Investigation of Adaptive Markov Chain Monte Carlo Algorithms for Inverse Uncertainty Quantification

Xu Wu, Tomasz Kozlowski

Department of Nuclear, Plasma and Radiological Engineering
University of Illinois at Urbana-Champaign
224 Talbot Laboratory, 104 South Wright Street, Urbana, Illinois, 61801, USA
xuwu2@illinois.edu, txk@illinois.edu

Abstract - *For best-estimate system thermal-hydraulics codes like TRACE, significant uncertainties come from the closure laws which are used to describe transfer terms in the balance equations. The accuracy and uncertainty information of these correlations are usually unknown to the code users, which results in the user simply ignoring or describing them using expert opinion or personal judgment during uncertainty and sensitivity analysis. The purpose of this paper is to replace such ad-hoc expert judgment of the uncertainty information of TRACE physical model parameters with inverse Uncertainty Quantification (UQ) based on OECD/NRC BWR Full-size Fine-Mesh Bundle Tests (BFBT) benchmark steady-state void fraction data.*

Inverse UQ seeks statistical descriptions of the physical model random input parameters that are consistent with the experimental data. Inverse UQ always captures the uncertainty of its estimates rather than merely determining point estimates of the best-fit input parameters. Bayesian analysis is used to establish the inverse UQ problems based on experimental data, with systematic and rigorously derived surrogate models based on Sparse Grid Stochastic Collocation (SGSC). Several adaptive Markov Chain Monte Carlo (MCMC) sampling techniques are investigated and implemented to explore the posterior probability density functions. This research solves the problem of lack of uncertainty information for TRACE physical model parameters for the closure relations. The quantified uncertainties will be useful for future uncertainty and sensitivity study of TRACE code in nuclear reactor system design and safety analysis.

I. INTRODUCTION

Within the BEPU (Best Estimate plus Uncertainty) methodology [1] uncertainties must be quantified in order to prove that the investigated design remains within acceptance criteria. For best-estimate system thermal-hydraulics codes like TRACE and RELAP5, significant uncertainties come from the closure laws which are used to describe transfer terms in the balance equations. The accuracy and uncertainty information of these correlations are usually unknown to the code users, which results in the user simply ignoring or describing them using expert opinion or personal judgment during uncertainty and sensitivity analysis. Inverse uncertainty quantification (UQ) can be used to replace such ad-hoc expert judgment.

Inverse uncertainty quantification (UQ) is the process of quantifying the uncertainty in input parameters given relevant experimental measurements and code simulation results [2] [3] [4]. It is same with Bayesian calibration when Bayesian inference methodology is used. Within the Bayesian framework we seek the posterior distributions of the model uncertain input parameters, which is updated from our prior knowledge given relevant experimental data.

In this paper, we focus on the TRACE [5] uncertain physical model parameters, and use BFBT benchmark steady-state void fraction data [6] to inversely quantify the uncertainties in these parameters. Markov Chain Monte Carlo (MCMC) algorithms [7] [8] [9] are needed to explore the posterior distribution. And our focus in this paper is to investigate and compare various adaptive MCMC sampling algorithms. Because MCMC sampling typically requires thousands of samples which is impractical for expensive computer codes, we implement the Sparse Grid Stochastic Collocation (SGSC) surrogate model during the MCMC sampling to greatly reduce the computational cost.

II. THEORY

1. Inverse UQ formulation under the Bayesian Framework

Consider a forward model:

$$\mathbf{d} \approx G(\mathbf{m}) \tag{1}$$

where $\mathbf{m}$ is a vector of input model parameters and $\mathbf{d}$ is a vector of observable quantities, or data. The forward model G yields predictions of the data as a function of the input model parameters. Define a posterior Probability Density Function (PDF) for the input model parameters $\mathbf{m}$, given observation of the data $\mathbf{d}$:

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{\int p(\mathbf{d}|\mathbf{m})P(\mathbf{m})d\mathbf{m}} \tag{2}$$

In the above equation, $p(\mathbf{m})$ is the prior, $p(\mathbf{d}|\mathbf{m}) = L(\mathbf{m})$ is the likelihood function, $p(\mathbf{m}|\mathbf{d})$ is the posterior. Prior and posterior probabilities represent *degrees of belief* about possible values of $\mathbf{m}$, before and after observing the data $\mathbf{d}$. The likelihood function is the probability that observed outcome will happen given the prior information.

In order to estimate $\mathbf{m}$ based on observable outputs $\mathbf{d}$, we start with an assumption for the likelihood function. A simple model for the likelihood assumes that independent additive errors [10] [11] account for the deviation between predicted and observed values of $\mathbf{d}$:

$$\mathbf{d} = G(\mathbf{m}) + \mathbf{e} \tag{3}$$

The components of $\mathbf{e}$ are i.i.d. random variables with density p_e . A typical assumption is $e_i \sim \mathcal{N}(0, \sigma^2)$, in which case $p(\mathbf{d}|\mathbf{m})$ becomes $\mathcal{N}\left(G(\mathbf{m}), \sigma^2\mathbf{I}\right)$. Taking these quantities as Gaussian is convenient, and indeed might reflect some forms of error, but generally there is no expectation that this is accurate. Here σ is the standard deviation of the measurement error. The likelihood function is thus:

$$L(\mathbf{m}) = p(\mathbf{d}|\mathbf{m}) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^N} \exp\left[-\sum_{i=1}^{N} \frac{[d_i - G(\mathbf{m}_i)]^2}{2\sigma^2}\right] \tag{4}$$

where $\mathbf{m}_i$ is the set of input parameters correpsponds to the data d_i.

And for the posterior PDF, we have:

$$p(\mathbf{m}|\mathbf{d}) \propto p(\mathbf{m}) \cdot p(\mathbf{d}|\mathbf{m}) = \frac{p(\mathbf{m})}{\left(\sqrt{2\pi}\sigma\right)^N} \exp\left[-\sum_{i=1}^{N} \frac{[d_i - G(\mathbf{m}_i)]^2}{2\sigma^2}\right] \tag{5}$$

Note that in situations when one believes the measurement data are correlated, σ^2 needs to be replaced by an error covariance matrix $\mathbf{\Sigma}$. It is expected that the measurement error variances are provided along with benchmark data. The posterior PDF $p(\mathbf{m}|\mathbf{d})$ is the Bayesian solution to the inverse problem. Compared with most of deterministic inverse methods, it results in not just a single value, but a PDF. Various moments and marginal densities can be computed from the posterior PDF. However, posterior PDF is often non-standard and impliit, not normalized. We need numerical sampling method to explore posterior PDF. MCMC sampling is commonly used to explore the posterior PDF based on the above formulation.

2. Adaptive Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) [12] methods are commonly used to numerically approximate integrals of the following form:

$$\mathcal{I}(f) = \int f(x)\pi(x)dx \tag{6}$$

where $\pi(x)$ is the target probability density function, and the objective is to produce a set of random samples $(x_i)_{i=1}^N$ from the target distribution π and approximate the integral of $\mathcal{I}(f)$ by $\frac{1}{N}\sum_{i=1}^N f(x_i)$. $(x_i)_{i=1}^N$ is called a Markov chain, with π defined as the unique invariant distribution.

MCMC is widely used to sample from complicated distributions without explicitly knowing the normalizing constant. The most popular algorithm is the Metropolis-Hastings (MH) algorithm [12], which defines a family of possible transitions from one Markov chain state to the next from a proposal distribution (e.g. Gaussian). Algorithm 0 shows a popular choice of MH algorithm in which the symmetric random walk Metropolis algorithm (SRWM) is used to produce transitions [9].

Algorithm 0 Metropolis-Hastings (MH) algorithm

```
Initialize $x_0$
Choose appropriate proposal distribution $g(x^*|x)$
**for** iteration $i + 1$, $i \geq 0$, given $x_i$ **do**
    Proposal a sample $\xi \sim g(\xi|x_i)$
    Calculate the acceptance probability $\alpha = \min\left[1, \frac{\pi(\xi) \cdot g(x_i|\xi)}{\pi(x_i) \cdot g(\xi|x_i)}\right]$
    Sample $\eta \sim$ uniform$(0, 1)$
    **if** $\alpha \geq \eta$ **then**
        Accept the proposed sample, $x_{i+1} = \xi$
    **else** Reject the proposed sample, stay at the current location, $x_{i+1} = x_i$
    **end if**
**end for**
```

Although the symmetric random-walk MH algorithm is simple in design and widely applicable (even in high-dimension problems), the convergence is usually very slow and special attention is required to adjust the acceptance rate. Adaptive methods are useful in tuning critical parameters that are necessary for efficient mixing. One approach for an adaptive method is to optimally adjust the variance of transition probability, however without care the adaptive process will lose its ergodic properties (consistency of estimates and convergence to the target distribution). Therefore, rules must be defined to provide acceptable approximate sampling methods of optimizing the transition probability to ensure that ergodicity is maintained asymptotically.

Andrieu and Thoms (2008) [9] outlined several algorithms that incrementally implement adaptive MCMC techniques, and in the current study three adaptive algorithms have been implemented to examine their practical capabilities.

The first algorithm, called the "Adaptive Metropolis" (AM) algorithm, uses the classical multivariate Gaussian as the proposal distribution and recursively updates the distribution covariance to converge to the optimal choice of the true covariance of the target distribution. In this algorithm, the covariance matrix of the proposal distribution is scaled by λ. It is shown [12] that the "optimal" covariance matrix for the normal symmetric random walk MH algorithm is $2.38^2/N_x \cdot \Sigma_\pi$ where Σ_π is the true covariance matrix of the target distribution π and N_x is the dimension of the input space. This value is later used as the "optimal scaling factor" in AM algorithm by [8]. In this way, the true covariance matrix Σ_π is learned "on-the-fly". Note that in Algorithm 1 the value of the scaling factor is not updated.

The second algorithm ("Rao-Blackwellized AM") [9] takes the Rao-Blackwell approach and adjusts the estimator to update the covariance matrix depending on weighted averages using the current acceptance probability. The adjusted recursions for the mean and variance are explicitly defined as:

$$\mu_{i+1} = \mu_i + \gamma_{i+1}\left[\alpha(x_i, \xi) \cdot (\xi - \mu_i) + (1 - \alpha(x_i, \xi)) \cdot (x_i - \mu_i)\right] \tag{7}$$

$$\begin{aligned}\Sigma_{i+1} = \Sigma_i + \gamma_{i+1}\big[&\alpha(x_i, \xi) \cdot (\xi - \mu_i)(\xi - \mu_i)^\intercal + \\ &(1 - \alpha(x_i, \xi)) \cdot (x_{i+1} - \mu_i)(x_{i+1} - \mu_i)^\intercal - \Sigma_i\big]\end{aligned} \tag{8}$$

Algorithm 1 Adaptive Metropolis (AM) algorithm

```
Initialize $x_0$, $\mu_0$ and $\Sigma_0$
for iteration $i+1$, $i \geq 0$, given $x_i$, $\mu_i$ and $\Sigma_i$ do
    Proposal a sample $\xi \sim \mathcal{N}(x_i, \lambda\Sigma_i)$
    Accept the proposed sample with probability $\alpha(x_i, \xi)$, $x_{i+1} = \xi$; otherwise keep the current sample, $x_{i+1} = x_i$
    Update:
        $\mu_{i+1} = \mu_i + \gamma_{i+1}(x_{i+1} - \mu_i)$
        $\Sigma_{i+1} = \Sigma_i + \gamma_{i+1}\left[(x_{i+1} - \mu_i)(x_{i+1} - \mu_i)^\intercal - \Sigma_i\right]$
end for
```

We define the short notation:

$$\overline{x_{i+1}} = \alpha(x_i, \xi) \cdot \xi + (1 - \alpha(x_i, \xi)) \cdot x_i \tag{9}$$

$$\overline{(x_{i+1} - \mu_i)(x_{i+1} - \mu_i)^\intercal} = \alpha(x_i, \xi) \cdot (\xi - \mu_i)(\xi - \mu_i)^\intercal + (1 - \alpha(x_i, \xi)) \cdot (x_{i+1} - \mu_i)(x_{i+1} - \mu_i)^\intercal \tag{10}$$

Then Algorithm 2 can be defined as follows:

Algorithm 2 Rao-Blackwellized AM algorithm

```
Initialize $x_0$, $\mu_0$ and $\Sigma_0$
for iteration $i+1$, $i \geq 0$, given $x_i$, $\mu_i$ and $\Sigma_i$ do
    Proposal a sample $\xi \sim \mathcal{N}(x_i, \lambda\Sigma_i)$
    Accept the proposed sample with probability $\alpha(x_i, \xi)$, $x_{i+1} = \xi$; otherwise keep the current sample, $x_{i+1} = x_i$
    Update:
        $\mu_{i+1} = \mu_i + \gamma_{i+1}(\overline{x_{i+1}} - \mu_i)$
        $\Sigma_{i+1} = \Sigma_i + \gamma_{i+1}\left[\overline{(x_{i+1} - \mu_i)(x_{i+1} - \mu_i)^\intercal - \Sigma_i}\right]$
end for
```

The third algorithm ("AM with global adaptive scaling") [9] attempts to adjust the scaling of proposal covariance matrix to adjust the sampling to a target acceptance probability, rather than using a pre-set constant for the scaling factor . Specifically, the log of the scaling parameter is adjusted recursively to the predefined "optimal acceptance rate" α^{opt} (e.g. 0.234 is recommended) [8]. This allows the algorithm to control its rate of exploration by adjusting the scale if the acceptance rate is lower or higher than the optimum acceptance rate. The advantage of this algorithm is that one might expect a more rapid exploration of the target distribution following a poor initialization.

There are many other adaptive methods available, such as "Component-wise AM" [9]. Component-wise adaption addresses the issue that adaptive scaling may not be efficient in all directions simultaneously, and strategizes to use "timid" moves to initiate sampling. However, in our experience these adaptive methods will not capture the correlation between different input parameters well enough, since the chains for highly correlated parameters usually do not mix well. Another issue is that most component-wise adaptive algorithms require posterior evaluation for every dimension when a new sample is proposed. In this case, the algorithm will be at least times more expensive than Algorithms 1-3 listed above. Therefore, component-wise adaptive method is not considered appropriate for this study.

Algorithm 3 AM algorithm with global adaptive scaling

```
Initialize $x_0$, $\mu_0$ and $\Sigma_0$
for iteration $i+1$, $i \geq 0$, given $x_i$, $\mu_i$ and $\Sigma_i$ do
    Proposal a sample $\xi \sim \mathcal{N}(x_i, \lambda\Sigma_i)$
    Accept the proposed sample with probability $\alpha(x_i, \xi)$, $x_{i+1} = \xi$; otherwise keep the current sample, $x_{i+1} = x_i$
    Update:
        $\log(\lambda_{i+1}) = \log(\lambda_i) + \gamma_{i+1}\left(\alpha(x_i, \xi) - \alpha^{\text{opt}}\right)$
        $\mu_{i+1} = \mu_i + \gamma_{i+1}(\overline{x_{i+1}} - \mu_i)$
        $\Sigma_{i+1} = \Sigma_i + \gamma_{i+1}\left[\overline{(x_{i+1} - \mu_i)(x_{i+1} - \mu_i)^\intercal - \Sigma_i}\right]$
end for
```

3. Sparse Grid Stochastic Collocation

One significant issue with MCMC sampling is that every sample involves an evaluation of the forward model $G(\mathbf{m})$, which is computationally prohibitive for expensive computer codes as thousands of samples are required. In this current study, a single TRACE simulation takes about one minute, which is not expensive if we only needs tens to hundreds of simulations. However, a typical Markov chain needs 10^4 or more samples, which makes the sampling time too lengthy. Many computer codes are much more expensive than TRACE and can take hours to days to finish one simulation. In this case, surrogate models that are significantly cheaper than the original model can alleviate the computational burden, as long as they can accurately represent the input-output mapping of the original simulation. Carefully constructed surrogate models can reduce the computational effort by orders of magnitude compared with full model. In the current study we implemented the SGSC surrogate model for TRACE.

The basic idea of the Stochastic Collocation (SC) technique [13] [14] is to approximate the multi-dimensional stochastic space of the problem $f(x)$ with interpolation functions at a set of collocation points $\{x_i\}_{i=1}^{d}$. Deterministic solutions of the problem at each point x_i are used to construct an interpolant of $f(x)$ by using linear combinations of $f(x_i)$. There are two choices for such multi-dimensional interpolation; full tensor product of one-dimensional interpolation rules or sparse grid interpolation rules based on the Smolyak algorithm [15].

Suppose we want to approximate smooth functions $f : [-1, 1]^d \to \mathbb{R}$, using a finite number of function values. For a one-dimensional problem we use:

$$\mathcal{U}^i(f) = \sum_{j=1}^{m_i} f(x_j^i) \cdot \alpha_j^i \tag{11}$$

where $i \in \mathbb{N}$ denotes that the rule is in the i^{th} dimension. m_i is the number of nodes. $\{\alpha_j^i\}_{j=1}^{m_i}$ are the Lagrange polynomials for interpolation and $\{x_j^i\}_{j=1}^{m_i}$ are nodes in the i^{th} dimension. Finally,

$\{f(x_j^i)\}_{j=1}^{m_i}$ are evaluation of function at these interpolation nodes. We assume that a sequence of formulas is given for $\{i = 1, 2, ..., d\}$. For the multivariate case $d > 1$ we first define tensor product formulas:

$$\left(\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}\right)(f) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} f\left(x_{j_1}^{i_1}, \cdots, x_{j_d}^{i_d}\right) \cdot \left(\alpha_{j_1}^{i_1} \otimes \cdots \otimes \alpha_{j_d}^{i_d}\right) \qquad (12)$$

which serves as building blocks for the Smolyak algorithm [15]. The above product formula needs $\left(\prod_{j=1}^{d} m_{i_j}\right)$ function evaluations. If we have the same number of nodes m_i in each dimension then the total number of nodes will be $(m_i)^d$. The number of nodes grows exponentially as the number of dimensions d increases and quickly exceeds the available computational power. This is called *"curse of dimensionality"*.

The *Smolyak algorithm* $\mathcal{A}(q, d)$, where q is called the level of the sparse grid that is independentof the dimension d, is a linear combination of product formulas in one dimension with the following key properties. Only products with a relatively small number of nodes are used and the linear combination is chosen in such a way that the interpolation property for $d = 1$ is preserved for $d > 1$ [13] [16] [17].

For $i \in \mathbb{N}$, define the difference operator:

$$\mathcal{U}^0(f) = 1 \qquad (13)$$

$$\Delta^i(f) = \mathcal{U}^i(f) - \mathcal{U}^{i-1}(f) \qquad (14)$$

Moreover, define $|\mathbf{i}| = i_1 + i_2 + \ldots i_d$ for $i \in \mathbb{N}^d$. Then, the Smolyak algorithm $\mathcal{A}(q, d)$ is given by:

$$\mathcal{A}(q, d)(f) = \sum_{|\mathbf{i}| \leq q+d} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right)(f) \qquad (15)$$

Another form of the Smolyak algorithm is:

$$\mathcal{A}(q, d)(f) = \sum_{q+1 \leq |\mathbf{i}| \leq q+d} (-1)^{q+d-|\mathbf{i}|} \binom{d-1}{q+d-|\mathbf{i}|} \cdot \left(\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}\right)(f) \qquad (16)$$

To compute $\mathcal{A}(q, d)(f)$, one only needs to know function values at the “sparse grid” defined as:

$$\mathcal{H}(q, d) = \bigcup_{q+1 \leq |\mathbf{i}| \leq q+d} \left(\mathcal{X}^{i_1} \otimes \cdots \otimes \mathcal{X}^{i_d}\right) \qquad (17)$$

where $\mathcal{X}^{i_k} = \left\{x_{j_1}^{i_k}, x_{j_2}^{i_k}, \ldots, x_{j_d}^{i_k}\right\} \subset [-1, 1]$ denotes the set of interpolation nodes used by $\mathcal{U}^{i_k}$.

To build the sparse grid, one should start from one-dimensional interpolation/integration rules (for example, Gauss rules and Clenshaw-Curtis rules). Figure 1 shows the comparison of full tensor grids and sparse grids using Clenshaw-Curtis rule for isotropic grids of level 4 and 5. Full tensor grids include 289 and 1089 nodes for levels 4 and 5 respectively, while sparse grids only need 65 and 145 nodes respectively. The sparse grid is more advantageous with increasing number of levels, as the reduction of nodes becomes significant.

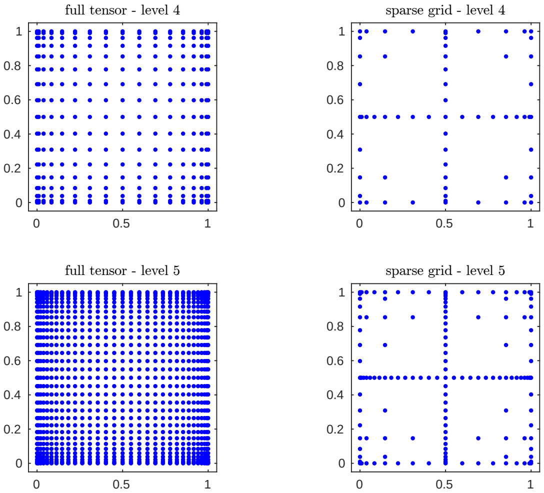

Fig. 1: Comparison of full tensor grid and classical isotropic sparse grid based on Clenshaw-Curtis rule.

III. TRACE AND BFBT BENCHMARK

TRACE [5] has been designed to perform best-estimate analyses of loss-of-coolant accidents (LOCAs), operational transients, and other accident scenarios in pressurized light-water reactors (PWRs) and boiling light-water reactors (BWRs). It can also model phenomena occurring in experimental facilities designed to simulate transients in reactor systems. TRACE version 5.0 Patch 4 includes options for user access to 36 physical model parameters from the input file. For forward uncertainty propagation, the users are free to perturb these parameters by addition or multiplication according to their personal or expert judgment. The work presented in this paper will inversely quantify the uncertainties of the parameters relevant to the considered experimental data using SGSC surrogate model by MCMC sampling. All quantified uncertainties will be multiplicative factors of the nominal values.

The international OECD/NRC BWR Full-size Fine-Mesh Bundle Tests (BFBT) [6] benchmark, based on the Nuclear Power Engineering Corporation (NUPEC) database, was created to encourage advancement in sub-channel analysis of two-phase flow in rod bundles, which has great relevance to the nuclear reactor safety evaluation. In the frame of the BFBT test program, single- and two-phase pressure losses, void fraction, and critical power tests were performed for steady-state and transient conditions.

The facility is full-scale BWR assembly, with measurement performed under typical reactor power and high-pressure, high-temperature fluid conditions found in BWRs. The full-scale fuel assembly inside the pressure vessel corresponds to the General Electric 8 by 8 assembly rod design, where each rod is electrically heated to simulate an actual reactor fuel rod. The heated length of the bundle corresponds to 3.7 m. Five different types of bundle assembly design with different combinations of geometries and power shapes were tested in the void distribution experiments.

Two types of void distribution measurement systems were employed: an X-ray computer tomography (CT) scanner and

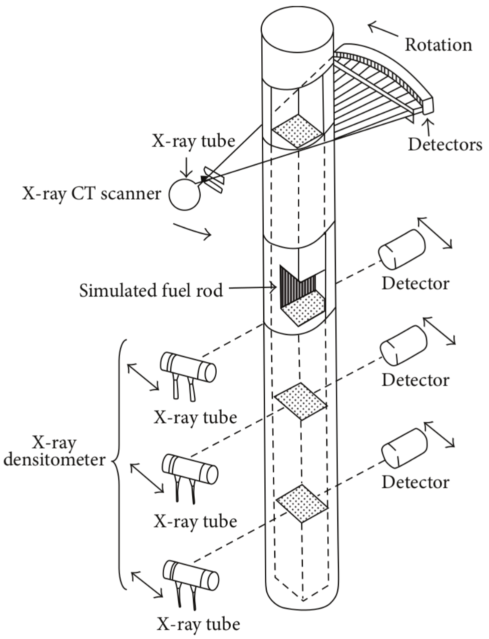

Fig. 2: BFBT benchmark void fraction measurement structure.

an X-ray densitometer (DEN). Under steady-state conditions, fine mesh void distributions were measured using the X-ray CT scanner located 50 mm above the heated length (i.e. at the assembly outlet). The X-ray densitometer measurements were performed at three different axial elevations from the bottom (i.e. 682 mm, 1706 mm and 2730 mm) under both steady-state and transient conditions. For each of the four different axial locations, the cross-sectional averaged void fraction was also measured. Figure 1 shows the void fraction measurement facility and locations. The void fraction data will be used in the current study, and they will be referred to respectively from lower to upper positions as VoidF1, VoidF2, VoidF3 and VoidF4 in the following.

The benchmark contains 392 steady-state void distribution test cases. For the current study, it is not practical to use all the test cases (each test case consists of 4 measurements, 1 at each of 4 axial elevations). Starting from the 86 test cases in assembly 4, we select the test cases by the following criteria:

1. Remove all the tests with negative void fraction data;
2. Remove all the tests that have lower void fractions at higher elevations;
3. Only keep one set of any duplicated tests;
4. Only keep measurements performed at high pressure and high power (above 7 MPa and 3MW), as these combinations will produce high void fractions thus more accurate surrogate model.
5. Remove all the tests that have low void fraction (less than 1%) at low elevations;

Only 8 test cases satisfy these criteria and are selected for inverse UQ, their process conditions and void fraction data are included in Table 1.

The X-ray densitometers can only capture the void fraction between the rod rows, therefore the measured data only shows the void fraction of a limited area of the subchannel. However, void fraction in the subchannel is not equally distributed as pointed out in [18]. For example, at low void fraction with bubbly flow, the void is concentrated in small bubbles close to the heat surface, while at high void fractions with slug flow, large bubbles are more likely to be located in the subchannel center. Consequently, the void fractions are under predicted at low void fractions and over predicted at high void fractions with the present X-ray densitometers. To resolve this issue, data correction has been suggested [18] and applied [9]. The correction formulas are proposed in [18], and the correction for assembly 4 data is shown in Equation 1.

$$\alpha_{\text{corrected}} = \frac{\alpha_{\text{measured}}}{1.167 - 0.001 \cdot \alpha_{\text{measured}}} \tag{18}$$

All the void fractions are in (%) and Equation 1 is recommended for measured void fractions between 20% and 90% (note that VoidF4 is not corrected because it is measured by CT scanner which does not have the aforementioned limitations of an X-ray densitometer).

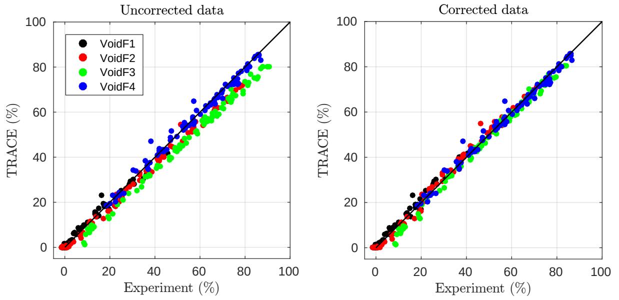

Fig. 3: Comparison of void fractions from BFBT measurement and TRACE simulation for all 86 test cases of assembly 4

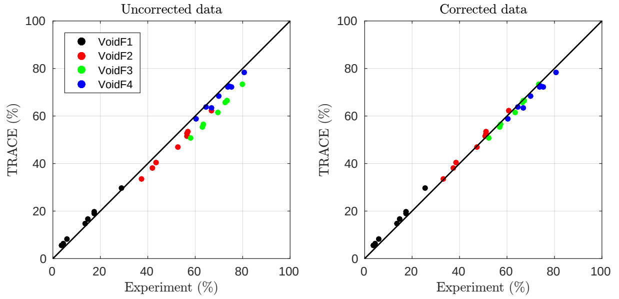

Fig. 4: Comparison of void fractions from BFBT measurement and TRACE simulation for 8 selected test cases of assembly 4

Figure 2 shows a comparison of void fraction from BFBT measurements (with and without correction) and TRACE simulations. All 86 test cases are presented. Before data correction, the majority of the void fractions are under predicted especially for VoidF1, VoidF2 and VoidF3. After data correction, the data points are more concentrated close to the diagonal line, meaning that the agreement between measurement (BFBT) and calculation (TRACE) is improved. Figure 3 shows the comparison of the 8 selected cases.

TABLE I: Process conditions and void fraction data for 8 selected cases of assembly 4

Test ID	Pressure (MPa)	Flow rate (t/h)	Inlet subcooling (kJ/kg)	Power (MW)	VoidF1 (%)	VoidF2 (%)	VoidF3 (%)	VoidF4 (%)
4101-58	7.152	54.58	50.6	3.52	5.80	43.4	63.4	64.5
4101-59	7.190	54.57	52.1	4.88	17.4	56.7	73.5	73.7
4101-60	7.178	54.62	50.5	4.89	17.3	56.8	73.3	74.0
4101-61	7.180	54.65	52.5	6.48	29.0	66.7	79.8	80.7
4101-67	7.248	69.58	54.6	4.48	4.50	42.1	63.0	66.8
4101-68	7.275	69.56	56.0	6.22	14.9	56.5	72.7	75.1
4101-84	8.680	54.66	53.2	3.35	3.80	37.4	57.9	60.2
4101-86	8.705	54.59	54.2	4.62	13.5	52.8	69.7	69.8

IV. RESULTS AND ANALYSIS

1. Sensitivity Analysis

We first performed a sensitivity analysis to find the important physical model parameters. The motivation is that not all of the 36 physical model parameters are active during the simulation of BFBT benchamrk. Many of the closure models are not relevant to the BFBT benchmark and will not be called by TRACE. For example, stratified flow (parameter P1003 and P1007) and reflooding (parameter P1034 and P1035) do not occur in the BFBT benchmark experiment. After a preliminary centered parameter study, 8 parameters are identified as potential important parameters, as listed in Table II.

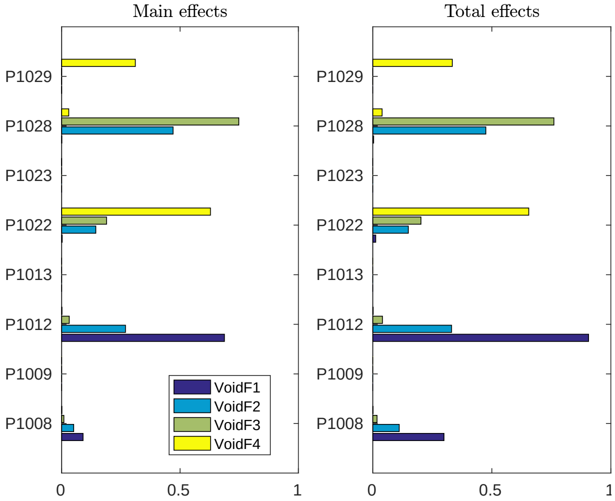

Fig. 5: Sobol' indices (main and total effects) for 8 physical model parameters.

We performed global sensitivity analysis by calculating the Sobol' indices using Polynomial Chaos Expansion (PCE) method [19]. Sobol' indices represent the part of output variance that can be attributed to each parameter. The detailed process and results are presented in a companion paper [20]. Figure 5 shows the main and total effects Sobol' indices for the 8 selected physical model parameters. It can be seen that P1009, P1013 and P1023 have very small Sobol' indices, indicating that all the four void fraction responses are not sensitive to them. They are removed from the following inverse UQ study.

2. Build the SGSC Surrogate Model

In this work, we use the Sparse Grid module of the Toolkit for Adaptive Stochastic Modeling And Non-Intrusive Approximation (TASMANIAN) [21] [22], developed at Oak Ridge National Laboratory to build the SGSC surrogate model. Clenshaw-Curtis rule and a more recently developed $\mathcal{R}$-leja rule [23] are used as the one-dimensional building block. For five-dimensional surrogate model, Leja rule of precision 4 requires 126 nodes, while Clenshaw-Curtis rule of precision 3 and 4 require 145 and 301 nodes, respectively. Here, "precision" means that the underlying one-dimensional rule can exactly interpolates polynomials of a degree up to and including the precision value. Figure 6 illustrates the adopted sparse grids in two dimensions.

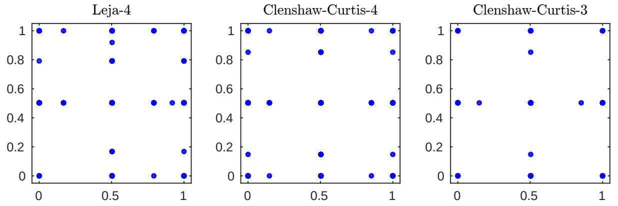

Fig. 6: Demonstration of two-dimensional nodes for Leja rule (precision 4), Clenshaw-Curtis rule (precision 4) and Clenshaw-Curtis rule (precision 3)

The constructed SGSC surrogate models are then validated by evaluating their accuracy to predict the void fractions at randomly sampled inputs. The detailed validation data will not be reported here. All of the surrogate models are able to reproduce the TRACE void fraction results accurately. Leja-4 sparse grid rule is used for the inverse UQ process, as it utilizes the least number of TRACE runs for the construction of the surrogate model.

3. MCMC Sampling

Non-informative uniform priors are used in this study to reflect our ignorance with respect to the input parameters. In the current study we assume uniform distributions over the

TABLE II: List of 8 selected physical model parameters selected after centered parameter study

Parameter	Description
P1008	Single phase liquid to wall heat transfer coefficient
P1009	Single phase vapor to wall heat transfer coefficient
P1012	Subcooled boiling heat transfer coefficient
P1013	Nucleate boiling heat transfer coefficient
P1022	Wall drag coefficient
P1023	Form loss coefficient
P1028	Interfacial drag (bubbly/slug Rod Bundle - Bestion) coefficient
P1029	Interfacial drag (bubbly/slug Vessel) coefficient

range $[0, 5]$ for all the five selected physical model parameters: P1008, P1012, P1022, P1028, P1029. Algorithms 1-3 are used for MCMC sampling. All three adaptive MCMC algorithms take about 72 core-minutes to produce 100,000 samples using a current generation Intel CPU, which would otherwise take about 1167 core-hours (48 core-days) using direct TRACE simulation with the same processor. The first 10,000 samples are discarded as burn-in and then only every 20th sample was kept for thinning of the chain, leaving us with 4500 samples. Thinning is performed to reduce auto-correlation of the samples.

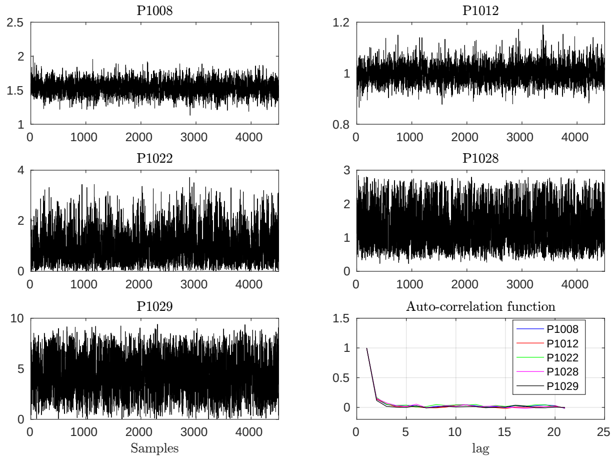

Fig. 7: MCMC chain trace plots and auto-correlation functions

Figure 7 shows mixing for the five parameters and the decay of the auto-correlation function of the Markov chain for Algorithm 3 (Markov chains for Algorithm 1-2 show very similar behavior). Note that all the five parameters are enforced to be positive, as negative values would not be physical. Very good mixing can be identified, and the auto-correlations for all the five parameters decay quickly after thinning, indicating that the Markov chain has converged.

Table III presents the statistics of MCMC chains for each parameter. The mean values and standard deviations from different MCMC algorithms are very similar for each parameter. In the following analysis the Markov chain from Algorithm 3 will be used, as it is representative of the other two Markov chains.

TABLE III: MCMC chain statistics for different physical model parameters. Chain 1 - 3 are from adaptive MCMC Algorithms 1 - 3, respectively.

Parameter	Chain 1		Chain 2		Chain 3	
	mean	std	mean	std	mean	std
P1008	1.5359	0.1115	1.5315	0.1159	1.5384	0.1141
P1012	1.0005	0.0387	1.0015	0.0399	1.0000	0.0387
P1022	1.0113	0.6634	1.0478	0.6937	1.0095	0.6842
P1028	1.3000	0.6084	1.2861	0.6126	1.3046	0.6128
P1029	4.4053	2.0451	4.3410	2.0374	4.4119	2.0570

4. Posterior Distributions

Figure 8 shows the plot for pairwise joint density contours and marginal densities for the five physical model parameters. The marginal PDFs are evaluated using Kernel Density Estimation (KDE). This plot is useful for identifying potential correlation between the parameters. Highly linear correlations are observed between some parameters, such as P1008 (single phase liquid to wall HTC) and P1012 (subcooled boiling HTC). This indicates that in future forward uncertainty propagation studies, these input parameters should be sampled jointly, not independently, so that their correlation is captured. Table IV shows the correlation coefficient matrix of all the parameters.

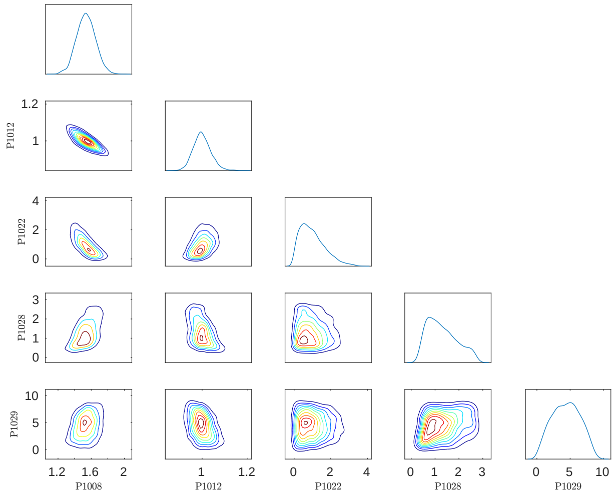

Fig. 8: MCMC chain pairwise joint density contours and marginal densities

TABLE IV: Correlation matrix

Parameter	P1008	P1012	P1022	P1028	P1029
P1008	1.00				
P1012	-0.84	1.00			
P1022	-0.73	0.39	1.00		
P1028	0.52	-0.45	-0.06	1.00	
P1029	0.29	-0.39	-0.02	0.27	1.00

To make the posterior samples more applicable to future uncertainty and sensitivity analysis, we need to fit posterior samples to well-known distributions, such as Normal or Log-normal distributions, so that they will be more easily sampled. From the marginal PDFs in Figure 8, it is obvious that for P1008, P1012 and P1029, we can consider these parameters as normal random variables. For P1022 and P1028, the PDFs are more skewed toward 0. Natural choices for these distributions include Gamma and Log-normal. Gamma distributions are chosen because Log-normal distribution has a longer tail which results in a poor match with these samples.

Figure 9 and Table V show the fitted distribution for each physical model parameter and the parameters associated with each distribution, i.e. mean (μ) and standard deviation (σ) for normal distribution, shape α and scale β parameter for Gamma distribution. All the fitted distributions are accepted by Kolmogorov-Smirnov test at the 5% significance level. Figure 10 shows that good agreement can be achieved between the empirical cumulative distribution function (CDF) and fitted CDF for every parameter.

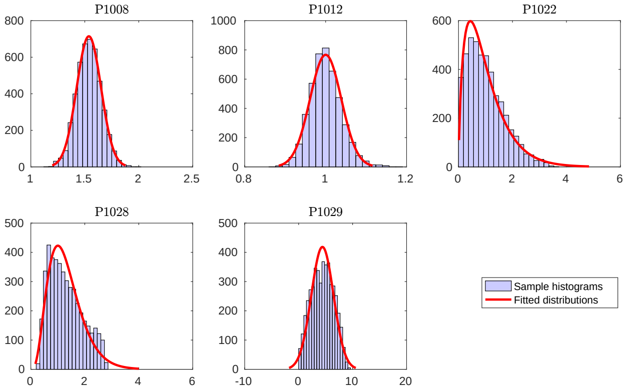

Fig. 9: Fitted posterior probability densities

TABLE V: Fitted distribution for each physical model parameter

Parameter	PDF type	PDF parameter 1	PDF parameter 2
P1008	Normal	$\mu = 1.5377$	$\sigma = 0.1131$
P1012	Normal	$\mu = 1.0001$	$\sigma = 0.0386$
P1022	Gamma	$\alpha = 1.7581$	$\beta = 0.5767$
P1028	Gamma	$\alpha = 4.3106$	$\beta = 0.3027$
P1029	Normal	$\mu = 4.4080$	$\sigma = 2.0594$

The fitted normal distribution for P1029 a has very large standard deviation and a mean value that deviates significantly from the nominal value 1.0. This is most likely because the TRACE model for BFBT benchmark is insensitive to P1029 (as shown in Figure 5, P1029 only affects VoidF4), therefore our ignorance with regards to P2019 cannot be reduced by using observation data from this benchmark in our inverse UQ process. More informative data or P1029-sensitive benchmarks are required to better quantify its uncertainty.

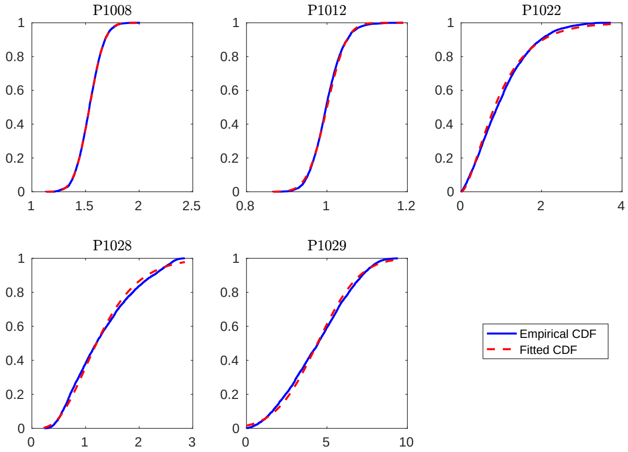

Fig. 10: Comparison of empirical CDFs and fitted CDFs.

V. CONCLUSIONS

In this paper, we applied inverse uncertainty quantification under the Bayesian framework using a SGSC surrogate model for TRACE physical model parameters. Inverse UQ aims to quantify the uncertainty in input parameters such that the discrepancies between code output and observed experimental data is minimized. The inverse UQ problem is defined using Bayesian inference theory and the solutions are the posterior distributions of uncertain input parameters. Inverse UQ always captures the uncertainty in its estimates rather than merely determining point estimates of the best-fit input parameters.

A SGSC surrogate model was constructed for TRACE based on selected BFBT test cases. The surrogate model was used for the MCMC process, due to the fact that every MCMC sample requires a model evaluation, which makes direct simulation (using TRACE) impractical. Different adaptive MCMC sampling methods were investigated; the AM algorithm, Rao-Blackwellized AM algorithm and AM algorithm with global adaptive scaling. Markov chains from each adaptive MCMC algorithm reached convergence and produced similar statistical information for the 5 input parameters (i.e. the mean values, standard deviations, and PDFs). Lastly, Gaussian and Gamma distributions were fitted for 5 physical model parameters and the parameters for these distributions were reported.

This research addresses the problem of lacking uncertainty information about TRACE physical model parameters, which has been often ignored or described using expert opinion or personal judgment in previous uncertainty and sensitivity analysis work. The results of inverse UQ of TRACE physical model parameters are important for future uncertainty and sensitivity study of TRACE code for its application in nuclear reactor system design and safety analysis.

REFERENCES

1. G. E. WILSON, "Historical insights in the development of Best Estimate Plus Uncertainty safety analysis," *Annals of Nuclear Energy*, **52**, 2–9 (2013).
2. X. WU and T. KOZLOWSKI, "Inverse uncertainty quantification of reactor simulations under the Bayesian framework using surrogate models constructed by polynomial chaos expansion," *Nuclear Engineering and Design*, **313**, 29–52 (2017).
3. R. SHRESTHA and T. KOZLOWSKI, "Inverse uncertainty quantification of input model parameters for thermal-hydraulics simulations using expectation–maximization under Bayesian framework," *Journal of Applied Statistics*, **43**, *6*, 1011–1026 (2016).
4. G. HU and T. KOZLOWSKI, "Inverse uncertainty quantification of trace physical model parameters using BFBT benchmark data," *Annals of Nuclear Energy*, **96**, 197–203 (2016).
5. USNRC, *TRAC/RELAP Advanced Computational Engine (TRACE) V4. 0 User's Manual*, Division of Safety Analysis, Office of Nuclear Regulatory Research, U. S. Nuclear Regulatory Commission, Washington, DC. (2014).
6. B. NEYKOV, F. AYDOGAN, L. HOCHREITER, K. IVANOV, H. UTSUNO, F. KASAHARA, E. SARTORI, and M. MARTIN, *NUPEC BWR full-size fine-mesh bundle test (BFBT) benchmark*, OECD/NEA, NEA/NSC/DOC(2005)5 (2005).
7. A. GELMAN, J. B. CARLIN, H. S. STERN, and D. B. RUBIN, *Bayesian data analysis*, Chapman & Hall/CRC Boca Raton, FL, USA, third ed. (2014).
8. H. HAARIO, E. SAKSMAN, and J. TAMMINEN, "An adaptive Metropolis algorithm," *Bernoulli*, pp. 223–242 (2001).
9. C. ANDRIEU and J. THOMS, "A tutorial on adaptive MCMC," *Statistics and Computing*, **18**, *4*, 343–373 (2008).
10. Y. M. MARZOUK, H. N. NAJM, and L. A. RAHN, "Stochastic spectral methods for efficient Bayesian solution of inverse problems," *Journal of Computational Physics*, **224**, *2*, 560–586 (2007).
11. Y. MARZOUK and D. XIU, "A Stochastic Collocation Approach to Bayesian Inference in Inverse Problems," *Communications in Computational Physics*, **6**, *4*, 826–847 (2009).
12. A. GELMAN, G. ROBERTS, and W. GILKS, "Efficient metropolis jumping hules," *Bayesian statistics*, **5**, *599-608*, 42 (1996).
13. D. XIU, "Efficient collocational approach for parametric uncertainty analysis," *Commun. Comput. Phys*, **2**, *2*, 293–309 (2007).
14. M. S. ELDRED, "Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design," *AIAA Paper*, **2274**, *2009*, 37 (2009).
15. S. A. SMOLYAK, "Quadrature and interpolation formulas for tensor products of certain classes of functions," in "Dokl. Akad. Nauk SSSR," (1963), vol. 4, p. 123.
16. T. GERSTNER and M. GRIEBEL, "Numerical integration using sparse grids," *Numerical algorithms*, **18**, *3-4*, 209–232 (1998).
17. V. BARTHELMANN, E. NOVAK, and K. RITTER, "High dimensional polynomial interpolation on sparse grids," *Advances in Computational Mathematics*, **12**, *4*, 273–288 (2000).
18. M. GLÜCK, "Validation of the sub-channel code F-COBRA-TF: Part II. Recalculation of void measurements," *Nuclear Engineering and Design*, **238**, *9*, 2317–2327 (2008).
19. B. M. ADAMS, W. BOHNHOFF, K. DALBEY, J. EDDY, M. ELDRED, D. GAY, K. HASKELL, P. D. HOUGH, and L. SWILER, *DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.4 UserâĂŹs Manual*, Sandia National Laboratories, Tech. Rep. SAND2014-4633 (2016).
20. X. WU, C. WANG, and T. KOZLOWSKI, "Global Sensitivity Analysis of TRACE Physical Model Parameters based on BFBT benchmark," in "Proceedings of M&C-2017," Jeju, Korea, April 16-20. (2017).
21. M. STOYANOV, "User manual: Tasmanian sparse grids v2," Tech. rep., Technical report, Oak Ridge National Laboratory, Computer Science and Mathematics Division, Oak Ridge, TN (2013).
22. M. K. STOYANOV and C. G. WEBSTER, "A dynamically adaptive sparse grids method for quasi-optimal interpolation of multidimensional functions," *Computers & Mathematics with Applications*, **71**, *11*, 2449–2465 (2016).
23. M. A. CHKIFA, "On the Lebesgue constant of Leja sequences for the complex unit disk and of their real projection," *Journal of Approximation Theory*, **166**, 176–200 (2013).