OpenTURNS release highlights

J. Schueller (Phimeca)

User Day #18, June 13th 2025, EDF Lab

Overview

New features since last year in releases:

- v1.24: fall 2024
- v1.25: spring 2025

Contents

Not about

- New conditional modelling features
- New quantile estimation features
- New GP API

Line sampling simulation 1/3

Contract EDF with E. Ardillon, A. Ajenjo, J. Mure with Phimeca

- Koutsourelakis, H. Pradlwarter, G. Schueller, *Reliability of structures in high dimensions, part i: algorithms and applications*, Probabilistic Engineering Mechanics 19 (4) (2004) 409–417
- Angelis M., Patelli E., Beer M., *Advanced line sampling for efficient robust reliability analysis*, Structural safety, 52 :170-182, 2015.

Line sampling simulation 3/3

```
# 1. preliminary FORM
optim = ot.Cobyla()
optim.setStartingPoint(X.getMean())
algo = ot.FORM(optim, event)
algo.run()
resultFORM = algo.getResult()
alpha = resultFORM.getStandardSpaceDesignPoint()

# 2. LS
alpha = [0.0, 0.0, 0.0, 0.0, 1.0]
rootStrategy = ot.SafeAndSlow(ot.Brent(1e-3, 1e-3, 1e-3, 5), 8, 0.01)
algo = otexp.LineSampling(event, alpha, rootStrategy)
algo.setMaximumOuterSampling(2000)
algo.setMaximumCoefficientOfVariation(5e-2)
algo.setSearchOppositeDirection(False)
algo.setAdaptiveImportantDirection(True)
algo.run()
result = algo.getResult()
pf = result.getProbabilityEstimate()
```

Contract Airbus with R. Lebrun with Phimeca

Crombecq, K., *Surrogate Modelling of Computer Experiments with Sequential Experimental Design*, PhD thesis, Universiteit Gent, Belgium, 2011.

- Exploration criterion based on Voronoi cell volume:

$$v(\boldsymbol{p}_i) = \int_{\boldsymbol{x} \in \mathcal{V}_i} \mu_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = \mathbb{E}\left[1_{\mathcal{V}_i}(\boldsymbol{X})\right]$$

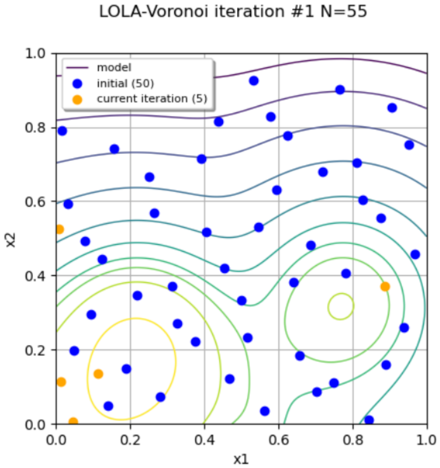
- Exploitation criterion based on a measure of the nonlinearity of the model:

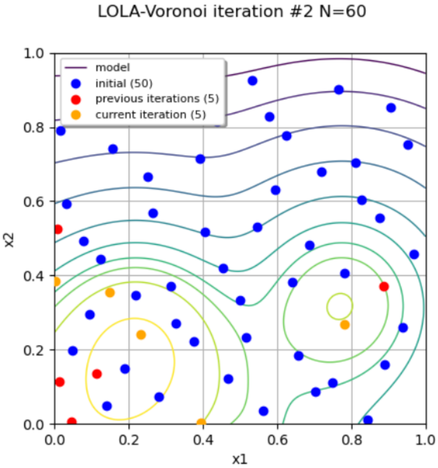
$$e_k(\boldsymbol{p}_r) = \sum_{i=1}^{m} \left| g_k(\boldsymbol{p}_{ri}) - (g_k(\boldsymbol{p}_r) + \boldsymbol{J}_i(\boldsymbol{p}_{ri} - \boldsymbol{p}_r)) \right|$$

Combined into hybrid score:

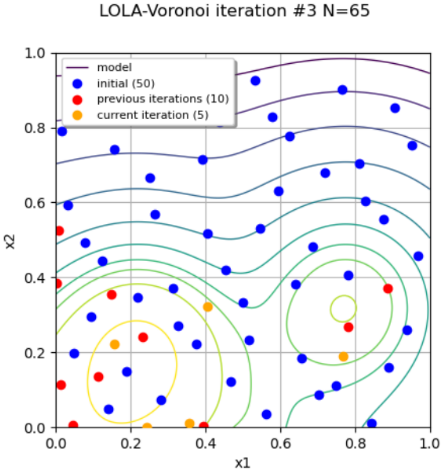
$$h(\boldsymbol{p}_i) = \lambda v(\boldsymbol{p}_i) + (1-\lambda) \frac{e(\boldsymbol{p}_i)}{\sum_{j=1}^{n} e(\boldsymbol{p}_j)} \forall i \in [1, n]$$

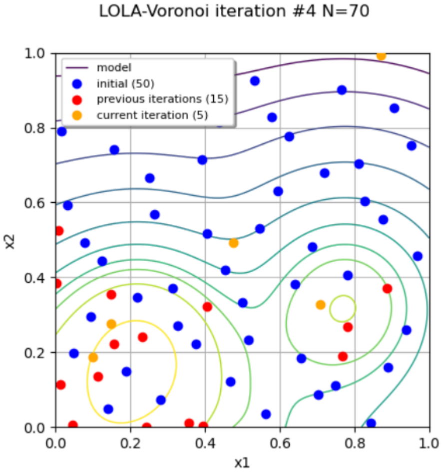
LOLA-Voronoi sequential design 2/7



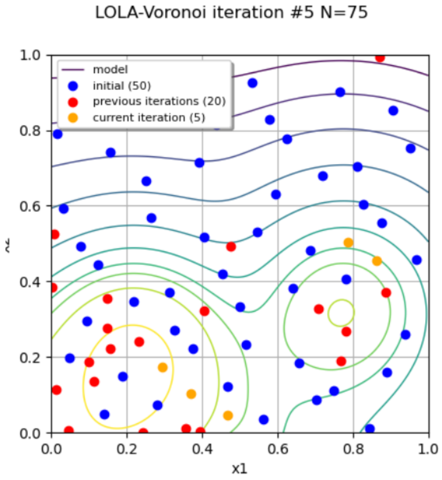


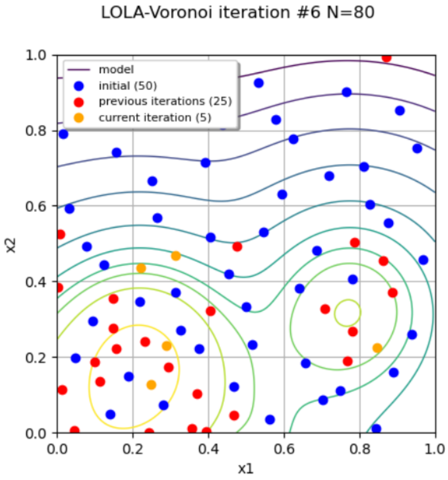
LOLA-Voronoi sequential design 3/7



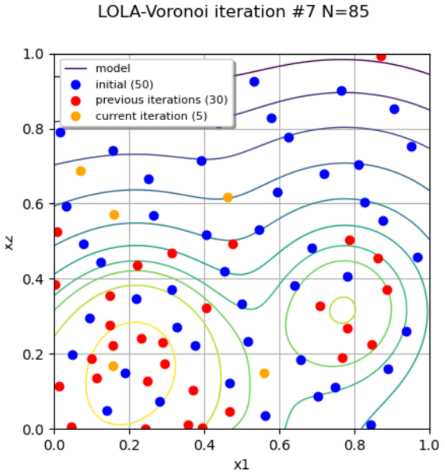


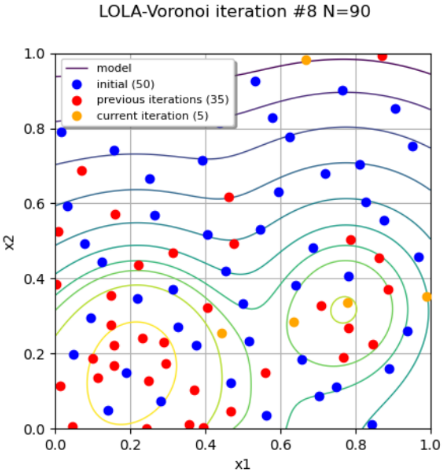
LOLA-Voronoi sequential design 4/7

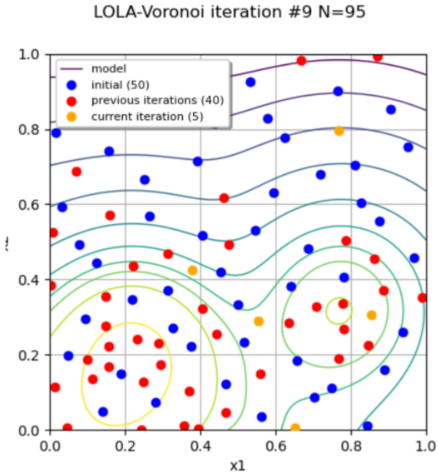


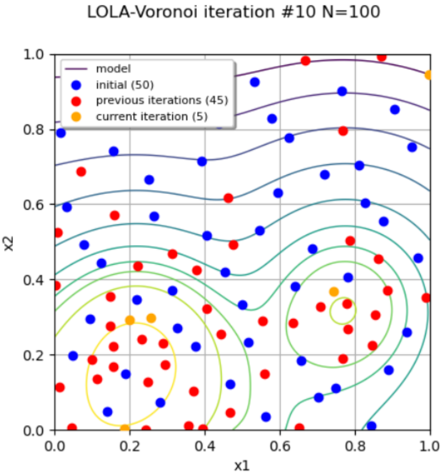


LOLA-Voronoi sequential design 5/7






LOLA-Voronoi iteration #9 N=95
model
initial (50)
previous iterations (40)
current iteration (5)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x1


LOLA-Voronoi iteration #10 N=100
model
initial (50)
previous iterations (45)
current iteration (5)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x1
x2

LOLA-Voronoi sequential design 7/7

```
# 1. initial design
distribution = ot.JointDistribution([ot.Uniform(-1.0, 1.0)] * 2)
N = 50
x0 = ot.LowDiscrepancyExperiment(ot.HaltonSequence(), distribution, N).generate()
y0 = f1(x0)

# 2. sequential experiment
algo = otexp.LOLAVoronoi(x0, y0, distribution)
for i in range(10):
    x = algo.generate(5)       # generate 5 new input samples
    y = f(x)                   # evaluate output samples
    algo.update(x, y)          # update state with new x/y pairs

# 3. learn metamodel on all samples
x_final = algo.getInputSample()
y_final = algo.getOutputSample()
algo_mm = ot.FunctionalChaosAlgorithm(x_final, y_final, distribution)
algo_mm.run()
metamodel = algo_mm.getResult().getMetaModel()
```

New distribution estimator for SmoothedUniform

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

```
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

Other improvements

- 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

Packaging 1/2

Python channels

- Pip / Conda
- Versions: 3.9+
- OS: Windows, Linux, MacOS
- Architectures: x86_64, arm64 (Linux+MacOS)

Packaging 2/2

Supported Linux distributions

- Ubuntu 22/24
- Debian 11/12
- Fedora 41/42
- CentOS 8
- OpenSUSE 15.6
- Mageia 9
- ArchLinux

... and FreeBSD

Outlook

2025-2026 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

END

Thank you for your attention!
Any questions?