OpenTURNS release highlights

J. Schueller (Phimeca)

User Day #17, June 14th 2024, EDF Lab

New features since last year in releases:
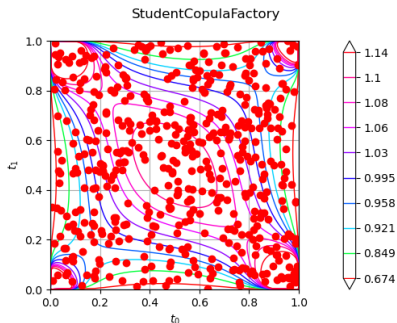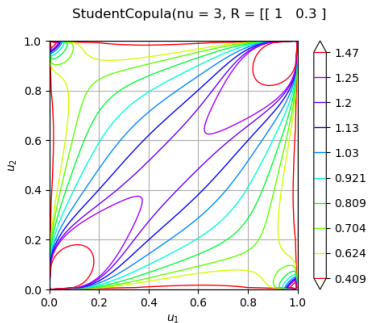
- v1.22: fall 2023
- v1.23: spring 2024

Contents

Not about

- TruncatedOverMesh
- BoundaryMesher
- UniformOrderStatistics
- New GEV/GPD estimators

New distribution: Student / t-Copula

- StudentCopula(nu, R): The distribution
- StudentCopulaFactory: Its estimator (xref "Interval Estimation for Bivariate t-Copulas via Kendall's Tau")
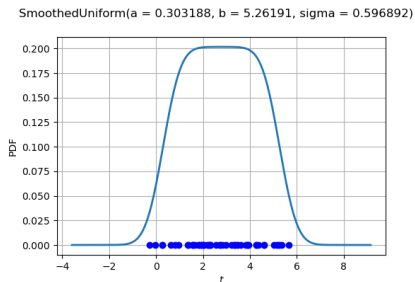


```
# OT < 1.23
R = ot.CorrelationMatrix(dim, [1.0, 0.5, 0.5, 1.0])
copula = ot.Student(3.5, [0.0] * 2, [1.0] * 2, R).getCopula()
```

New distribution estimator for SmoothedUniform

- Method of moments for initialization
- Maximum likelihood for the final estimator

SmoothedUniform(a = 0.303188, b = 5.26191, sigma = 0.596892)

```
import openturns.experimental as otexp
estimated = otexp.SmoothedUniformFactory().build(sample)
```

Latent variable model

- Covariance model, for eg Kriging
- Covariance between different unordered values (or levels) of a categorical variable
- Parameters: coordinates in the latent space
- Zhang2020: "A latent variable approach to Gaussian process modeling with qualitative and quantitative factors"

```
import openturns.experimental as otexp
covModel = otexp.LatentVariableModel(3, 2)
activeCoordinates = [0.1, 0.3, -0.4]
covModel.setLatentVariables(activeCoordinates)
```

New Field metamodeling capabilities

- Vector to field metamodeling and sensitivity using KL + chaos
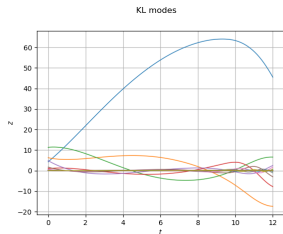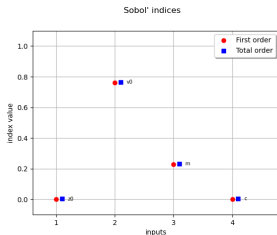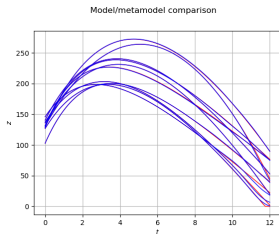

```
# metamodel
algo = otexp.PointToFieldFunctionalChaosAlgorithm(X, Y, distribution)
algo.run(); result = algo.getResult()
metaModel = result.getPointToFieldMetaModel()

# sensitivity
sensitivity = otexp.FieldFunctionalChaosSobolIndices(result)
s1 = sensitivity.getFirstOrderIndices()
st = sensitivity.getTotalOrderIndices()
```

Rank-based Sobol' indices

- Data-driven (no need for dedicated design of experiments, just iid design)
- Only first-order indices
- Gamboa2022: "Global sensitivity analysis: A novel generation of mighty estimators based on rank statistics"
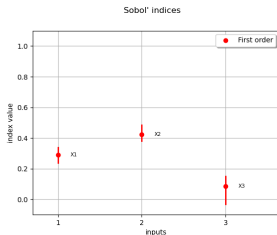

```
X = distributionX.getSample(N)
Y = f(X)
algo = otexp.RankSobolSensitivityAlgorithm(X, Y)
s1 = algo.getFirstOrderIndices()
```

Documentation improvements

- Lots of new examples: chaos, cv, regression, MLE, functions, integration, enumerate, …
- New usecases: fire satellite, wing weight, Linthurst/Coles datasets
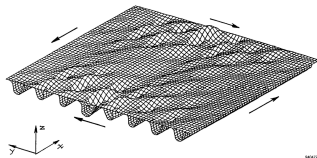
- Example minigalleries linking to relevant examples
- Automatic checking of every internal links
- Lot of time invested in the improvement of the documentation

- Multidimensional integration using cuba library (CubaIntegration)
- New class for integration from an existing design of experiment (ExperimentIntegration)
- Faster KDTree implementation using nanoflann library
- Faster TruncatedDistribution with n-d CDF inversion

Python channels

- Pip (and uv), Conda
- Versions: 3.8-3.12
- OS: Windows, Linux, MacOS
- Architectures: x86_64, arm64 (MacOS-only)

Supported Linux distributions

- Ubuntu 22/24
- Debian 11/12
- Fedora 39/40
- CentOS 8
- OpenSUSE 15.5/15.6
- Mageia 8
- ArchLinux

... and FreeBSD

2024-2025 work

- Conditional distributions / Bayesian
- Quantiles estimation / tolerance intervals
- Calibration (functional models, bound constraints)
- New GPR API
- LOLA-Voronoi sequential design
- Cross-validation of functional chaos expansion

Thank you for your attention!
Any questions?