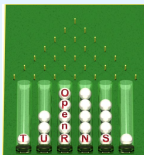


# OpenTURNS Developer training: first steps with OpenTURNS

Trainer : Régis LEBRUN, EADS/IW/SE/AM [regis.lebrun@eads.net](mailto:regis.lebrun@eads.net)

Developers training





## 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.

## Projects

- 1 Implementation of a BoundConstrainedAlgorithm using L-BFGS-B. The module must adopt the GPL license. The objective is to solve:

$$\min_{\underline{a} \leq \underline{x} \leq \underline{b}} f(\underline{x}) \quad (1)$$

where  $f \in \mathcal{F}(\mathbb{R}^n, \mathbb{R})$

- 2 Interface of the kissFFT library to convert a Collection< NumericalComplex > into another Collection< NumericalComplex > using direct or inverse Discrete Fourier Transform (DFT or iDFT):

$$DFT(\underline{x})_j = \sum_{k=0}^{N-1} x_k \exp(2i\pi jk/N) \quad iDFT(\underline{x})_j = \frac{1}{N} \sum_{k=0}^{N-1} x_k \exp(-2i\pi jk/N) \quad (2)$$

- 3 Weighted interpolation: create a new NumericalMathEvaluation that interpolate between points of a NumericalSample using weighted kernel interpolation.

$$f(\underline{x}) = \alpha \sum_{i=0}^{N-1} K(\underline{x} - \underline{X}^i) \underline{Y}^i \quad (3)$$

where  $K$  is a given Distribution and  $\alpha$  is a normalization factor.

## Projects

- 4 Add a drawing capability to the NumericalMathFunction in the 1D and 2D cases (take inspiration from the drawPDF() method of the DistributionImplementation class)
- 5 Convert the IntegralCompoundPoisson distribution python module into an OpenTURNS C++ module
- 6 Create a new Drawable using the pairs command of R.
- 7 Create an Event that check a point against an Interval.
- 8 Create a RandomVector distributed uniformly over an  $n$  dimensional sphere;
- 9 Create a RandomVector distributed uniformly over an  $n$  dimensional ball;
- 10 Create a RandomVector distributed uniformly over an  $n$  dimensional simplex;

The projects 2, 3 and 5 can be written in pure Python, without using the OpenTURNS module infrastructure.