

Ourovoros: An Integrated Framework for Forward and Inverse Uncertainty Quantification of Thermal-Hydraulic System Code

Byung-Hyun You *

Korea Atomic Energy Research Institute, 111, Daedeok-daero 989beon-gil, Yuseong-gu, Daejeon 34057,
Republic of Korea

*Corresponding author: bhyou@kaeri.re.kr

***Keywords :** sensitivity analysis, Bayesian calibration, adaptive Markov chain Monte Carlo, thermal-hydraulic system codes

1. Introduction

Predictions from best-estimate nuclear system analysis codes are subject to uncertainties arising from initial and boundary conditions, physical models, and numerical methods. Two complementary analyses quantify these uncertainties: forward uncertainty quantification (UQ), in which the parameter uncertainties are propagated through the simulation model to obtain response distributions, and inverse UQ, in which the parameter distributions themselves are calibrated against experimental data. To support both analyses for thermal-hydraulic system codes, the Korea Atomic Energy Research Institute (KAERI) developed PAPIRUS, a parallel computing toolkit for data assimilation, uncertainty propagation, Chi-square linearity testing, and sensitivity analysis [1].

This paper introduces Ourovoros, a framework developed at KAERI that integrates uncertainty propagation, sensitivity analysis, and probabilistic parameter estimation into a single desktop application. The framework was designed around three requirements. First, given the same project configuration, every generated design and every Markov chain must be reproducible, so that a reported distribution can always be traced to the exact inputs and random numbers that produced it. Second, the statistical methods, including the sampling designs, convergence diagnostics, and treatment of model discrepancy, should reflect the current literature. Third, the tool should manage long-running solver executions from preparation through analysis without external infrastructure. The coupling to the simulation code is non-intrusive in that parameters are perturbed directly in text-based input decks, and the architecture accommodates additional system analysis codes through dedicated adapters.

Section 2 describes the architecture of the framework. Sections 3 and 4 summarize the forward and inverse UQ features, respectively, and Section 5 concludes the paper.

2. Framework Architecture

Ourovoros consists of a cross-platform desktop application (ouro) and a computing server (voros). On a single workstation, the application launches and manages a local server, and the same interface can

manage a server on a remote cluster through an SSH tunnel. The server stores the project data and executes the solvers, allowing an analysis in progress to continue across user-interface restarts. The framework imposes no whole-run timeout, and solver runs lasting from seconds to weeks are treated uniformly. Ensemble members in the forward workflow and the case simulations for each trial in the inverse workflow are dispatched concurrently on the available processors.

Each project is stored in a self-contained directory. The project manifest; registered copies of the input decks, solver binaries, and experimental data; and all run artifacts are stored in this directory under hashed references. Each run uses a copy of its inputs recorded during the preparation phase, so its results remain interpretable after subsequent project edits. The entire project can be exported and re-imported as one archive for exchange or long-term storage, and the principal run products, such as design matrices, stored run inputs, extracted response tables, posterior samples, and chain histories, can be exported individually in raw, CSV, or JSON form, depending on the product.

All random numbers are derived from a single project seed through counter-based substreams addressed by fixed coordinates, e.g., (chain, step) for an MCMC proposal, and the resulting streams are reproduced bit-for-bit across machines. Given the same configuration, a design matrix is therefore regenerated exactly; any random number used by a chain can be re-derived independently, and an interrupted chain run resumes deterministically from its stored inputs and execution record. Numerical regression tests verify this behavior for the statistical kernels.

Parameters are defined directly in the input deck: the user binds each parameter to numeric tokens, and a perturbation expression (by default a multiplicative factor) maps a sampled value to the token. Substituted values follow the numeric notation of the original token and preserve the column layout whenever the replacement fits within the original width.

2.1 Forward Uncertainty Quantification

The forward workflow proceeds through sampling design, ensemble execution, response extraction, and analysis. Four methods are provided: uncertainty

propagation, Wilks tolerance-limit analysis, Morris screening, and Sobol variance decomposition. Responses are extracted from the solver output tables using registered channel identities; transient channels are read as time series, steady-state channels as aggregates over a designated row range, and extraction failures are reported for each ensemble member.

Ensembles for propagation are generated by simple random sampling or Latin Hypercube Sampling (LHS) [2]. Wilks' method derives the required number of runs from the requested coverage, confidence, and order statistic [3], and supports higher-order tolerance limits commonly used in best-estimate analyses. The Morris method constructs its ensemble from standard grid-jump trajectories [4] using distance-optimized selection, whereas the Sobol method uses pick-freeze blocks built on scrambled low-discrepancy sequences.

Each analysis reports diagnostics for assessing the reliability of its estimates. The propagation method computes standardized regression coefficients on the raw and rank scales together with the correlation coefficients, the coefficient of determination, and variance inflation factors, so that the validity of the linear summary can be judged. The tolerance bands are constructed from order statistics and presented with the achieved confidence, the position-wise coverage, and the excursions of the experimental curve. Morris screening reports the elementary-effect statistics with bootstrap confidence intervals and convergence diagnostics, so the sufficiency of the trajectory budget is examined rather than assumed. Sobol first-order and total indices are computed using the estimators recommended by Saltelli et al. [5], with bootstrap intervals and block-wise convergence traces.

2.2 Inverse Uncertainty Quantification

The inverse workflow estimates the posterior distributions of the model parameters from measured data using Bayes' theorem. Observables are declared as pairs comprising a calculated channel and an experimental channel. In the preparation phase, the experimental points falling within the span of the nominal calculated curve are selected as the comparison grid, and each trial curve is interpolated onto this grid, so that every trial is evaluated against the same data. The likelihood assumes independent Gaussian residuals whose variances follow a relative-error model with an absolute lower bound. The variance model can be augmented with two optional global quantities sampled jointly with the parameters: a scalar model discrepancy term following Kennedy and O'Hagan [6], and an error scale that calibrates the declared variances as a whole. A residual-based recommendation helps select the relative errors by solving for the value at which the declared noise explains the nominal residuals.

Sampling uses an adaptive Metropolis algorithm. The proposal covariance is based on the empirical covariance of the chain and uses the standard scaling of Haario et al. [7]; the overall scale is adjusted toward a target

acceptance rate by a stochastic-approximation recursion, and the adaptation continues throughout the run, with both adaptation processes diminishing and subject to explicit bounds. Parameters with log-family priors are sampled in logarithmic coordinates with the corresponding Jacobian correction. Parameters designated as known retain their prior uncertainty without being calibrated; they are excluded from the Metropolis update and resampled from the prior at every trial, so the reported posterior marginalizes over them exactly. Multiple chains run in parallel; the case simulations for each trial are evaluated concurrently, and the chains report their progress during execution: trial outcomes, acceptance rates, relative log-likelihood traces, and the evolution of the convergence diagnostics.

Convergence is monitored with the rank-normalized split R-hat and the bulk and tail effective sample sizes (ESS) of Vehtari et al. [8]. A chain run has no predefined number of steps; automatic completion occurs only when all monitored quantities satisfy the R-hat and ESS targets over the latter half of each chain, and must satisfy them again after the chains have doubled in length, so that a single accidentally favorable window cannot end a run prematurely. Thinning is deliberately omitted, since discarding samples only loses information, and burn-in is selected during posterior analysis rather than fixed as a sampling parameter.

Post-processing of the posterior samples includes marginal summaries with Monte Carlo standard errors for the reported quantiles, candidate distribution fits ranked by the Akaike Information Criterion (AIC), scatter and correlation views, and posterior predictive bands. Cases designated for validation are excluded from the likelihood; the same predictive computation then tests the calibrated model against the withheld data.

3. Summary and Conclusions

Ourovoros integrates forward and inverse uncertainty quantification for thermal-hydraulic system codes in a single desktop framework. These capabilities are supported by reproducible sampling; designs and convergence diagnostics drawn from the current literature; treatment of model discrepancy and error scale; diagnostic-driven stopping of MCMC simulations; and a self-contained, portable project format. The statistical kernels are verified through numerical regression tests and, where available, comparisons with independently derived reference values. Future work includes maximum a posteriori estimation and coupling to additional simulation codes.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2022-00144466).

REFERENCES

- [1] J. Heo and K. D. Kim, PAPIRUS, a parallel computing framework for sensitivity analysis, uncertainty propagation, and estimation of parameter distribution, *Nuclear Engineering and Design*, Vol. 292, pp. 237-247, 2015.
- [2] M. D. McKay, R. J. Beckman, and W. J. Conover, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics*, Vol. 21, No. 2, pp. 239-245, 1979.
- [3] S. S. Wilks, Determination of Sample Sizes for Setting Tolerance Limits, *Annals of Mathematical Statistics*, Vol. 12, No. 1, pp. 91-96, 1941.
- [4] M. D. Morris, Factorial Sampling Plans for Preliminary Computational Experiments, *Technometrics*, Vol. 33, No. 2, pp. 161-174, 1991.
- [5] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index, *Computer Physics Communications*, Vol. 181, No. 2, pp. 259-270, 2010.
- [6] M. C. Kennedy and A. O'Hagan, Bayesian Calibration of Computer Models, *Journal of the Royal Statistical Society: Series B*, Vol. 63, No. 3, pp. 425-464, 2001.
- [7] H. Haario, E. Saksman, and J. Tamminen, An Adaptive Metropolis Algorithm, *Bernoulli*, Vol. 7, No. 2, pp. 223-242, 2001.
- [8] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner, Rank-Normalization, Folding, and Localization: An Improved R-hat for Assessing Convergence of MCMC, *Bayesian Analysis*, Vol. 16, No. 2, pp. 667-718, 2021.