

The OpenTURNS uncertainty quantification library

Michaël Baudin ¹ Julien Schueller 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,
schueller@phimeca.com

EuroSciPy 2017, Erlangen, Germany



Contents

Uncertainty Quantification Methodology

The OpenTURNS library

Example

Software architecture

Modules

Bibliography

Uncertainty Quantification Methodology



OpenTURNS

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



OpenTURNS

Overview :

- ▶ First release: 2007
- ▶ technical committee (including 4 developers), steering committee
- ▶ Users: mainly in France
- ▶ Number of users: ≈ 1000 (10900 Conda downloads in 2016-2017)
- ▶ Size of the project (2017): 5420 files, 680 classes, 158k sloc
- ▶ Documentation: theoric, API, developer

Features overview 1/2

Data analysis:

- ▶ Visual analysis
- ▶ (Non)parametric estimation
- ▶ Distribution fitting tests
- ▶ Bayesian calibration

Probabilistic modeling:

- ▶ Dependence modeling: Copulas
- ▶ Univariate & multivariate distributions
- ▶ Process: ARMA, Gaussian

Features overview 2/2

Meta modeling:

- ▶ Gaussian process regression (Kriging)
- ▶ Spectral methods: Functional chaos expansion, Karhunen-Loeve, low-rank tensors

Reliability, sensitivity:

- ▶ Monte Carlo, LHS, low-discrepancy sequences
- ▶ Variance reduction methods: importance sampling, subset sampling
- ▶ Approximation methods: FORM, SORM
- ▶ Indices: Spearman, Sobol, ANCOVA

Flood example 1/5

The flood model of a river compares the water level to the dike height:

$$S = \left(\frac{Q}{K_s \times 300 \times \sqrt{(Z_m - Z_v)/5000}} \right)^{3/5} + Z_v - 55.5 - 3$$



Flood example 2/5

- ▶ $Q \sim \text{Gumbel}(\alpha=0.00179, \beta=1013)$, flow rate [m^3s^{-1}]
- ▶ $K_s \sim \text{Normal}(\mu=30.0, \sigma=7.5)$, strickler [$m^{1/3}s^{-1}$]
- ▶ $Z_v \sim \text{Uniform}(a=49, b=51)$, downstream depth [m]
- ▶ $Z_m \sim \text{Uniform}(a=54, b=56)$, upstream depth [m]

Failure occurs when S is positive, lets estimate $P_f = \mathbb{P}(S(\underline{X}) > 0)$.

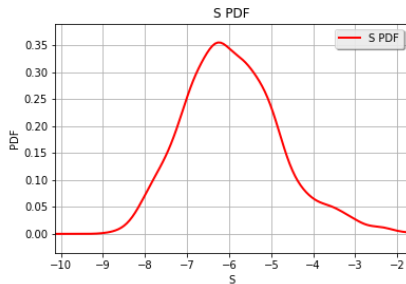
Flood example 3/5

```
# Probabilistic model
Q = ot.Gumbel(1./558., 1013.)
Ks = ot.Normal(30.0, 7.5)
Zv = ot.Uniform(49.0, 51.0)
Zm = ot.Uniform(54.0, 56.0)
copula = ot.IndependentCopula(4)
dist = ot.ComposedDistribution([Q, Ks, Zv, Zm], copula)
```



Flood example 4/5

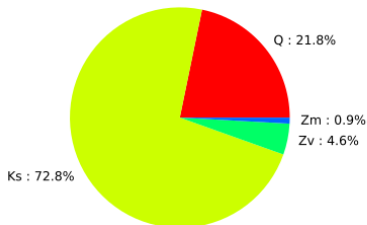
```
# Random variable S=G(Q,Ks,Zv,Zm)
rv = ot.RandomVector(dist)
f = ot.SymbolicFunction(['Q', 'Ks', 'Zv', 'Zm'],
    ['(Q/(Ks*300.*sqrt((Zm-Zv)/5000)))^(3.0/5.0)+Zv-55.5-3.')]
S = ot.CompositeRandomVector(f, rv)
```



Flood example 5/5

```
# Compute  $P(S>0)$  using First Order Reliability Method algorithm
event = ot.Event(S, ot.Greater(), 0.0) # event  $S>0$ 
optimAlgo = ot.Cobyla()
algo = ot.FORM(optimAlgo, event, dist.getMean())
algo.run()
result = algo.getResult()
print('Pf=', result.getEventProbability()) #  $P(S>0)=0.00053$ 
result.drawImportanceFactors()
```

Importance Factors from Design Point - overflow



Architecture

- ▶ C++ core / SWIG Python bindings
- ▶ BLAS / LibXml2 / TBB / muParser / NLOpt
- ▶ Packaging: Debian, Conda, Windows
- ▶ Compilation: cmake
- ▶ Documentation: Sphinx
- ▶ Repository: <https://github.com/openturns>

Modules

- ▶ otrobopt: Robust optimization module
- ▶ otfmi: FMI models manipulation
- ▶ otsvm: SVM classifiers, metamodels
- ▶ otagrum: Bayesian networks module
- ▶ otwrapy: wrap external simulation codes
- ▶ ...

Contributions welcome!

END

Thank you for your attention!
Any questions?



Bibliography

- ▶ Airbus, EDF, Phimeca Engineering, IMACS. OpenTURNS, a scientific library usable as a Python module dedicated to the treatment of uncertainties, www.openturns.org.
- ▶ Airbus, EDF, Phimeca Engineering, IMACS. Documentation of OpenTURNS, version 1.9.
<http://openturns.github.io/openturns/1.9/contents.html>
- ▶ Michaël Baudin, Anne Dutfoy, Bertrand Iooss, and Anne-Laure Popelin. OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation, Handbook of Uncertainty Quantification, pages 1-38. Springer International Publishing, 2016
- ▶ Open TELEMAC-MASCARET. Electricité de France, Sogreah, Hydraulic Research Wallingford, Centre d'Etudes Techniques Maritimes et Fluviales, Bundesanstalt für Wasserbau, and Daresbury Laboratory. www.opentelemac.org.