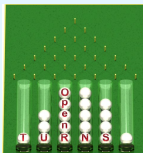


OpenTURNS Developer training: first steps

Trainer : Sofiane Haddad
Airbus
sofiane.s.haddad@airbus.com

Developers training



OpenTURNS: first steps

1 Navigation in the source code

2 Library development

3 Module development

Navigation in the source code

The Uniform distribution

- Locate the class within the library source code;
- Follow its inheritance graph in order to explore the Bridge pattern;
- Locate the associated regression test;
- Execute the test;
- Locate its SWIG interface file and its associated Python module;
- Execute the associated python test.

Library development 1/2

Projects

- ① Weighted interpolation: create a new Evaluation that interpolate between points of a Sample using weighted kernel interpolation.

$$f(\underline{x}) = \alpha \sum_{i=0}^{N-1} K(\underline{x} - \underline{X}^i) \underline{Y}^i \quad (1)$$

where K is a given Distribution and α is a normalization factor.

- ② CloudMesher: mesh generation over a cloud of points using kernel mixture, pca, rotation, then levelset mesher on an interval
- ③ ClenshawCurtis 1-d integration:
<https://people.maths.ox.ac.uk/trefethen/cctalk.pdf> +
ClenshawCurtisProductExperiment
- ④ TawnCopula/JoeCopula/MaxCopula 2-d copulas
- ⑤ Add some statistical tests (Levene test for example), Anderson-Darling for Weibull/lognormal?

Library development 2/2

Projects

- ⑤ Extend archimedean copulas from 2-d to n -d (FarlieGumbelMorgenstern, FrankCopula, ClaytonCopula, AliMikhailHaqCopula)
- ⑥ ArchiMaxCopula, see ExtremeValueCopula
- ⑦ Extend ComposedDistribution to accept a list of n -d distributions
- ⑧ Add a method to SobolSimulationAlgorithm to draw sobol indices (openturns/issues/1001)

Module development

Projects

- 1 Convert the IntegralCompoundPoisson distribution python module into an OpenTURNS C++ module
- 2 Create a RandomVector distributed uniformly over an n dimensional sphere/ball/simplex;
- 3 Create a sensitivity analysis module that implements the Csizar indices;