

PERSALYS, the graphical interface of OpenTURNS

M. Baudin ¹ T. Delage ¹ A. Dumas ² A. Dutfoy ¹
G. Garcia ² A. Geay ¹ O. Mircescu ¹ A. Ladier ²
J. Schueller ² T. Yalamas ²

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

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

19 June 2020, OpenTURNS User's day



Contents

Overview

PERSALYS v8: Calibration

Coupling tools

Website

What's next ?

Bring Uncertainty Methodology to Engineers

- ▶ 5 years ago
 - ▶ EDF R&D wants to maximize the use of OpenTURNS® by its engineer/researcher (and improve an existing GUI) => develop a GUI to make more easy to use
 - ▶ Phimeca has already developed an "OpenTURNS GUI" (PhimecaSoft®) which satisfy some needs of EDF R&D but not all.
 - ▶ EDF R&D and Phimeca decide to start a specific partnership in order to develop a new GUI based on OpenTURNS® and "Salome Tools": Paraview, Yacs, ...
- ▶ Persalys is available, on Salome website, in EDF Specific Salome version and commercialized by Phimeca

Some expectations regarding the GUI

- ▶ As easy to use as possible and, when it is possible, a GUI which can guide the user
- ▶ Possibility to use it inside Salome Platform to
 - ▶ Use supercomputing resources (e.g. Gaïa, 3 052 Tflops peak, 41 000 cores)
 - ▶ Connect to EDF numerical code users (Code_Aster for example)
- ▶ Take benefit from the advanced visualization capability from Paraview
- ▶ Drive the GUI from a python script usable in an "expert" mode

PERSALYS, the graphical user interface of OpenTURNS

- ▶ Main goal : provide a graphical interface of OpenTURNS in the SALOME integration platform
- ▶ Features
 - ▶ Uncertainty quantification : definition of the probabilistic model (including dependence), distribution fitting (including copulas), physical model with vector input and vector output or 1D Fields, central tendency, sensitivity analysis, probability estimate, metamodeling (polynomial chaos, kriging), screening (Morris), optimization, design of experiments
 - ▶ 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 (free) : SALOME_EDF in the "CONTRIBUTIONS" section since 2018 on <https://www.salome-platform.org>

Calibration

Given a physical model H , observed inputs x , observed outputs y we can calibrate θ so that

$$y_i = H(x_i, \theta) + \epsilon$$

where ϵ is a random variable.

Calibration outputs:

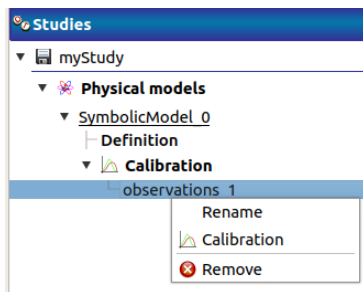
- ▶ the optimal value θ^* ,
- ▶ the distribution of θ^* ,
- ▶ the distribution of the residuals $r_i = y_i - H(x_i, \theta^*)$.

This allows to get confidence intervals of θ^* .

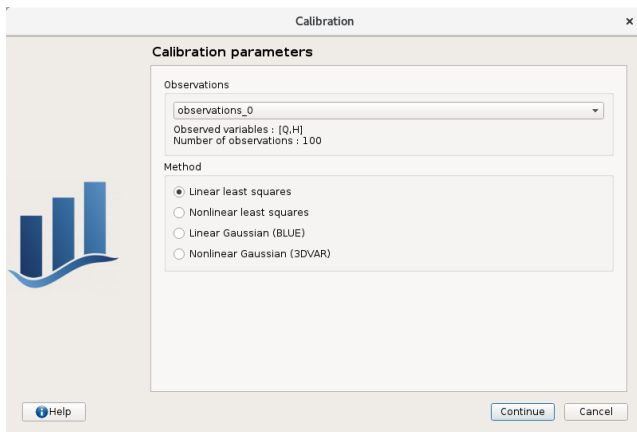
Calibration

In PERSALYS, this requires:


- ▶ a physical model H ,
- ▶ a data file containing the observed inputs and outputs.



Calibration



Calibration



Calibration


Choose the input variables to calibrate and define the reference point of the algorithm

Number of inputs to calibrate :

	<input checked="" type="checkbox"/> Name	Description	Value
1	<input checked="" type="checkbox"/> Ks	Strickler ($m^{1/3}/s$)	20
2	<input checked="" type="checkbox"/> Zv	Côte de la rivière en aval (m)	49
3	<input checked="" type="checkbox"/> Zm	Côte de la rivière en amont (m)	51

[Help](#)[Back](#)[Continue](#)[Cancel](#)

Calibration



Calibration

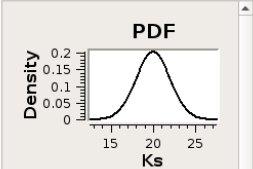
Gaussian prior distribution

Define the covariance matrix of the parameters θ to calibrate.

Marginals

Dependence

	Variable	Distribution
1	Ks	Normal
2	Zv	Normal
3	Zm	Normal



Parameters

μ

σ

Help

Back

Continue

Cancel

Calibration

Calibration

Output

H

θ

Prediction

Parameters

Model

Optimal

PDF

Optimal θ

Input	Value	Confidence interval at 95%
Ks	20.0432	[15.5523, 24.5341]
Zv	47.7043	[39.7458, 55.6628]
Zm	52.4036	[44.4431, 60.3641]

Calibration



Calibration



Calibration



Coupling with external code

A new physical coupling dialog was created:

- ▶ Execute any computer code from system e.g. with command line statement
- ▶ Create a pipeline of commands
- ▶ Exchange data with input and output files
- ▶ Manage input and output cache to save simulations
- ▶ Can be parallel : multithread or distributed (in SALOME)
- ▶ Makes the tuning of input template file and the reading of output files easy

Coupling with external code



Coupling with external code

A diff can be generated between the input template and the actual input file.

The screenshot shows the Persalys software interface. The main window is titled 'Modèle de couplage'. It has a sidebar on the left with a tree view showing 'Etudes' > 'Etude_0' > 'Modèles physiques' > 'ModèleCouplage_0' > 'Définition' > 'Modèle probabiliste'. The main area has tabs for 'Définition', 'Différentiation', and 'Résumé'. The 'Définition' tab is selected.

Under 'Définition', there is a 'Commande 1' section with a '+' icon. Below it, there is a table with columns: 'Nom', 'Description', 'Marqueur', 'Valeur', and 'Format'.

Nom	Description	Marqueur	Valeur	Format
1 E		@E	2	()
2 F		@F	30000	()
3 I		@I	1	()
4 L		@L	5	()

Below the table, there is a section 'Comparer fichier modèle' with a '+' icon. It shows a comparison of two XML files. The left file is the template, and the right file is the actual input file. The comparison highlights differences in the 'inputs' and 'derivates' sections.

```

<?xml version="1.0"?>
<beam>
  <description name="beam" title="UseCase beam with XML input file" version="1.0" date="2014-04-07">
    <tool name="beam exe" version="1.0"/>
  </description>
  <inputs F=" @F" E=" @E" L=" @L" I=" @I"/>
  <computation>
    <derivate activate="on"/>
    <hessian activate="off"/>
  </computation>
</beam>

<?xml version="1.0"?>
<beam>
  <description name="beam" title="UseCase beam with XML input file" version="1.0" date="2014-04-07">
    <tool name="beam exe" version="1.0"/>
  </description>
  <inputs F=" 30000" E=" 2" L=" 5" I=" 1"/>
  <computation>
    <derivate activate="on"/>
    <hessian activate="off"/>
  </computation>
</beam>
  
```

At the bottom, there is a 'Paramètres avancés' section with a '+' icon and a 'Vérifier le modèle' button.

Coupling with external code

Resources are files to copy to run the simulation : mesh, parameters, etc...



Coupling with external code

Persalys

Fichier Vue Aide

Etudes

- Étude_0
 - Modèles physiques
 - ModèleCouplage_0
 - Définition
 - Modèle probabiliste

Modèle de couplage

Définition Différentiation Résumé

Commande 1

Commande Entrée Ressource **Sortie** Additional processing

Fichier

Fichier de sortie _beam_outputs_xml

	Nom	Description	Marqueur	Saut de marqueur	Saut de ligne	Saut de colonne	Valeur
1	deviation	deviation="	0	0	0	625000	

Vérifier sortie Ajouter Supprimer

deviation=0.130208

Paramètres avancés

Fichier d'entrée de cache

Fichier de sortie de cache

Répertoire de travail

☒ Garder répertoire de travail

☐ Vérifier le modèle

Vider le cache

Coupling with external code

On-disk cache management: one input point X is only evaluated once.

Cache ▾

Cache input file ...

Cache output file ...

Website

`persalys.fr` : download (source, binaries), doc (videos tutorials), news

Persalys

Persalys est une interface graphique dédiée au **traitement des incertitudes et la gestion des variabilités**. Le logiciel est un outil à la frontière de la simulation numérique, des analyses probabilistes et des data sciences.

Persalys permet de :

- créer des **modèles mathématiques**: analytique, couplage avec un modèle externe (éléments finis, ...), FMU
- **analyser la variabilité** de vos paramètres grâce à un grand nombre de méthodes et d'outils de visualisation
- **analyser vos données de mesures** statistiquement, inférer des distributions de probabilités ou encore créer des métamodèles

Dernières actualités

10/04/2020 : Développement personnalisé : Penstock Pipe

24/02/2020 : Sortie de Persalys 8.0

[Voir les actualités](#)

Qui utilise Persalys ?


Solutions for nuclear engineering

Développement et étude

Qui développe Persalys ?

Persalys est développé en partenariat par Phimeca et EDF. Il est codé en C++ et Qt avec principalement la librairie OpenTURNS comme moteur de calcul.



What's next ?

PERSALYS Roadmap :

- ▶ 2D Fields, 3D Fields
- ▶ In-Situ fields based on the MELISSA library (with INRIA): when we cannot store the whole sample in memory or on the hard drive, update the statistics (e.g. the mean, Sobol' indices) sequentially, with distributed computing.
- ▶ Linear regression.



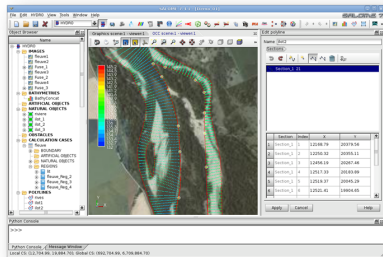
The end

Thanks !

Questions ?

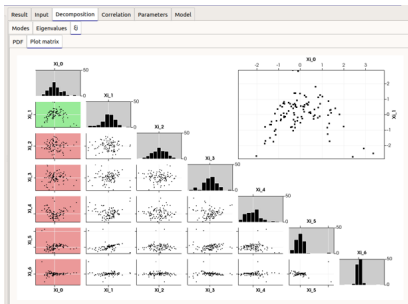
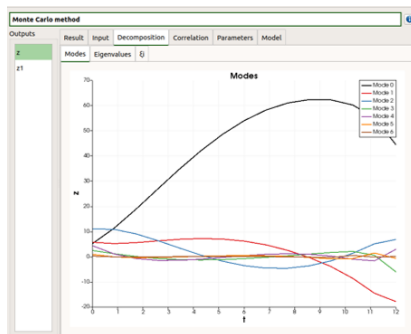
SALOME

- ▶ 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



PERSALYS: 1D fields

- ▶ Karhunen Loeve decomposition
- ▶ Show modes, eigenvalues and projection coefficients

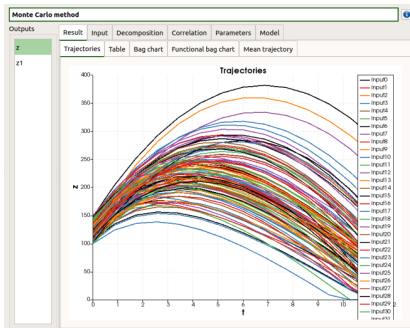
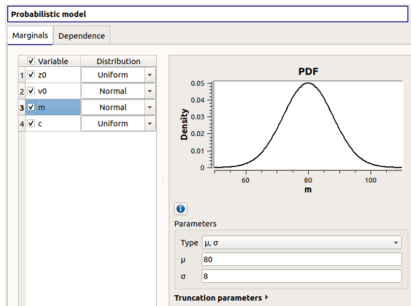


Interactive uncertainty visualization with Paraview



1D fields

- ▶ Probabilistic model
- ▶ Uncertainty propagation with simple Monte-Carlo sampling



1D fields

- ▶ BagChart and Functional Bagchart (from Paraview) based on High Density Regions (Hyndman, 1996).
- ▶ To do this, Paraview uses a principal component analysis decomposition.
- ▶ Linked and interactive selections in the views.

