The Journal of Open Source Software

OpenTURNS: A Python package for uncertainty quantification

Michaël Baudin[1], **Anne Dutfoy**[1], **Régis Lebrun**[2], **Julien Schueller**[3], **Sofiane Haddad**[2], **Loïc Brevault**[4], **Mathieu Balesdent**[4], **Julien Pelamatti**[1], **and Joseph Muré**[1]

1 EDF R&D, France **2** Airbus, France **3** Phimeca, France **4** ONERA, France

DOI: 10.21105/joss.10484

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @abhijit-c
- @dannys4
- @tupui

Submitted: 25 February 2026
Published: 02 September 2026

Summary

OpenTURNS (www.openturns.org) is an open-source Python package for probabilistic modeling, uncertainty quantification (UQ), and machine learning. It provides a unified interface, granting users access to a comprehensive set of state-of-the-art algorithms and modeling tools. The library is supported by extensive, regularly updated, easy to read documentation, including introductory examples, various use cases, tutorials, and user guides, enhancing its accessibility. OpenTURNS integrates seamlessly with NumPy, Pandas, and other scientific Python packages, allowing users to develop UQ workflows and generate reproducible results. Its modular architecture enables both newcomers and advanced users to access reliable, well-tested methods for uncertainty propagation, sensitivity analysis, and reliability assessment. Its active community provides additional support and resources through a Discourse forum (*OpenTURNS's Forum*, 2020) and a website. Stable releases are published on a semiannual basis, and the library is distributed through Conda, pip, and Debian repositories and available on Linux, Windows, and macOS.

Statement of need

UQ plays a critical role in robust decision-making during the design and operation of complex systems. It encompasses probabilistic modeling (multivariate distributions, stochastic processes), sampling, functional algebra, uncertainty propagation, sensitivity analysis, reliability, optimization, inverse problems, and surrogate modeling. Validated and well-documented methods are essential for teaching, academic research, and industrial applications. However, many probabilistic modeling libraries are domain-specific—focusing on fields such as mechanical engineering, computational fluid dynamics, or hydrology—which limits the sharing of best practices. This underscores the need for a general-purpose, cross-disciplinary framework for probabilistic modeling.

State of the field

There are many software packages and libraries for uncertainty quantification. Although the landscape is highly fragmented with significant specialization, these tools fall into two broad categories. First, there are general-purpose libraries, which aim to provide most of the features used in research and engineering to perform UQ studies. In this class of tools, we find Dakota (Adams et al., 2025), Raven (Rabiti et al., 2017), Sparse Grid++ (Pflüger, 2010), PSUADE (Tong, 2024), URANIE (Blanchard et al., 2019), UQLab (Marelli & Sudret, 2014), UQTk (Debusschere et al., 2017), OpenCOSSAN (Patelli, 2016), UQPy (Olivier et al., 2020), EasyVVUQ (Suleimenova et al., 2021), SmartUQ (Absil et al., 2016), and NESSUS

(McFarland & Riha, 2017), among others. These tools and libraries are provided in a variety of languages including MATLAB (e.g., UQLab), Python (e.g., UQPy), and ROOT/C++ (e.g., URANIE). Some of these tools are open-source (e.g., EasyVVUQ), while others are commercial products (e.g., NESSUS). Other libraries have become open-source over time (e.g., UQLab became open-source in 2022). Many of these libraries are massive software projects (e.g., Dakota has more than 300,000 lines of code). Some of these tools have a graphical user interface (e.g., SmartUQ), but not all. The second category comprises a large number of specialized libraries. In this class of tools, we find, for example, GPy (GPy, 2012) for Gaussian processes, QUESO (McDougall et al., 2017) for Bayesian inference, SALib (Herman & Usher, 2017) and sensitivity (Iooss et al., 2026) for sensitivity analysis and ChaosPy (Feinberg & Langtangen, 2015) for polynomial chaos expansion, among many others. Most of these tools are relatively small software projects (e.g., ChaosPy has approximately 30,000 lines of code). These specialized libraries are sometimes maintained by a small number of core developers, although some are based on many contributions (e.g., SALib has more than 40 contributors on GitHub). Furthermore, they are often easier to adapt for research purposes. This is the case, for example, for Python projects, which are easier for many users to extend compared to a C++ library. On the other hand, maintaining these libraries over time can be challenging.

Other Python packages exist for Monte Carlo sampling and other numerical methods (Virtanen et al., 2020), statistical modeling (Seabold & Perktold, 2010), machine learning (Pedregosa et al., 2011), UQ test functions (Wicaksono & Hecht, 2023), and quasi-Monte Carlo methods (P. Roy et al., 2023), among others. One of the goals of OpenTURNS is to provide a consistent interface for all these tasks, serving as a general-purpose tool for UQ.

Developing OpenTURNS as a standalone framework was necessary because integrating its specialized, tightly coupled C++ architecture for advanced reliability analysis and stochastic modeling into general-purpose libraries like SciPy or scikit-learn would have been architecturally incompatible and misaligned with their primary scopes. OpenTURNS was created in 2005 at a time when SciPy was a relatively new library (the development of SciPy was started in 2001) and scikit-learn did not exist (the first public release was in 2010). Today, this choice remains relevant as OpenTURNS is integrated into other C++ libraries such as Persalys, SALOME (see SALOME Consortium, 2026) and pSeven (see pSeven SAS, 2026), where the C++ implementation of OpenTURNS is mandatory.

Software design

OpenTURNS relies on a dual-language architecture to balance computational efficiency with user accessibility. The core library is implemented in C++, which handles memory management and heavy computational workloads required by Monte Carlo simulations, high-dimensional integration, and optimization algorithms. This C++ core is then exposed through a Python interface using SWIG.

The primary trade-off in this design is the increased complexity of maintaining a mixed-language codebase and managing bindings. However, this cost is outweighed by the performance gains in intensive numerical tasks.

From an architectural perspective, OpenTURNS is strictly object-oriented, mirroring mathematical concepts directly in the code. Users manipulate objects such as `Distribution`, `Process`, and `Function`. The `Function` class, for instance, provides an abstract representation of mathematical models, allowing for caching, composition, and automatic or finite-difference differentiation. This abstraction ensures that reliability and sensitivity algorithms can operate on complex industrial black-box models exactly as they would on simple analytical equations.

Features

OpenTURNS provides a wide range of easy to build multivariate probabilistic models: more than 60 continuous and discrete distributions, including extreme-value distributions, copula-based models (more than 15 copulas), and distribution transformations (such as truncation, conditioning, mixture, push-forward, and distribution algebra).

Multivariate distributions can be sampled using general sampling algorithms (Monte Carlo, Latin Hypercube Sampling (LHS), and optimized LHS) and quasi-Monte Carlo low-discrepancy sequences (such as Sobol' or Halton) for integration and model analysis.

OpenTURNS offers modeling features for stationary and non-stationary stochastic processes (e.g., ARMA processes and Gaussian processes in arbitrary dimensions) associated with several covariance kernels (e.g., exponential, Matérn, and squared exponential).

The library provides extensive data analysis capabilities. Dependency indicators can be computed (e.g., Pearson and Spearman correlations). Inference of parametric and non-parametric univariate or multivariate probabilistic models, including copula models, can be performed, along with several statistical tests to validate the estimation (quantitative tests such as the Kolmogorov-Smirnov or Lilliefors tests, and graphical tests such as the quantile-quantile plot).

The library provides an abstract representation of vector-valued or field-based mathematical functions that allows for composition, aggregation, symbolic or finite-difference derivatives, and caching. It also includes tools for wrapping black-box code that communicates through input/output file exchange.

Several sensitivity and reliability indices can be used to quantify the influence of one variable on another. Notable examples include Standardized Regression Coefficients (SRC), Sobol' indices, and Hilbert–Schmidt Independence Criterion (HSIC) indices. For the latter two, OpenTURNS provides the asymptotic distribution of the estimators, which allows the computation of confidence intervals.

OpenTURNS implements numerous reliability algorithms for rare-event probabilities. These include classic algorithms such as the First-Order Reliability Method, Second-Order Reliability Method, subset simulation, and other variance reduction methods. It also provides more recent methods such as Nonparametric Adaptive Importance Sampling (NAIS) and Importance Sampling by Cross-Entropy. An event can be a single occurrence (e.g., a threshold exceedance), a system event, or a condition based on a stochastic process.

For supervised learning, OpenTURNS enables the construction of surrogate models from data, such as linear regression models, polynomial chaos expansions, and Gaussian process regression. These surrogate models can also be built for models with functional inputs. The optimization can be performed on an arbitrary subset of the Gaussian process regression hyperparameters.

In many cases, probabilistic analyses use optimization algorithms (e.g., for supervised learning or calibration). OpenTURNS can be used to define a wide variety of optimization problems in a unified manner and provides users with solution algorithms from numerous existing libraries: NLopt, IPOPT, pagmo, BONMIN, dlib, Ceres, cminpack, and COBYLA, among others. It can solve optimization problems with single or multiple objectives, with equality or inequality constraints, and with continuous or discrete variables.

Since many analyses require the computation of integrals, OpenTURNS provides integration methods based on sampling or quadrature: multivariate Gaussian quadrature (e.g., Gauss–Legendre quadrature), Gauss–Kronrod, Smolyak, simplicial cubature, and other algorithms, including those from the Cuba library.

Research impact statement

OpenTURNS is widely used in research, industrial engineering, and academic education. It has been integrated into university curricula (e.g., DTU Denmark) and engineering schools (e.g., ENSTA, ENPC, Centrale Nantes, among others) for courses on reliability analysis, risk assessment, and UQ.

The library has been introduced in several publications (e.g., Baudin et al., 2016) and used in a broad range of UQ studies across multiple science and engineering domains (see Dutfoy, 2023; Espoeys et al., 2025; Faure et al., 2024).

The Persalys graphical user interface (EDF & Phimeca Engineering, 2016), built on top of OpenTURNS, enables users to perform UQ analyses with no (or minimal) Python programming. OpenTURNS is a requirement for several other tools, including GEMSEO (IRT Saint Exupéry, 2026) and BATMAN (P. T. Roy et al., 2018), for example.

An annual Users' Day has been held since 2008, fostering collaboration and knowledge exchange among researchers, engineers, and educators.

AI usage disclosure

Generative AI (Gemini 3.1 Pro) was used to verify the grammar and spelling of this paper, check word choices, and check that the paper complies with the requirements. No generative AI was used in the creation of the software, the technical documentation, or the algorithm design.

Acknowledgements

We acknowledge support from the OpenTURNS consortium. Additional funding was provided by EDF and Airbus through external development contracts. The project has received funding from the European Commission within the LOCOMACHS (grant agreement No. 314003) and the UMRIDA (grant agreement No. 605036) projects. It has also received funding from the French National Research Agency (ANR) within the OPUS (ANR-07-TLOG-0015) and CHORUS (ANR-13-MONU-0005) projects.

References

Absil, C. O., Riccardi, A., Vasile, M., & Tardioli, C. (2016, March). SMART-UQ: Uncertainty Quantification Toolbox for Generalised Intrusive and Non Intrusive Polynomial Algebra. *Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques*. https://indico.esa.int/event/111/contributions/238/

Adams, B. M., Bohnhoff, W. J., Bolstad, M. B., Dalbey, K. R., Ebeida, M. S., Eddy, J. P., Eldred, M. S., Hooper, R. W., Hough, P. D., Hu, K. T., Jakeman, J. D., Khalil, M., Maupin, K. A., Monschke, J. A., Prudencio, E. E., Ridgway, E. M., Rizzi, F., Robbe, P., Rushdi, A. A., … Winokur, J. G. (2025). *Dakota 6.23.0 Documentation* (No. SAND2025-14179O). Sandia National Laboratories. http://snl-dakota.github.io

Baudin, M., Dutfoy, A., Iooss, B., & Popelin, A.-L. (2016). OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation. In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of uncertainty quantification* (pp. 1–38). Springer International Publishing. https://doi.org/10.1007/978-3-319-11259-6_64-1

Blanchard, J.-B., Damblin, G., Martinez, J.-M., Arnaud, G., & Gaudier, F. (2019). The Uranie platform: an open-source software for optimisation, meta-modelling and uncertainty analysis. *EPJ Nuclear Sciences & Technologies*, *5*, 4. https://doi.org/10.1051/epjn/2018050

Debusschere, B., Sargsyan, K., Safta, C., & Chowdhary, K. (2017). Uncertainty Quantification Toolkit (UQTk). In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of uncertainty quantification* (pp. 1807–1827). Springer International Publishing. https://doi.org/10.1007/978-3-319-12385-1_56

Dutfoy, A. (2023). Uncertainty on estimated magnitudes: A New Approach Based on a Poisson Point Process of Dimension 2. *Pure and Applied Geophysics*, *180*(10), 919–933. https://doi.org/10.1007/s00024-022-03222-6

EDF, & Phimeca Engineering. (2016). *Persalys: Graphical User Interface for Uncertainty Quantification*. https://www.persalys.fr/.

Espoeys, R., Brevault, L., Balesdent, M., Ricci, S., & Mycek, P. (2025). Multifidelity Bayesian Sequential Optimization and Reliability Assessment for Aerospace Systems Design. *Journal of Aerospace Information Systems*, 1–24. https://doi.org/10.2514/1.I011614

Faure, X., Lebrun, R., & Pasichnyi, O. (2024). Impact of time resolution on estimation of energy savings using a copula-based calibration in UBEM. *Energy and Buildings*, *311*, 114134. https://doi.org/10.1016/j.enbuild.2024.114134

Feinberg, J., & Langtangen, H. P. (2015). Chaospy: An open source tool for designing methods of uncertainty quantification. *Journal of Computational Science*, *11*, 46–57. https://doi.org/10.1016/j.jocs.2015.08.008

GPy. (2012). *GPy: A Gaussian process framework in Python*. http://github.com/SheffieldML/GPy.

Herman, J., & Usher, W. (2017). SALib: An open-source Python library for Sensitivity Analysis. *Journal of Open Source Software*, *2*(9). https://doi.org/10.21105/joss.00097

Iooss, B., Da Veiga, S., Janon, A., & Pujol, G. (2026). *sensitivity: Global Sensitivity Analysis of Model Outputs*. https://CRAN.R-project.org/package=sensitivity

IRT Saint Exupéry. (2026). *GEMSEO: Generic Engine for Multidisciplinary Scenarios, Exploration and Optimization*. https://gemseo.readthedocs.io/

Marelli, S., & Sudret, B. (2014). UQLab: A Framework for Uncertainty Quantification in Matlab. *Proceedings of Vulnerability, Uncertainty, and Risk*, 2554–2563. https://doi.org/10.1061/9780784413609.257

McDougall, D., Malaya, N., & Moser, R. D. (2017). The Parallel C++ Statistical Library for Bayesian Inference: QUESO. In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of uncertainty quantification* (pp. 1829–1865). Springer International Publishing. https://doi.org/10.1007/978-3-319-12385-1_57

McFarland, J. M., & Riha, D. S. (2017). Probabilistic Analysis Using NESSUS (Numerical Evaluation of Stochastic Structures Under Stress). In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of uncertainty quantification* (pp. 1733–1764). Springer International Publishing. https://doi.org/10.1007/978-3-319-12385-1_54

Olivier, A., Giovanis, D. G., Aakash, B. S., Chauhan, M., Vandanapu, L., & Shields, M. D. (2020). UQpy: A general purpose Python package and development environment for uncertainty quantification. *Journal of Computational Science*, *47*, 101204. https://doi.org/10.1016/j.jocs.2020.101204

OpenTURNS's forum. (2020). https://openturns.discourse.group/.

Patelli, E. (2016). COSSAN: A Multidisciplinary Software Suite for Uncertainty Quantification and Risk Management. In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of uncertainty quantification* (pp. 1–69). Springer International Publishing. https://doi.org/10.1007/978-3-319-11259-6_59-1

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel,

M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Pflüger, D. (2010). *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Institut für Informatik, Technische Universität München; Verlag Dr. Hut. ISBN: 9783868535556

pSeven SAS. (2026). *pSeven: Design Space Exploration and Predictive Modeling Framework*. https://www.pseven.io/product/pseven-core/

Rabiti, C., Alfonsi, A., Cogliati, J., Mandelli, D., Kinoshita, R., Sen, S., Wang, C., & Chen, J. (2017). *RAVEN User Manual* (INL/EXT-15-34123). Idaho National Laboratory (INL). https://doi.org/10.2172/1466671

Roy, P. T., Ricci, S., Dupuis, R., Campet, R., Jouhaud, J.-C., & Fournier, C. (2018). BATMAN: Statistical analysis for expensive computer codes made easy. *Journal of Open Source Software*, *3*(21), 493. https://doi.org/10.21105/joss.00493

Roy, P., Owen, A. B., Balandat, M., & Haberland, M. (2023). Quasi-Monte Carlo Methods in Python. *Journal of Open Source Software*, *8*(84), 5309. https://doi.org/10.21105/joss.05309

SALOME Consortium. (2026). *SALOME: The Open Source Platform for Pre- and Post-Processing for Numerical Simulation*. https://www.salome-platform.org/

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. *Proceedings of the 9th Python in Science Conference*. https://doi.org/10.25080/majora-92bf1922-011

Suleimenova, D., Arabnejad, H., Edeling, W. N., Coster, D., Luk, O. O., Lakhlili, J., Jancauskas, V., Kulczewski, M., Veen, L., Ye, D., Zun, P., Krzhizhanovskaya, V., Hoekstra, A., Crommelin, D., Coveney, P. V., & Groen, D. (2021). Tutorial applications for Verification, Validation and Uncertainty Quantification using VECMA toolkit. *Journal of Computational Science*, *53*, 101402. https://doi.org/10.1016/j.jocs.2021.101402

Tong, C. H. (2024). *PSUADE: Problem Solving environment for Uncertainty Analysis and Design Exploration* (Version 3.0). https://github.com/llnl/psuade; GitHub.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wicaksono, D., & Hecht, M. (2023). UQTestFuns: A Python3 library of uncertainty quantification (UQ) test functions. *Journal of Open Source Software*, *8*(90), 5671. https://doi.org/10.21105/joss.05671