter...	
		Lower ...	Upper ...
Mean	-5.93784	-5.96141	-5.91428
Standard deviation	1.20243	1.18599	1.21933
Skewness	0.7509...		
Kurtosis	4.93195		
First quartile	-6.77167		
Third quartile	-5.24254		

Probability 0.50 **Quantile** -6.03213565

OpenTURNS Graphical User Interface

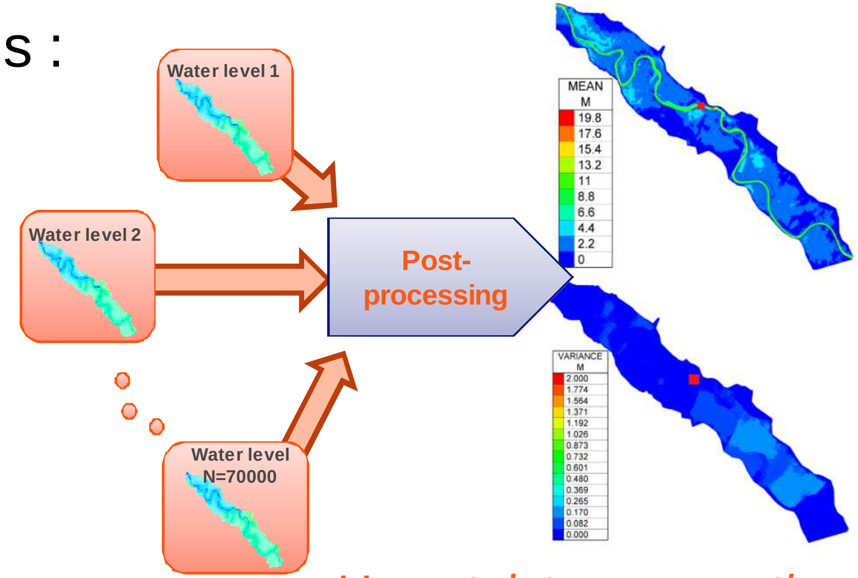
- For the v2 : data visualization with dedicated tool « Paraview »



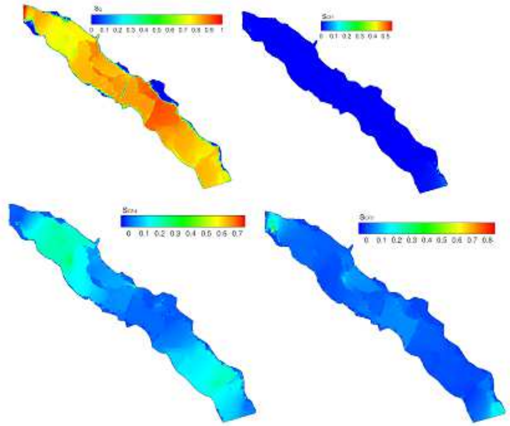
Visualization : next steps

See also : Wednesday, April 6 - MS61 : Visualization in Computer Experiments

- Fields :

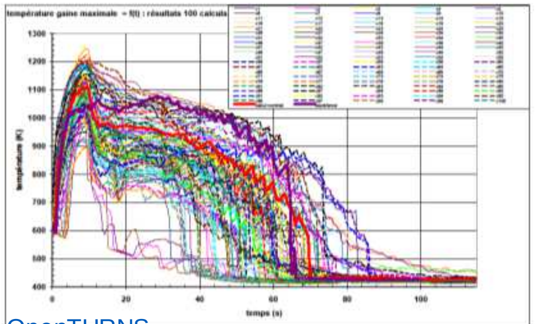


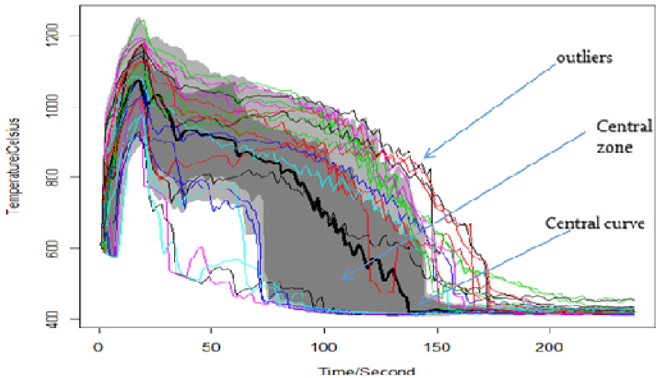
Uncertainty propagation



Sensitivity Analysis (Sobol')

- Functionnal (2D or 3D) boxplot:





Thanks !

[1] www.openturns.org

[2] Handbook for UQ, Springer, 2016

users@openturns.org

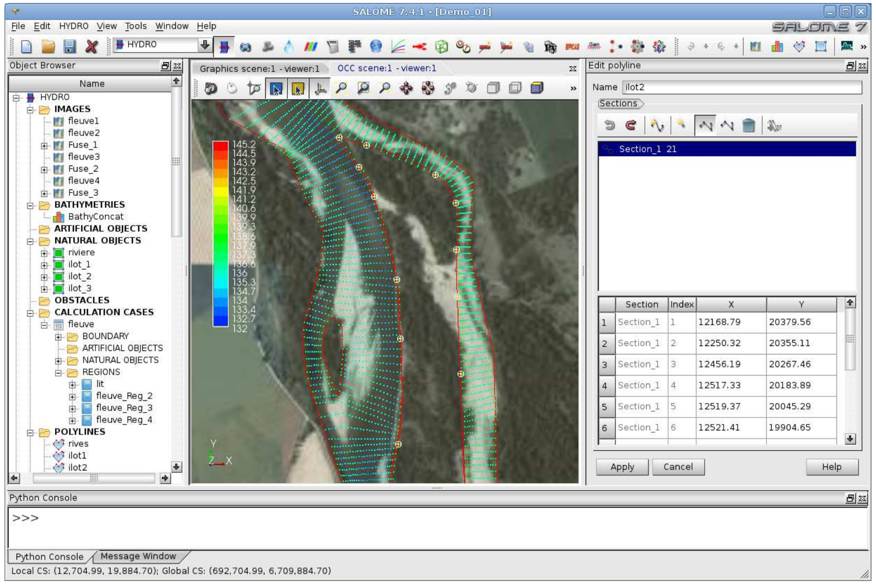
Next users day : 21st June 2016, Chatou, France

OpenTURNS and SALOME

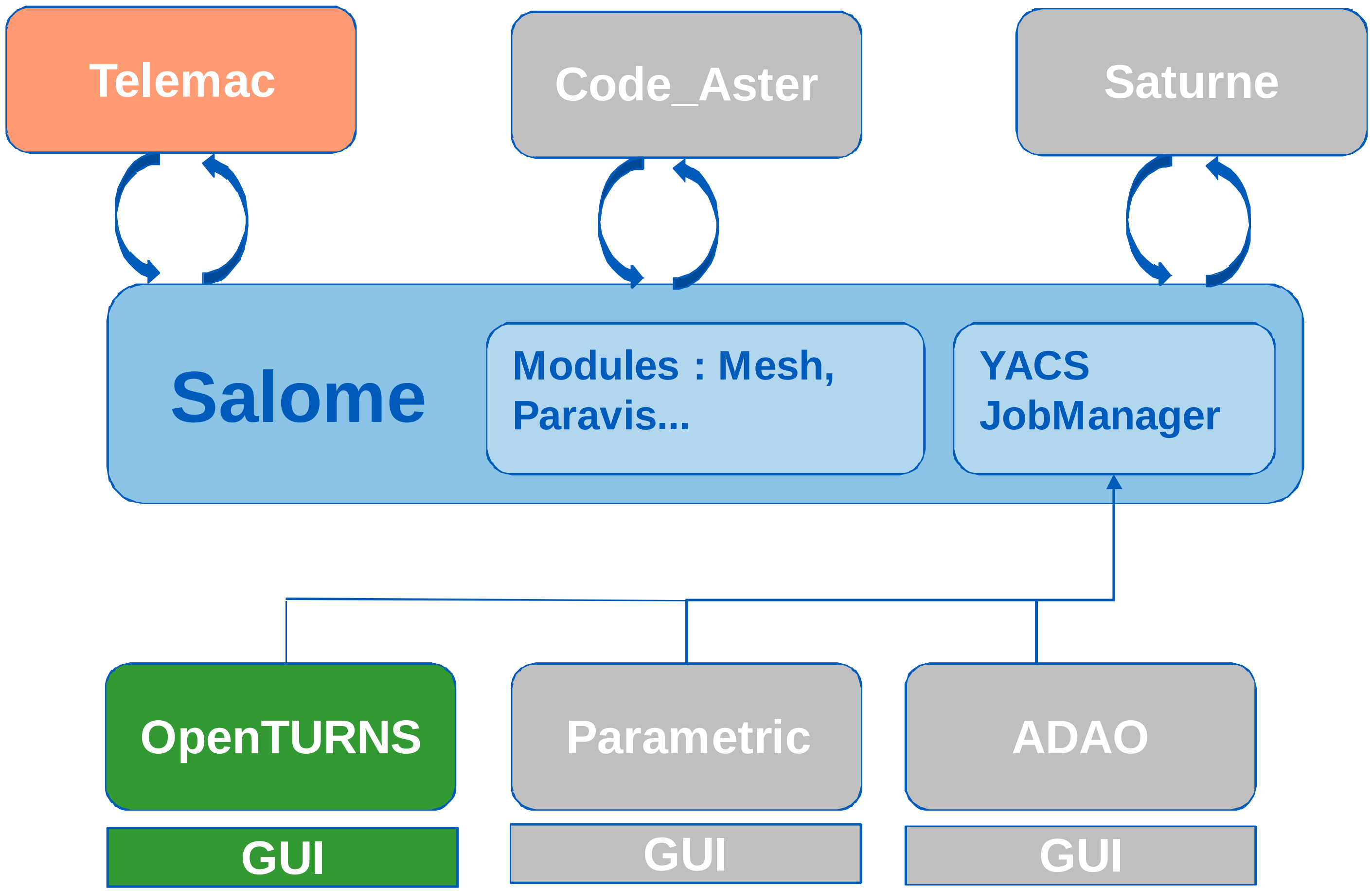
SALOME

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

A specific platform : SALOME-HYDRO

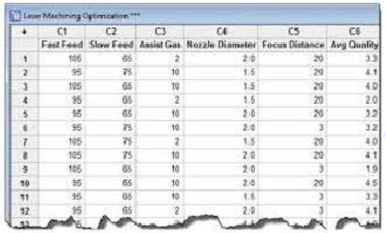


OpenTURNS and Salome



OPENTURNS : UTILISATION DE YACS

Plan d'expérience
(à la main ou avec OpenTURNS)

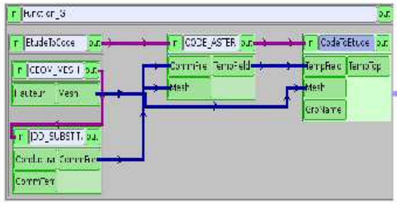


Laser Machining Optimization ***

+	C1	C2	C3	C4	C5	C6
	Fast Feed	Slow Feed	Assist Gas	Nozzle Diameter	Focus Distance	Avg Quality
1	105	65	2	2.0	20	3.3
2	95	75	10	1.5	20	4.1
3	105	65	10	1.5	20	4.0
4	95	65	2	1.5	20	2.0
5	95	65	10	2.0	20	3.2
6	95	75	10	2.0	3	3.2
7	105	75	2	1.5	20	4.0
8	105	75	10	2.0	20	4.1
9	105	65	10	2.0	3	1.9
10	95	65	10	2.0	20	4.5
11	95	65	10	1.5	3	3.3
12	95	65	2	2.0	3	4.1

X_i

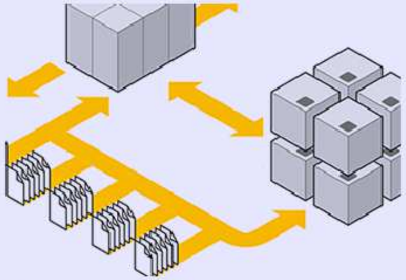
Calcul unitaire
(avec SALOME.YACS + codes)



$Y=f_c(X)$

Calcul paramétrique
(avec SALOME.YACS)

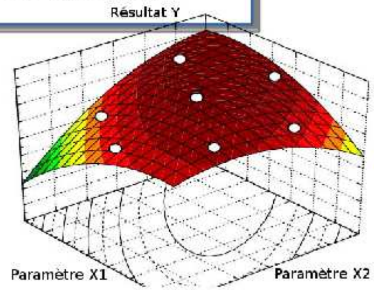
- Distribution des calculs
- Optimisation des ressources HPC
- Gestion des données (X,Y)
- Reprise sur panne/arrêt



(X_i, Y_i)

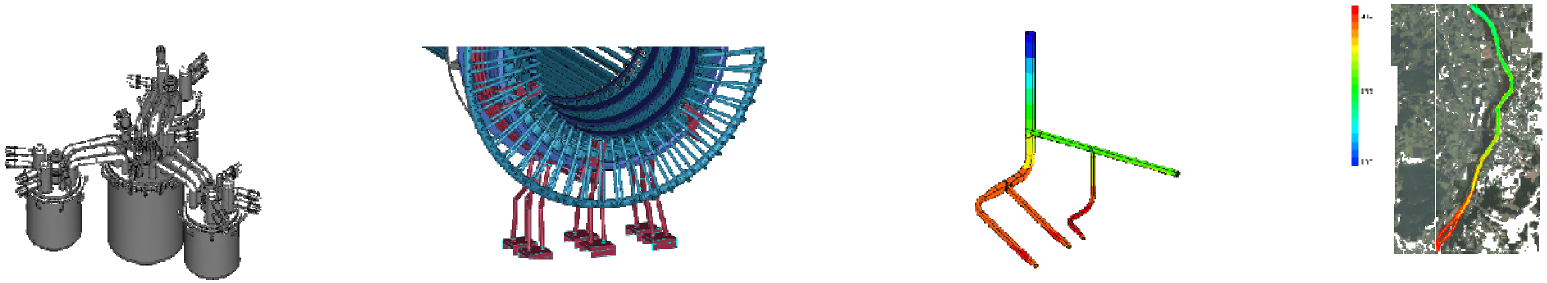
Analyse des résultats
(avec OpenTURNS, SALOME.PARAVIS)

- Méta-modélisation
- Analyse statistique
- Visualisation



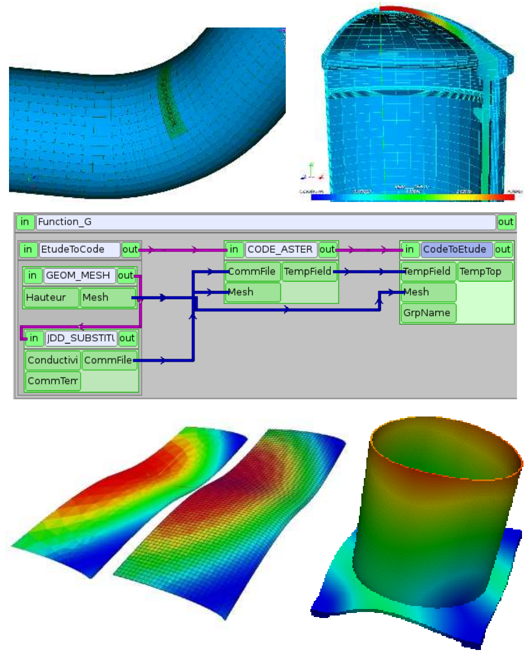
Contexte général de SALOME

- Simulation numérique des systèmes physiques d’intérêt EDF



  - Mécanique (Code_Aster)
  - Thermohydraulique (Code_Saturne, Neptune_CFD)
  - Electromagnétisme (Code_CARMEL3D)
  - Neutronique (ANDROMEDE)
  - Hydraulique à surface libre (Telemac, Mascaret)
  - …

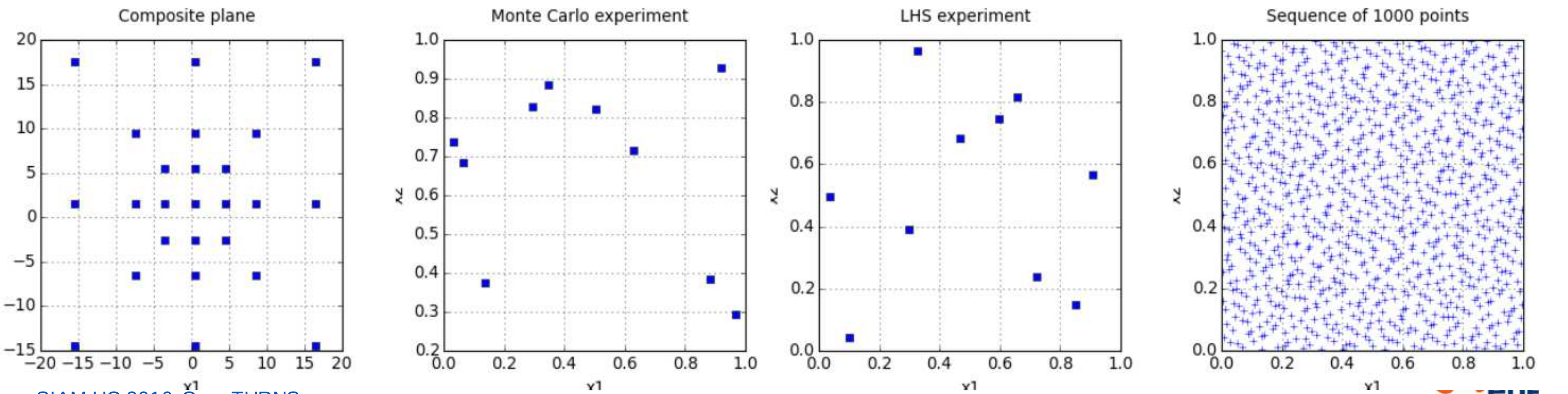
- Besoins génériques de la simulation dans ce domaine
  - Modélisation 3D (CAO, maillages, visualisation)
  - Orchestration des calculs (composition, distribution)
  - Traitement de données complexes (champs, matrices)



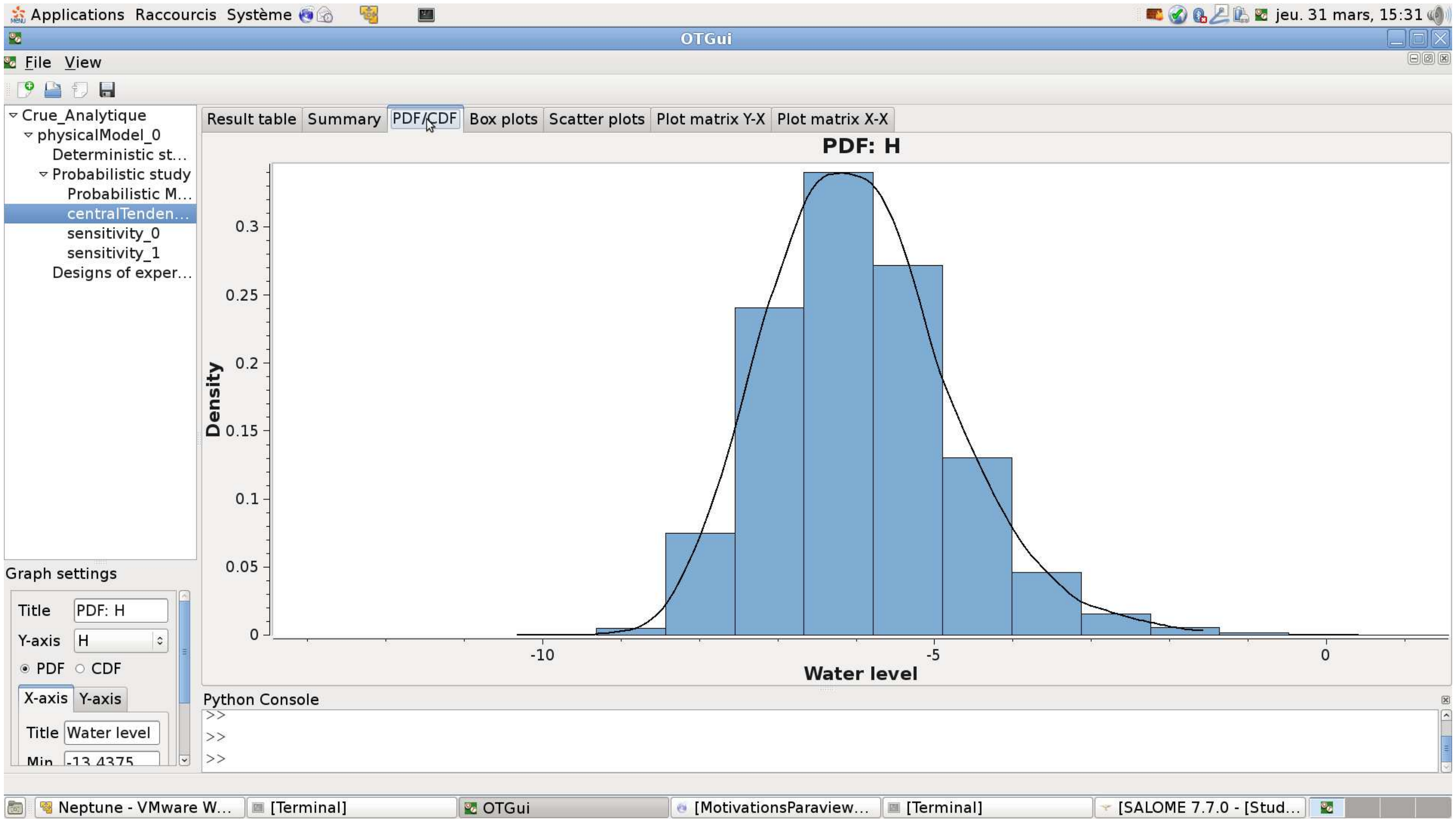
= Plateforme SALOME

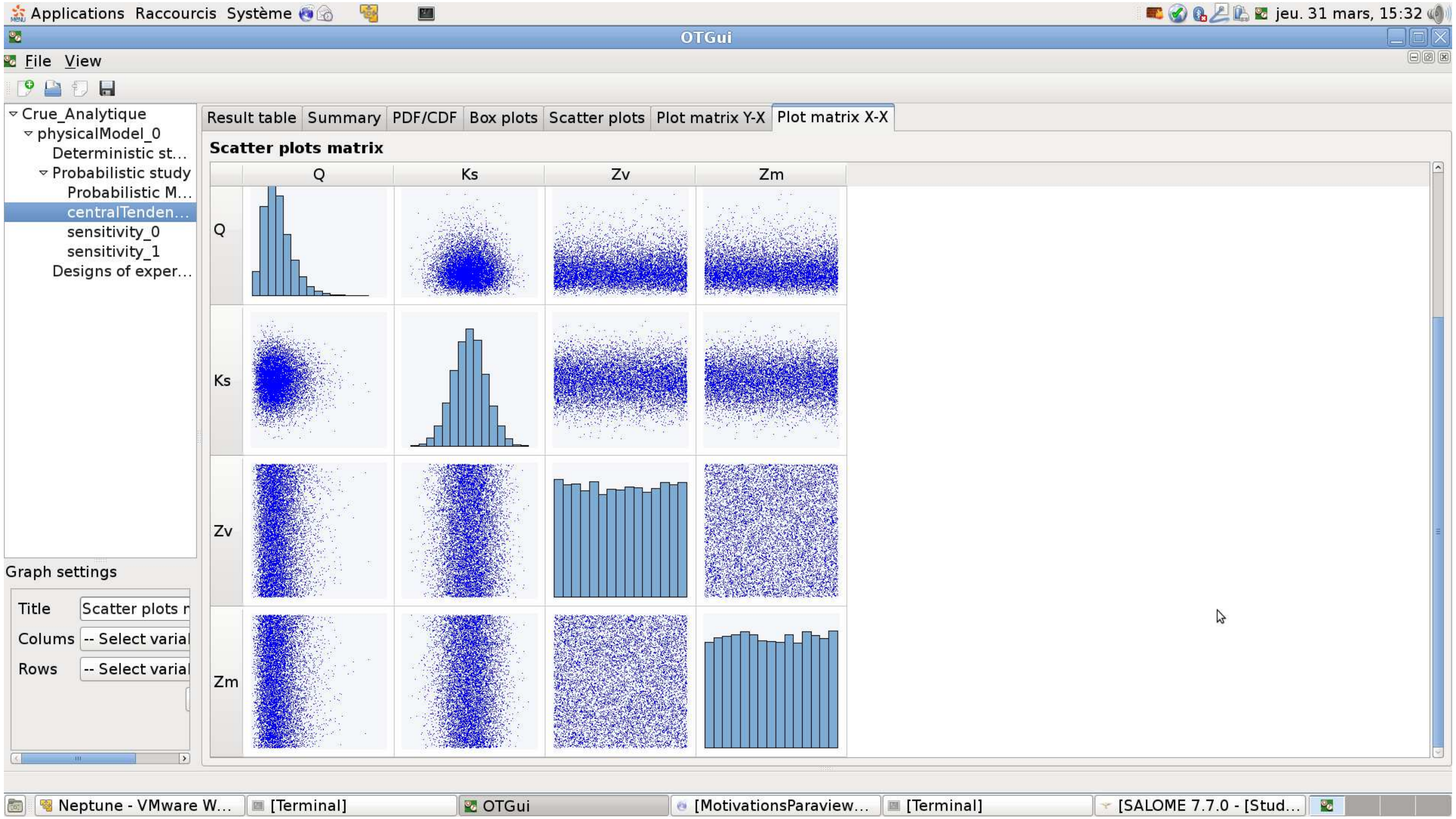
Design of experiments

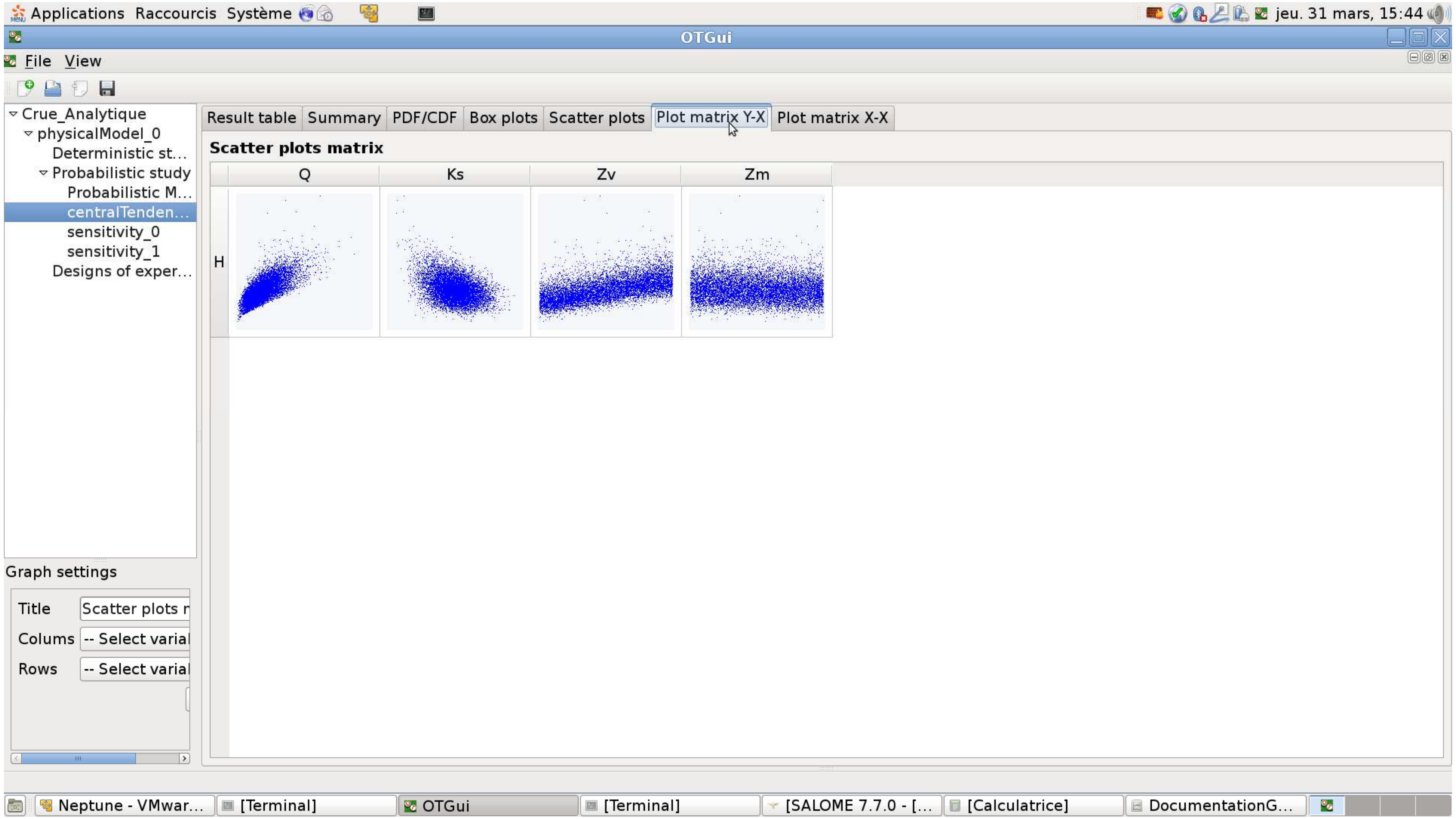
- What for ?
  - Model exploration
  - Central tendency : estimate mean, standard variation
  - Create meta-models, e.g. Polynomial chaos, Kriging, etc…
- Deterministic DOEs :
  - central composite, factorial : axial, composite, box
- Probabilistic DOEs :
  - Monte Carlo with given distribution
  - Bootstrap resampling
  - Importance sampling
  - Latin Hypercuve Sampling (LHS) and optimized LHS
- Low discrepancy sequences : Sobol, Faure, Halton



OpenTURNS GUI




Applications Raccourcis Système
jeu. 31 mars, 15:32
OTGui
File View
Crue_Analytique
physicalModel_0
Deterministic st...
Probabilistic study
Probabilistic M...
centralTenden...
sensitivity_0
sensitivity_1
Designs of exper...
Result table
Summary
PDF/CDF
Box plots
Scatter plots
Plot matrix Y-X
Plot matrix X-X
Scatter plots matrix
Q
Ks
Zv
Zm
Graph settings
Title
Scatter plots r
Colums
-- Select varial
Rows
-- Select varial
Neptune - VMware W...
[Terminal]
OTGui
[MotivationsParaview...
[Terminal]
[SALOME 7.7.0 - [Stud...



Applications Raccourcis Système
jeu. 31 mars, 15:44
OTGui
File View
Crue_Analytique
physicalModel_0
Deterministic st...
Probabilistic study
Probabilistic M...
centralTenden...
sensitivity_0
sensitivity_1
Designs of exper...
Result table
Summary
PDF/CDF
Box plots
Scatter plots
Plot matrix Y-X
Plot matrix X-X
Scatter plots matrix
Q
Ks
Zv
Zm
H
Graph settings
Title
Scatter plots r
Colums
-- Select varial
Rows
-- Select varial
Neptune - VMwar...
[Terminal]
OTGui
[Terminal]
[SALOME 7.7.0 - [...
[Calculatrice]
DocumentationG...

OpenTURNS Trainings

EDF

- « Uncertainty Management : Open TURNS » : 3 days
  - next session : 5-7/09/2016, ITECH ref : 4889
- « Uncertainty Management : Methodology » : 3 days
  - next session : 14-16/09/2016, ITECH ref : 4888
- contact : Corine Tripet : 01 47 65 58 41

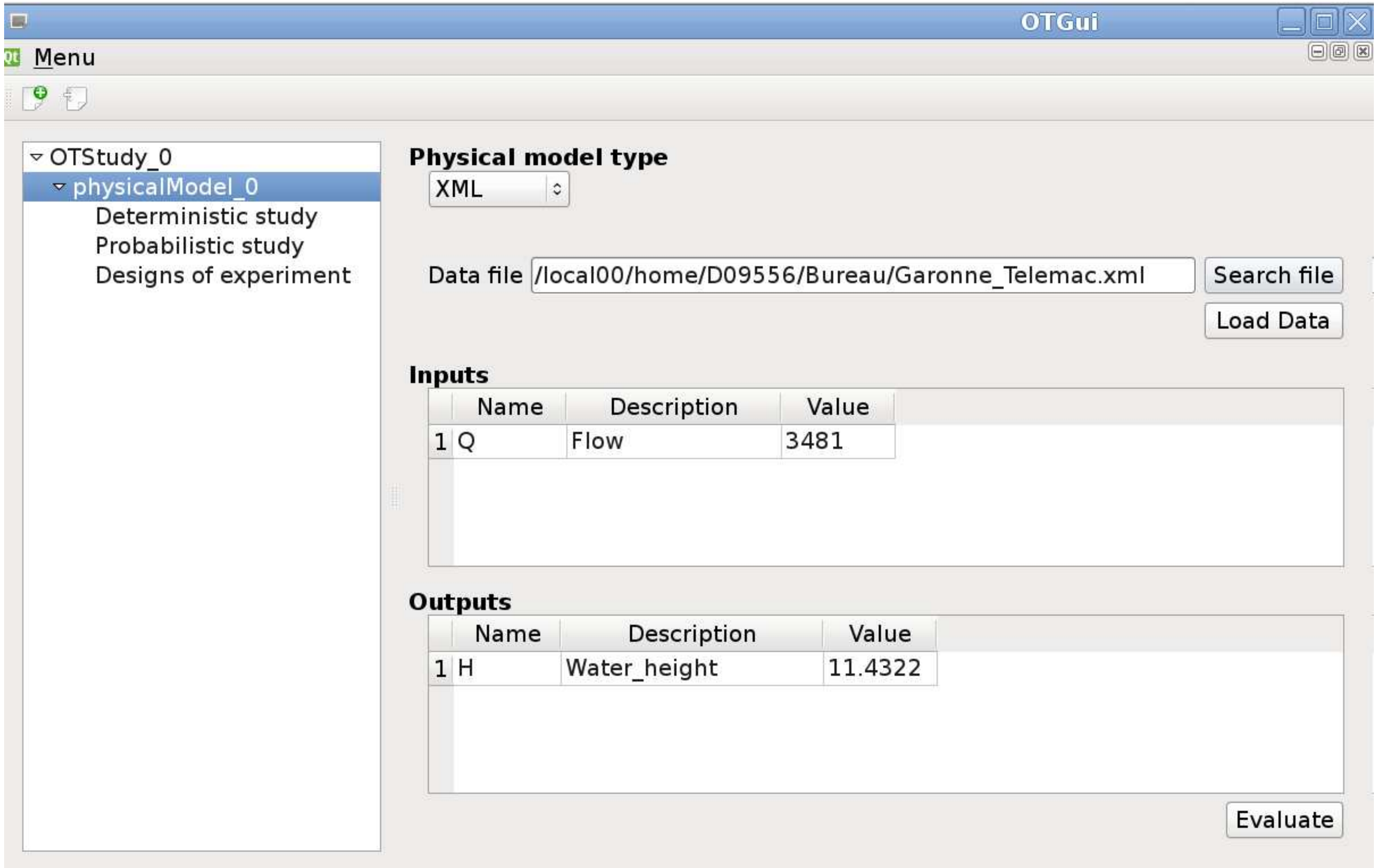
PRACE

- 1 annual session at « Maison de la Simulation (Saclay) », 3 days
- Methodology + TP with OpenTURNS or Uranie

Phimeca

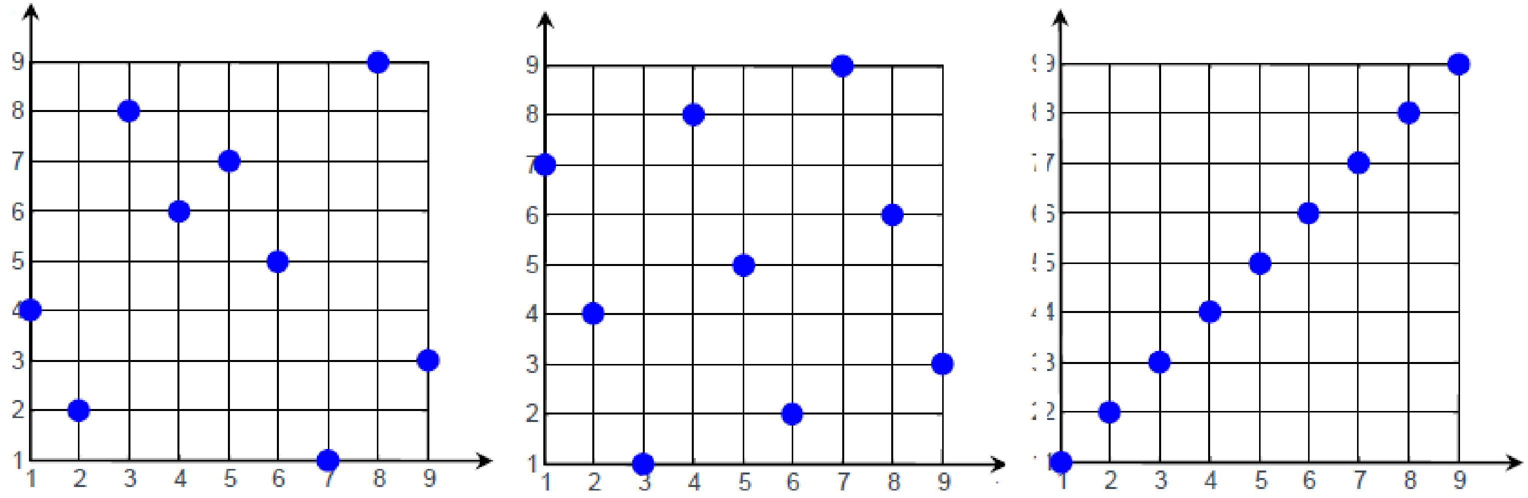
- 2 sessions each year
- www.phimeca.com

OpenTURNS Graphical User Interface



The LHS module

- **Goal : create Latin Hypercube Sampling (LHS) designs with good space filling properties**



- **What for ?**
  - **Model exploration**
  - **Central tendency : estimate mean, standard variation**
  - **Create meta-models, e.g. Polynomial chaos, Kriging, etc…**
- **In this module :**
  - **Types of designs : centered or randomized**
  - **Criteria : C2, mindist, φp**
  - **Algorithms : simulated annealing (or Monte-Carlo)**
  - **Limitation : independent copula**

The LHS module

```
>>> import openturns as ot
>>> import otlhs
# Fix the uniform bounds (0 ,1)^2, size = 100
>>> Bounds = ot.Interval(2)
>>> lhs = otlhs.LHSDesign (bounds, 100)
# Fix criterion
>>> crit_sf = otlhs.SpaceFillingPhiP ()
# Defining a temperature profile (10,0.95,2000)
>>> temp_prof = otlhs.GeometricProfile ()
>>> algo = otlhs.SimulatedAnnealingLHS (lhs , temp_prof , crit_sf )
>>> result = algo.generate ()
# Retrieve optimal design
>>> design = result.getOptimalDesign()
```