A general software tool for time-varying global sensitivity analysis and uncertainty quantification at scale and its application to hydrological models

Ivana Jovanovic Buha[1]

[1]) Technical University of Munich, Chair of Scientific Computing in Computer
Science, TUM School of Computation, Information and Technology,
Boltzmannstraße 3, 85748 Garching, Germany
ivana.jovanovic@tum.de

In this work, we present a software tool that allows for efficient time-varying forward uncertainty quantification (FUQ) and variance-based global sensitivity analysis (GSA) of (large-scale) hydrological models. It can easily be extended with other environmental and computational models. The main use case of our tool is to have a unified codebase for performing UQ and SA on models that produce time-series as an output, while offering scalability and portability from development systems to large compute systems, such as compute clusters or cloud environments.

From an algorithmic perspective, our main focus is on constructing a surrogate model in a parameter space of five dimensions or higher, using the general polynomial chaos expansion (gPCE) and pseudo-spectra approach [8] to emulate the output of complex models. We also rely on sparse grid (SG) strategies to minimize the number of necessary model runs. To preserve the black-box property of the combination technique and to focus on regions of interest adaptively, we utilize a recently proposed spatially adaptive SG combination technique with a dimension-wise refinement algorithm [5]. Finally, we use the gPCE coefficients to produce the Sobol sensitivity indices [7], which provide insights into how changes in model input parameters affect variations in model predictions.

We implemented our tool using Python and relied on the Chaospy numerical toolbox [1] to offer a wide range of methods for UQ and SA, as well as the SparseSpACE toolbox [6] for spatially adaptive SG. To enable efficient work with time-series data, we used the Pandas library and the Dash library to create the front end of our toolbox. Due to the model's runtime, parallel execution and post-processing of UQ analyses are critical in our solution. Our tool builds on the existing Uncertainty Quantification Execution Framework (UQEF) [3], which offers dynamic scheduling features to automatically distribute the work, resulting in optimal utilization of available computing resources.

Using the HBV hydrological model, which is commonly found in scientific literature [2], and the highly parameterized and CPU-demanding LARSIM model [4], we demonstrate that our results can provide insight into the stochastic importance of parameters and generate a reliable uncertainty band for flow predictions in a reasonable time. Our work bridges the gap between earlier theoretical work on UQ and more complex real-world problems, offering a valuable tool for decision-makers.

References

[1] Feinberg, J. and Langtangen H. P. 2015. Chaospy: An open source tool for designing methods of uncertainty quantification. *Journal of Computational Science*

[2] Gupta, H. V. and Razavi, S. 2018. Revisiting the Basis of Sensitivity Analysis for Dynamical Earth System Models. *Water Resources Research*

[3] Künzner, F. 2021.*Efficient non-intrusive uncertainty quantification for large-scale simulation scenarios* Technische Universität München

[4] Ludwig, K. and Bremicker, M. 2006. The Water Balance Model LARSIM: Design, Content and Applications.

[5] Obersteiner, M. and Bungartz H. J. 2021. Chaospy: A generalized spatially adaptive sparse grid combination technique with dimension-wise refinement. *SIAM Journal on Scientific Computing*

[6] Obersteiner, M. 2023. SparseSpACE - The Sparse Grid Spatially Adaptive Combination Environment. *GitHub repository* https://github.com/obersteiner/sparseSpACE

[7] Sudret, B. 2008. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety* Vol. 93. pp. 964–979

[8] Xiu, D. and Karniadakis, G. E. 2002. The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM Journal of Scientific Computing* Vol. 24. pp. 619–644