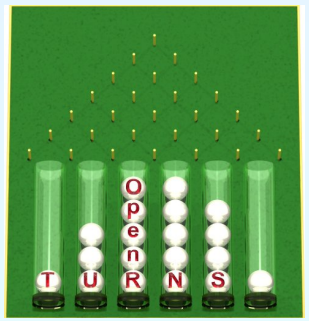
# OpenTURNS Developer training: first steps

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

Developers training



# OpenTURNS: first steps

# 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.

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

   where $K$ is a given Distribution and $\alpha$ 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 archimedian 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 sensitiviry analysis module that implements the Csizar indices;