

PoUnce: A framework for automatized uncertainty quantification simulations on high-performance clusters

Jakob Duerrwaechter¹✉, Thomas Kuhn¹, Fabian Meyer², Andrea Beck^{1,3}, and Claus-Dieter Munz¹

¹ Institute of Aerodynamics and Gas Dynamics, University of Stuttgart, Germany ² Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Germany ³ The Laboratory of Fluid Dynamics and Technical Flows, Otto von Guericke University Magdeburg, Germany ✉ Corresponding author

DOI: [10.21105/joss.04683](https://doi.org/10.21105/joss.04683)

Software

- [Review](#) ✉
- [Repository](#) ✉
- [Archive](#) ✉

Editor: [Nikoleta Glynatsi](#) ✉

Reviewers:

- [@georgiastuart](#)
- [@salm8](#)

Submitted: 02 August 2022

Published: 14 February 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

PoUnce (Propagation of Uncertainties) is a Python framework for fully automatized runs of non-intrusive forward uncertainty quantification (UQ) simulations on high performance computers.

In UQ, some input parameters are not known precisely, but are uncertain with a given random distribution. In non-intrusive UQ methods, a standard simulation model is evaluated many times with different input parameter values and the model outputs are post-processed to obtain information about the output quantity of interest (QoI). In the Multilevel Monte Carlo ([Giles, 2008](#); [Heinrich, 2001](#)) and Multifidelity Monte Carlo ([Peherstorfer et al., 2018, 2016](#)) methods, different models with vastly different cost and fidelity are combined. The number of evaluations with each model is usually determined adaptively and iteratively at simulation runtime.

PoUnce enables UQ simulations with a computational cost that requires high performance computing (HPC) clusters. It serves as a connecting piece between UQ methods, simulation codes, and cluster software. The framework is designed to generate simulation input for each single model evaluation, schedule and run all model evaluations on the cluster, and post-process their results. It can be easily adapted to individual needs, such as new means of interaction with different baseline simulation codes.

Three UQ methods are currently implemented: the multilevel Monte Carlo method, the multifidelity Monte Carlo method, and the non-intrusive polynomial chaos method.

Statement of need

Uncertainty quantification has become a central tool over the last years to increase reliability in numerical simulations across a wide range of scientific fields. It captures uncertain simulation input data as a potential source of error and quantifies its effect on the simulation output.

There are already several UQ software packages, such as the Dakota toolbox ([Adams et al., 2014](#)) as the most prominent, and others such as the UQ toolkit ([B. Debusschere et al., 2017](#); [B. J. Debusschere et al., 2004](#)), UQpy ([Olivier et al., 2020](#)), PyMLMC ([Sukys et al., 2017](#)), ChaosPy ([Feinberg & Langtangen, 2015](#)), UQLab ([Marelli & Sudret, 2014](#)), and UQit ([Rezaeiravesh et al., 2021](#)). These existing packages often include a large variety of UQ methods, but most do not provide an integrated framework for fully automatized UQ runs. Some provide basic scheduling capabilities, which entail, however, several performance

bottlenecks, as outlined below. PoUncle closes this gap and provides the following capabilities, which sets it apart from other software:

- **Integration and automatization:** A large-scale UQ simulation involves many individual steps, which are usually carried out separately by hand: stochastic input generation for sample simulations, determining the required HPC resources, interaction with a HPC scheduler, extracting post-processed quantities of interest from the sample simulations and stochastic evaluation. In some methods, these steps even have to be carried out several times in an iterative loop. Unlike with other packages, PoUncle fully automatizes these runs, such that they can be executed with one single command.
- **Efficiency on HPC clusters:** Non-intrusive UQ simulations consist of large numbers of smaller sample simulations. This is particularly the case in Multilevel and Multifidelity Monte Carlo simulations, where the cost between the computationally least and most expensive sample simulations can differ by many orders of magnitude. HPC clusters are not designed for this kind of applications, which entails performance bottlenecks, if no measures are taken to prevent them. This includes I/O bottlenecks due to very large numbers of relatively small files, as well as sub-optimal job-internal scheduling and idle times. PoUncle overcomes these issues by grouping large numbers of sample simulations into a common program execution with a common file I/O. Furthermore, in the interaction with the HPC scheduler, separate large-scale compute jobs are used for sample simulations of similar size, and post-processing is performed outside of these large compute jobs. This makes internal scheduling much more efficient and reduces idle times. Details can be found in the code documentation.
- **Potential for extension:** The modularity of PoUncle together with its very compact source code lowers the threshold to extend the code and adapt it to every users' individual needs. This includes adding adapters to new source codes, adapting interaction with new HPC clusters, and adding new UQ methods. Since users base UQ simulations on their own baseline codes and use different clusters, this modularity and extensibility is vital for PoUncle's applicability.

Research based on PoUncle

Studies on uncertainty quantification for cavity aeroacoustics (Kuhn et al., 2019) as well as on hp-MLMC methods (Beck et al., 2020) are based on earlier versions of this software. A publication on uncertainty quantification for iced airfoil performance based on a recent version of the software is currently in preparation.

Acknowledgements

The development of PoUncle was funded by the Elisabeth and Friedrich Boysen Foundation.

PoUncle makes use of several libraries which we would like to acknowledge. This includes GNU parallel (Tange, 2011), chaospy (Feinberg & Langtangen, 2015) and pyKriging (Paulson & Ragkousis, 2015).

References

- Adams, B. M., Ebeida, M. S., Eldred, M. S., Jakeman, J. D., Swiler, L. P., Stephens, J. A., Vigil, D. M., Wildey, T. M., Bohnhoff, W. J., Eddy, J. P., Hu, K. T., Dalbey, K. R., Bauman, L. E., & Hough, P. D. (2014). *Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis*. <https://doi.org/10.2172/1177077>

- Beck, A., Dürrwächter, J., Kuhn, T., Meyer, F., Munz, C.-D., & Rohde, C. (2020). *hp*-multilevel Monte Carlo methods for uncertainty quantification of compressible Navier–Stokes equations. *SIAM J. Sci. Comput.* [10.1137/18M1210575](https://doi.org/10.1137/18M1210575)
- Debusschere, B. J., Najm, H. N., Pebay, P. P., Knio, O. M., Ghanem, R. G., & Le Maitre, O. P. (2004). Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM J. Sci. Comput.*, 26(2), 698–719. <https://doi.org/10.1137/S1064827503427741>
- Debusschere, B., Sargsyan, K., Safta, C., & Chowdhary, K. (2017). The uncertainty quantification toolkit (UQtk). In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of uncertainty quantification* (pp. 1807–1827). Springer.
- Feinberg, J., & Langtangen, H. P. (2015). Chaospy: An open source tool for designing methods of uncertainty quantification. *J. Comput. Sci.*, 11, 46–57. <https://doi.org/10.1016/j.jocs.2015.08.008>
- Giles, M. B. (2008). Multilevel Monte Carlo path simulation. *Oper. Res.*, 56(3), 607–617. <https://doi.org/10.1287/opre.1070.0496>
- Heinrich, S. (2001). Multilevel Monte Carlo methods. In *Large-Scale Scientific Computing* (pp. 58–67). Springer. https://doi.org/10.1007/3-540-45346-6_5
- Kuhn, T., Dürrwächter, J., Meyer, F., Beck, A., Rohde, C., & Munz, C.-D. (2019). Uncertainty quantification for direct aeroacoustic simulations of cavity flows. *J. Theor. Comput. Acoust.*, 27(01), 1850044. <https://doi.org/10.1142/S2591728518500445>
- Marelli, S., & Sudret, B. (2014). UQLab: A framework for uncertainty quantification in Matlab. *American Society of Civil Engineers*, 2554–2563. <https://doi.org/10.1061/9780784413609.257>
- 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. *J. Comput. Sci.*, 47, 101204. <https://doi.org/10.1016/j.jocs.2020.101204>
- Paulson, C., & Ragkousis, G. (2015). pyKriging: A Python Kriging toolkit. *Zenodo*. <https://doi.org/10.5281/zenodo.21389>
- Peherstorfer, B., Willcox, K., & Gunzburger, M. (2018). Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Rev.* <https://doi.org/10.1137/16M1082469>
- Peherstorfer, B., Willcox, K., & Gunzburger, M. (2016). Optimal model management for multifidelity Monte Carlo estimation. *SIAM J. Sci. Comput.* <https://doi.org/10.1137/15M1046472>
- Rezaeiravesh, S., Vinuesa, R., & Schlatter, P. (2021). UQit: A Python package for uncertainty quantification (UQ) in computational fluid dynamics (CFD). *Journal of Open Source Software*, 6(60), 2871. <https://doi.org/10.21105/joss.02871>
- Sukys, J., Rasthofer, U., Wermelinger, F., Hadjidoukas, P., & Koumoutsakos, P. (2017). Optimal fidelity multi-level Monte Carlo for quantification of uncertainty in simulations of cloud cavitation collapse. *arXiv*. <https://arxiv.org/abs/1705.04374v1>
- Tange, O. (2011). GNU parallel - the command-line power tool; *login: The USENIX Magazine*, 36(1), 42–47. <https://doi.org/10.5281/zenodo.16303>