Title: An Information Theoretic Approach to Use High-Fidelity Codes to Calibrate Low-Fidelity Codes

Author(s):
Lewis, Allison
Smith, Ralph
Williams, Brian J.

Intended for: Report

Issued: 2014-09-30

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer,is operated by the Los Alamos National Security, LLC for the National NuclearSecurity Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Departmentof Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

An Information Theoretic Approach to Use High-Fidelity Codes to Calibrate Low-Fidelity Codes

Allison Lewis[*], Ralph Smith[*], and Brian Williams[†]

Introduction

Information theory in a Bayesian framework can be used for experimental design or optimal evaluation strategies for codes.

Within the context of nuclear plant design, there are a number of regimes for which it is prohibitively expensive or physically infeasible to obtain comprehensive data. For example, measuring CRUD data, in the ideal scenario, requires the complete shutdown of the plant and removal of the fuel rods. After data acquisition, engineers are left with only a low-resolution image of the deposit from which the data must be digitized and used in a least-squares inference setting. The process is further complicated by the fact that thermal contraction of cladding during cooling can cause CRUD to break off rods, thus distorting measurements. Furthermore, cold CRUD does not accurately reflect boron levels during operation where hot CRUD serves as a boron absorber. Due to the expense and limitations associated with collecting these data, optimal experimental design is critical.

High-fidelity models can, in some cases, be used to predict physical behavior in regimes where physical data are unavailable. To utilize these models, parameters such as those in closure relations must be inferred using either data or other validated high resolution codes. Due to their complexity, these codes are computationally expensive and may take on the order of hours to days to run. This motivates the use of surrogate or low-resolution models, which are less comprehensive but provide the numerical efficiency required for design optimization or control implementation. These models also have parameters or inputs that must be calibrated using experimental data or simulations from validated high-fidelity codes.

A number of methods have recently been proposed to address the integration of high- and low-fidelity codes to predict future observations in an efficient manner. As described fully in [11] and introduced earlier in [14], multi-scale frameworks have been proposed for use in many nuclear transport models. In these models, the full domain is split into multiple sub-domains on which a micro-scale approach is applied. A fine-scale transport equation is solved over each sub-domain characterized by appropriate boundary conditions, and cross-sections over angular direction, energy, and space are determined in a manner

[*] Department of Mathematics, North Carolina State University, Raleigh, NC 27695

[†] Los Alamos National Laboratory, Los Alamos, NM 87545

that preserves neutron reaction rates. A coarse model is then applied to the domain as a whole; a low-order approximation of the transport equation is solved over the entire spatial domain, but without the dependence on the angular direction that was removed via the fine-scale method. This allows for a more efficient evaluation of the model. However, there are weaknesses to this approach, the most significant of which is that the boundary conditions used in the micro-scale model to preserve neutron reaction rates are not necessarily refined based on the coarse-scale solution. Schaefer et al. [11] discusses an approach that addresses the updating of the fine-scale boundary conditions to meet the requirements of the coarse-scale model.

A second method for integrating high and low-fidelity models is that of the Method of Manufactured Universes (MMU) proposed in [16]. This framework defines a manufactured reality from which "experimental data" is collected, a proposed model with uncertain inputs which may contain model discrepancy, a process under which uncertainty quantification on the uncertain parameters is to be performed, and a follow-up assessment that can determine the accuracy of those uncertainties. The researcher defines the laws of their manufactured "universe" and creates experimental data—coinciding with the high-fidelity measurements from this investigation—that follows these laws and may contain measurement error. These "experiments" are then simulated via the proposed model—the low-fidelity model—and the differences between simulation and reality are measured using a proposed method. Once the input and parameter uncertainties are quantified, the simulated model may be used to predict future observations with a corresponding level of uncertainty.

We employ an information-theoretic approach to calibrate the parameters in the low-fidelity model, achieving an accurate approximation to the computationally expensive high-fidelity model while minimizing the number of required high-fidelity code evaluations. Our goal is similar to that in the method of manufactured universes (MMU) detailed in [16]: predict the behavior of a complex system using simulations or experimental data and be able to assess the accuracy of those predictions. We first define a high-fidelity model that accounts for the physical characteristics of the "manufactured reality" in which the problem exists—this model may be a set of governing equations that abide by the laws we have defined for our problem's "universe". From this model, we can simulate the behavior of a quantity of interest (QOI)—optionally, measurements here may be perturbed by a prescribed error model for a more realistic setting. In the MMU framework, an approximate model is then constructed to test a particular uncertainty quantification method and to assess its predictive capabilities [16]. Similarly, we construct a low-fidelity model for use in prediction. Using high-fidelity model evaluations chosen according to our information-theoretic methods, we calibrate this low-fidelity model to be a low-cost alternative to the original code. This approximation is then used in place of the high-fidelity model to make future predictions when the evaluation of the high-fidelity code is prohibitively expensive.

During the calibration stage, we address the issue of optimal high-fidelity code evaluation using a Bayesian framework. As previously mentioned, the goal is to accurately calibrate low-fidelity model parameters using as few high-fidelity model evaluations as possible. By measuring the amount of information between potential designs and parameter distributions, we can select the design that will most significantly reduce the amount of uncertainty in the parameters. We utilize a sequential design setting, where each specific design is selected based on its optimal ability to reduce uncertainties in the parameters. Once selected, the corresponding high-fidelity simulation is run and the newly acquired data is used to recalibrate the model parameters. The posterior distribution resulting from this calibration becomes the prior distribution for the next cycle. Mutual information between parameters and designs is computed again, and the next most profitable experiment or high-fidelity simulation is chosen. Once a point is reached where information gain is no longer significant, the process is terminated and the cost of additional high-fidelity model evaluations or expensive experimental data acquisition is avoided. We consider two methods for estimating the mutual information between random variables or distributions. The first is based on Monte Carlo evaluation whereas the second utilizes a k^{th}-nearest neighbor (kNN) algorithm to approximate the mutual information.

After defining these two methods of mutual information estimation and demonstrating each with a simple example, we include several applications to illustrate these concepts. In the first, we fit a quadratic model to the analytic solution of the steady state model of the heat equation. For the second, we fit a solution derived via a finite difference construction to the analytic solution of the diffusion equation. In our third example, we compare a low-fidelity point kinetics model to a one-dimensional neutron diffusion model. In each of these first three examples, Monte Carlo and kNN methods are used to accurately calibrate the low-fidelity model parameters for use in later system predictions with as few high-fidelity code evaluations as possible. Our final example delves further into the method of manufactured universes, illustrating the calibration of a low-fidelity diffusion model in a particle transport "universe" that behaves according to the S_N discrete ordinates method.

1 Design

We employ the following experimental design protocol to optimally evaluate high-fidelity codes to calibrate low-fidelity codes. Given a set of observations $D_{n-1} = \{\tilde{d}_1, \tilde{d}_2, ..., \tilde{d}_{n-1}\}$ of the high-fidelity code, we seek an evaluation strategy $\xi_n \in \Xi$ so that uncertainty in low-fidelity model parameters $\theta \in \mathbb{R}^p$ is reduced when the model is re-calibrated using the new high-fidelity data point $\tilde{d}_n$. To specify ξ_n, we employ the mutual information strategy reported in [1, 19] and more generally in [9].

We employ the statistical model

$$d_n = d_\ell(\theta, \xi_n) + \delta(\xi_n) + \varepsilon_n(\xi_n) \tag{1}$$

where $d_\ell(\theta, \xi_n)$ denotes the low-fidelity model, which depends on parameters $\theta \in \mathbb{R}^p$ that we seek to optimally calibrate using synthetic data constructed using a high-fidelity model. Here $\xi_n \in \Xi$ denotes the n^{th} design or evaluation strategy where Ξ designates the set of possible evaluation strategies or experimental conditions. For the examples considered here, Ξ is taken to be a discrete set of independent variable values. We denote potential discrepancy in the low-fidelity model by $\delta(\xi_n)$ and random measurement or discretization errors by $\varepsilon_n(\xi_n)$.

When using mutual information measures to determine the next design point ξ_n, we employ independent and identically distributed (iid) Gaussian errors, $\varepsilon_n(\xi_n) \sim \mathcal{N}(0, \sigma^2)$, where σ is a user-specified parameter. Throughout this work, we take 2σ to be 10% of

$\max_{i=1,\cdots,n-1} d_h(\xi_i)$, where $d_h(\xi_i)$ is the high-fidelity solution evaluated using the ξ_i design strategy. An alternative option is to infer σ during the calibration process. We note that this error assumption is commonly valid for measurement errors but will likely need to be modified for biases common to numerical errors.

For this investigation, we neglect model discrepancies and take $\delta(\xi_n) = 0$. Discussion regarding the use of Gaussian processes, $\delta \sim GP(0; \lambda_\delta, \rho_\delta)$, to quantify δ is provided in [5, 6].

For a given design ξ_n, the n^{th} observation $\tilde{d}_n$, generated by the high-fidelity model $d_h(\xi_n)$, is given by

$$\tilde{d}_n = d_h(\xi_n) + \tilde{\varepsilon}_n(\xi_n) \tag{2}$$

where potential numerical or measurement errors $\tilde{\varepsilon}_n(\xi_n)$ are assumed to be iid and normally distributed, $\tilde{\varepsilon}_n(\xi_n) \sim \mathcal{N}(0, \tilde{\sigma}^2)$. Here $\tilde{\sigma}$ is also a user-specified parameter. We take $2\tilde{\sigma}$ to be 10% of $\max_{i=1,\cdots,n-1} d_h(\xi_i)$ but note that, in general, $\tilde{\sigma}$ can differ from σ. We employ the high-fidelity model (2) to generate the synthetic data used to calibrate the low-fidelity model.

The change in knowledge about the model parameters due to the addition of new synthetic or experimental data $\tilde{d}_n$ is given by Bayes' rule

$$p(\theta|D_n) = \frac{p(D_n|\theta)p(\theta)}{p(D_n)} = \frac{p(\tilde{d}_n, D_{n-1}|\theta)p(\theta)}{p(\tilde{d}_n, D_{n-1})}$$

for the new data set $D_n = \{\tilde{d}_n, D_{n-1}\}$. The goal in experimental design is to optimize the information provided by an experiment or high-fidelity observation $\tilde{d}_n$ based on the design ξ_n. Because our objective is to determine the distribution of the model parameters θ from the calibration of our model with data $\tilde{d}_n$ using as few experiments as possible, the strategy upon which we base our design decision should be chosen according to the amount of information provided by the proposed data as a result of measuring under design conditions ξ_n. Since $\tilde{d}_n$ has not yet been observed when we make a decision regarding the choice of ξ_n, we employ predictions d_n provided by the statistical model (1) to determine ξ_n.

We employ Shannon entropy estimates to quantify the mutual information between the proposed observation d_n and parameter values θ. For a random variable Θ having a corresponding density $p(\theta)$, the Shannon entropy is defined as

$$H(\Theta) = -\int_\Omega p(\theta) \log(p(\theta)) d\theta$$

for the prior and

$$H(\Theta|x) = -\int_\Omega p(\theta|x) \log(p(\theta|x)) d\theta$$

for the posterior distribution given data x. We define the utility of observing the high-fidelity code at condition ξ_n, perturbed by error, as

$$U(d_n, \xi_n) = \int_\Omega p(\theta|d_n, D_{n-1}) \log p(\theta|d_n, D_{n-1}) d\theta - \int_\Omega p(\theta|D_{n-1}) \log p(\theta|D_{n-1}) d\theta, \tag{3}$$

which is a random function of unobserved data d_n. By marginalizing over the domain $\mathcal{D}$, the set of all unknown future observations, we obtain the average amount of information contributed by the proposed experiment ξ_n. The result is the mutual information

$$I(\theta; d_n|D_{n-1}, \xi_n) = \mathbb{E}[U(d_n, \xi_n)|D_{n-1}] = \int_{\mathcal{D}} U(d_n, \xi_n) p(d_n|D_{n-1}, \xi_n) dd_n \tag{4}$$

between the model parameters θ and the proposed observation d_n at design condition ξ_n. This gives a measure of the parameter uncertainty reduction provided by knowing the new data. We choose the optimal set of design conditions ξ_n^* to be the design that maximizes this quantity, namely

$$\xi_n^* = \arg\max_{\xi_n \in \Xi} I(\theta; d_n|D_{n-1}, \xi_n).$$

The high-fidelity code is then evaluated using the design condition ξ_n^* and the resulting data $\tilde{d}_n$ is used to recalibrate the model parameters θ. This design is then eliminated from the design set, as design replication is not considered in this study. We note, however, that by allowing replication in design selections, one can formally quantify uncertainty in high-fidelity code calculations due to parameter variation; e.g., turbulence models in CFD codes. A basic implementation of our method is outlined in Algorithm 1.

Algorithm 1 Design Implementation

(1) Define N, the number of samples to be used in either the Monte Carlo or kNN algorithms.

(2) Initialize a list of pre-existing high-fidelity data, $[(\xi_1, \tilde{d}_1), (\xi_2, \tilde{d}_2), ..., \xi_r, \tilde{d}_r)]$.

(3) Define list of possible design conditions, $[\xi_{r+1}, \xi_{r+2}, ..., \xi_s]$.

(4) If $r \geqslant 1$, run the DRAM algorithm (see Appendix A) to construct a chain $\{\theta^i\}_{i=1}^N$ of size $p \times N$ from the prior distribution $p(\theta|D_{n-1})$ where p is the number of parameters. If $r = 0$; that is, there is no pre-existing data, construct a chain of size $p \times N$ by sampling an informative prior $p(\theta)$ of choice.

(5) Send the chain $\{\theta^i\}$ to the Monte Carlo or kNN algorithms detailed in Sections 1.1 and 1.2.

(6) The Monte Carlo and kNN algorithms return a single design condition ξ_n. Append this value and the corresponding high-fidelity prediction $\tilde{d}_n = d_h(\xi_n) + \tilde{\varepsilon}_n(\xi_n)$ to the previous data list to obtain

$$[(\xi_1, \tilde{d}_1), (\xi_2, \tilde{d}_2), ..., (\xi_r, \tilde{d}_r), (\xi_n, \tilde{d}_n)].$$

(7) Repeat steps 4-6 until all designs are used or a user-specified error tolerance is met.

1.1 Monte Carlo Method

In general, the integral in (3) cannot be directly evaluated and requires numerical approximation. Substituting (3) into (4), we obtain

$$\begin{aligned} I(\theta; d_n|D_{n-1}, \xi_n) &= \int_{\mathcal{D}} \int_{\Omega} p(\theta, d_n|D_{n-1}, \xi_n) \log \frac{p(\theta, d_n|D_{n-1}, \xi_n)}{p(d_n|D_{n-1}, \xi_n)} d\theta dd_n \\ &- \int_{\mathcal{D}} \int_{\Omega} p(\theta, d_n|D_{n-1}, \xi_n) \log p(\theta|D_{n-1}) d\theta dd_n \\ &= \int_{\mathcal{D}} \int_{\Omega} p(\theta, d_n|D_{n-1}, \xi_n) \log \frac{p(\theta, d_n|D_{n-1}, \xi_n)}{p(\theta|D_{n-1}) p(d_n|D_{n-1}, \xi_n)} d\theta dd_n. \end{aligned} \tag{5}$$

The integral (5) is approximated via Monte Carlo quadrature. First, N samples $\{\theta^i\}_{i=1}^N$ are drawn from the prior $p(\theta|D_{n-1})$, which describes the state of knowledge about parameters θ existing before the new observation d_n is obtained. Given θ^i and ξ_n, d_n^i is drawn from the conditional predictive distribution $p(d_n(\xi_n)|\theta^i, D_{n-1}, \xi_n)$ derived from the statistical model (1). We note that the joint probability distribution $p(\theta, d_n|D_{n-1}, \xi_n)$ may be written

$$p(\theta, d_n|D_{n-1}, \xi_n) = p(d_n|\theta, D_{n-1}, \xi_n)\, p(\theta|D_{n-1}) \,,$$

leading to the following expression

$$p(d_n|D_{n-1}, \xi_n) = \int_{\Omega} p(\theta, d_n|D_{n-1}, \xi_n)\, d\theta = \int_{\Omega} p(d_n|\theta, D_{n-1}, \xi_n)\, p(\theta|D_{n-1})\, d\theta$$

for the evidence $p(d_n|D_{n-1}, \xi_n)$. Our samples of θ can thus be used to estimate the evidence for design condition ξ_n,

$$\hat{p}(d_n|D_{n-1}, \xi_n) = \frac{1}{N} \sum_{j=1}^{N} p(d_n|\theta^{(j)}, D_{n-1}, \xi_n).$$

With these expressions in hand, the logarithm in the integrand of (5) can be written

$$\log \frac{p(\theta, d_n|D_{n-1}, \xi_n)}{p(\theta|D_{n-1})\, p(d_n|D_{n-1}, \xi_n)} = \log \frac{p(d_n|\theta, D_{n-1}, \xi_n)}{p(d_n|D_{n-1}, \xi_n)} \,,$$

resulting in an estimate of the mutual information (5) for design condition ξ_n,

$$\hat{I}(\theta; d_n|D_{n-1}, \xi_n) = \frac{1}{N} \sum_{i=1}^{N} \log \frac{p(d_n^i|\theta^i, D_{n-1}, \xi_n)}{\hat{p}(d_n^i|D_{n-1}, \xi_n)}. \tag{6}$$

For $\delta(\xi_n) = 0$ and $\varepsilon_n(\xi_n) \sim \mathcal{N}(0, \sigma^2)$, (1) results in the conditional predictive distribution $p(d_n(\xi_n)|\theta, D_{n-1}, \xi_n)$ being Gaussian with mean $d_\ell(\theta, \xi_n)$ and variance σ^2.

After computing these quantities and estimating the mutual information, the design ξ_n that yields the largest mutual information $I(\theta; d_n|D_{n-1}, \xi_n)$ in (5) is chosen as the next design in the sequence. We evaluate $d_h(\xi_n)$, sample $\tilde{\varepsilon}_n(\xi_n) \sim \mathcal{N}(0, \tilde{\sigma}^2)$, augment the available data D_{n-1} by $\tilde{d}_n = d_h(\xi_n) + \tilde{\varepsilon}_n(\xi_n)$, and continue the sequence.

Each of our examples utilizes a parameter space of $\dim(\theta) \leqslant 3$ so tensored quadrature techniques are feasible. For moderate dimensionality—e.g., $p = 4$ to approximately 30—one can employ sparse grid quadrature techniques or employ the more efficient kth nearest neighbor (kNN) algorithm detailed in Section 1.2. In this work, our Monte Carlo estimates are used primarily to verify the kNN algorithm.

1.2 kNN Method

The second method of mutual information estimation considered in this paper uses the kNN (k^{th}-Nearest Neighbor) method proposed by Kraskov et al. [8] and summarized in Algorithm 2. We draw N samples from the prior distribution $p(\theta|D_{n-1})$ computed via the Delayed Rejection Adaptive Metropolis algorithm [3, 15], detailed in Appendix A, creating a chain $X = (\theta, d_n(\xi_n))$ with which we have appended the outputs for each sample of parameters predicted according to the statistical model (1). For each chain element X_i, we compute the distance $\epsilon(i)/2 = ||X_i - X_{k(i)}||_\infty$ where $X_{k(i)}$ represents the kth nearest neighbor to X_i in the chain $\{X_i\}_{i=1}^n$, and determine the number of points in each marginal subspace n_θ and n_d that lie within $\epsilon(i)/2$ of the projected point; see Figure 1 for an example of this computation. As detailed in [8], the mutual information can approximated by

$$I(\theta; d_n|D_{n-1}, \xi_n) \approx \psi(k) - \mathbb{E}\left[\psi(n_\theta + 1) + \psi(n_d + 1)\right] + \psi(N), \tag{7}$$

where $\psi(\cdot)$ is the digamma function. For a derivation of (7), see Appendix B.

We note that the value of k does not need to be fixed. For each design tested, the mutual information may be calculated for a vector of possible k values, and the k that

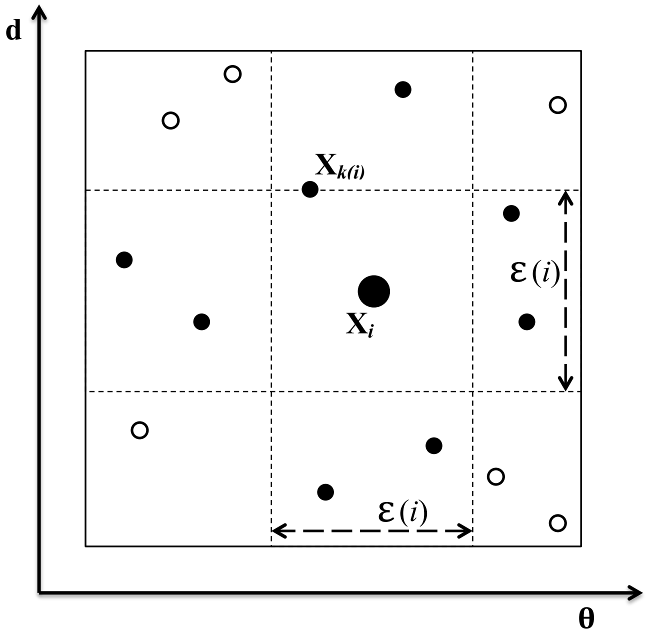


Figure 1: Calculation of $\epsilon(i)$, $n_\theta(i)$, and $n_d(i)$ for the case $k = 1$. Here we have $n_\theta(i) = 3$ and $n_d(i) = 4$. Note that the kth nearest neighbor is not included in the determination of n_θ and n_d.

Algorithm 2 kNN Method

```
(1) Fix value of $k$ (e.g., $k = 6$) and define number of $k$NN vector elements $N$ (e.g., $N = 2000$).
(2) For each potential design $\xi_n$,
    (a) Create a vector with $\dim(\theta) + \dim(d)$ rows and $N$ columns
        (i) In the first $\dim(\theta)$ rows, draw $N$ samples, $\{\theta^i\}_{i=1}^N$, from the prior distribution $p(\theta|D_{n-1})$.
        (ii) In the next $\dim(d)$ rows, place $d_n(\xi_n)$, $i = 1, ..., N$.
        (iii) Normalize the data vector,
                $X = \left\{(\mathrm{diag}(s^{-1})(X_i - \mu)\right\}_{i=1}^N$
              where $\mu = [\bar{\theta}, \bar{d}]^T$ is a $(\dim(\theta) + \dim(d)) \times 1$ vector of sample means and $s = [s_\theta, s_d]^T$ is the sample standard deviation.
    (b) For each sample $X_i$ in the vector, compute $\epsilon(i)/2 = ||X_i - X_{k(i)}||_\infty$.
    (c) For each sample $X_i$, compute $n_\theta(i) = \#$ points in $\theta$ marginal space with at least one coordinate within distance $\epsilon(i)/2$ and $n_d(i) = \#$ points in $d$ marginal space with at least one coordinate within distance $\epsilon(i)/2$.
    (d) Estimate the mutual information:
            $I(\theta; d_n|D_{n-1}, \xi_n) \approx \psi(k) - \mathbb{E}\left[\psi(n_\theta + 1) + \psi(n_d + 1)\right] + \psi(N),$
        where $\psi(\cdot)$ is the digamma function.
(3) Let $\xi^*$ be the design such that $\max_{\xi_n} I(\theta; d_n|D_{n-1}, \xi_n) = I(\theta; d_n|D_{n-1}, \xi^*)$.
```

yields the maximum mutual information value that we employ. In this paper, we fix $k = 6$ based on previous work done by Terejanu et. al [19], and save the analysis of a varying k for future work.

2 Demonstration of Mutual Information Algorithms

To demonstrate the estimation of mutual information using both the Monte Carlo and kNN methods, we consider a simple example of two Gaussian random variables, $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$. This example is chosen to illustrate the convergence of the Monte Carlo estimation to that of the kNN method when enough samples are taken. For our later examples, computing the Monte Carlo estimate with the number of samples required for convergence is often too cost-prohibitive so we demonstrate here that both methods do in fact yield the same results if the number of samples is taken to be sufficiently large. Here we compute the Monte Carlo estimate via direct integration of (5)—future work will focus on implementation of the mutual information estimate (6).

For this demonstration, the values of μ_x and μ_y are randomly selected from $\mathcal{U}(0,1)$. We randomly select a covariance matrix of the form

$$\mathrm{cov}(X,Y) = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$$

such that $\sigma_{xy} > 0$. This last requirement is due to the fact that mutual information is equal to zero if and only if the random variables are independent. As this is easy to test, we choose the covariance value σ_{xy} such that the variables X and Y are at least weakly correlated. The results included here are for mean and covariance matrices

$$\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} = \begin{bmatrix} 0.3005 \\ 0.1482 \end{bmatrix}$$

and

$$\mathrm{cov}(X,Y) = \begin{bmatrix} 0.5434 & 0.3971 \\ 0.3971 & 0.3500 \end{bmatrix}.$$

We estimate the mutual information via both the Monte Carlo and kNN algorithms, for increasing values of N, and report the results in Table 1. We observe that the Monte Carlo estimate of the mutual information converges to that provided by the kNN algorithm as the number of samples N is increased, thus verifying the accuracy of the two methods. However, the convergence of the two estimates comes at a cost. Whereas the computation time for the kNN method is nearly instantaneous in all four trials, the time required to estimate the mutual information via Monte Carlo quadrature grows exponentially as N is increased. This is due to the direct integration of (5) and future work will focus on the implementation of the estimate (6).

	# Samples	***k*NN**	**MC**	**MC Time (s)**
Trial 1	250	0.8281	1.0410	7.25
Trial 2	500	0.9027	0.9462	28.9
Trial 3	1000	0.9168	0.9338	116
Trial 4	2000	0.8987	0.9044	462

Table 1: Mutual information values with varying numbers of samples to verify the convergence of the Monte Carlo estimate to the kNN estimate. Computation times for the Monte Carlo method illustrate the exponential increase in time as the number of samples increases.

3 Examples

We provide four examples illustrating the high-to-low calibration framework detailed in Section 1. The first is a steady-state heat model for which we have experimental temperature data. In the second example, we illustrate the calibration framework for a general time-dependent diffusion model whereas in Example 3.3, we employ a 1-D kinetic diffusion equation for the high-fidelity model and a point kinetic equation for the low-fidelity model. In the final example, we illustrate the framework for a particle transport model quantifying angular flux in a 1-D slab.

3.1 Steady State Heat Model

To illustrate our methods for estimating mutual information between two random variables, we consider a steady state heat equation quantifying heat conduction in an aluminum rod of dimensions $a \times b \times L$, subjected to an ambient room temperature of T_{amb}. The thermal conductivity coefficient is denoted by K ($W/\text{cm}\cdot^\circ C$), the convective heat transfer coefficient by h ($W/\text{cm}^2 \cdot^\circ C$), and the source heat flux by Φ (W/cm^2). The model is

$$\frac{d^2T_s}{dx^2} = \frac{2(a+b)}{ab}\frac{h}{K}\left[T_s(x) - T_{amb}\right],$$

with boundary conditions

$$\begin{aligned} \frac{dT_s}{dx}(0) &= \frac{\Phi}{K}, \\ \frac{dT_s}{dx}(L) &= \frac{h}{K}\left[T_{amb} - T_s(L)\right]. \end{aligned}$$

The analytic solution to the steady state heat equation for parameter set $\phi = [\Phi, h]$ is

$$T_s(x;\phi) = c_1(\phi)e^{-\gamma x} + c_2(\phi)e^{\gamma x} + T_{amb}, \tag{8}$$

where

$$\begin{aligned} c_1(\phi) &= -\frac{\Phi}{K\gamma}\left[\frac{e^{\gamma L}(h+K\gamma)}{e^{-\gamma L}(h-K\gamma)+e^{\gamma L}(h+K\gamma)}\right], \\ c_2(\phi) &= \frac{\Phi}{K\gamma} + c_1(q), \end{aligned}$$

and $\gamma = \sqrt{\frac{2(a+b)h}{abK}}$.

Our physical system consists of an aluminum rod, heated at the secured end $x = 0$ and free at the other, of height and width $a = b = 0.95$ cm and length $L = 70$ cm. We use the thermal conductivity coefficient value of $K = 2.37$ W/cm$\cdot^\circ C$ reported for aluminum at $300^\circ C$. The independent spatial variable x_ξ is discretized into fifteen uniformly distributed points on the range $[10, 66]$ cm. Choosing a spatial location for evaluation is considered a design condition and Table 2 specifies the possible design choices. For the experimental

Design #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Location (cm)	14	18	22	26	30	34	38	42	46	50	54	58	62	66

Table 2: Possible design choices for the steady state heat equation example.

data set reported in [15], we calibrate the high-fidelity model parameters $\phi = [\Phi, h]$ for use in simulating our own experimental or synthetic data set. Specifically, we employ delayed rejection adaptive Metropolis (DRAM), as detailed in Appendix A and [15], to construct the marginal densities for Φ and h shown in Figure 2. From these densities, we draw a parameter pair $\phi = [\Phi, h]$ that is used for all high-fidelity model simulations. '

We consider a low-fidelity quadratic surrogate model

$$y = Ax^2 + Bx + C \tag{9}$$

with parameters $\theta = [A, B, C]$ and use parameter calibration to fit it to our analytic solution with as few high-fidelity model evaluations as possible. Upon selection of a design ξ_n, the high-fidelity model is evaluated at x_{ξ_n} and the resulting observation $\tilde{d}_n = T_s(x_{\xi_n}; \phi) + \tilde{\varepsilon}_n(\xi_n)$ is added to the data set for use in re-calibrating the low-fidelity model. Here we take $\tilde{\varepsilon}_n(\xi_n) \sim \mathcal{N}(0, \tilde{\sigma}^2)$ where $2\tilde{\sigma}$ is 10% of $\max_{i=1,\cdots,n-1} T_s(x_{\xi_i}; \phi)$.

We estimate the mutual information via the kNN method and compile the sequence of chosen designs in Table 3. If one has a proper prior parameter distribution, one can estimate the mutual information even in the case when the prior is "diffuse." In this example, however, we employ an noninformative and improper prior – as is very often the case when one simply has parameter bounds or conditions such as positivity — so we need enough data to ensure that the posterior $p(\theta|D_r)$ is proper. This requires that we have at least three data points to do an initial DRAM calibration, but results may be more consistent with a number of data points exceeding the dimensionality of the parameter space. Here we assume that high-fidelity data is already available at location $x = 10$. The remaining fourteen design conditions are then selected in order of decreasing mutual information yielding the results tabulated in Table 3.

Since the posterior $p(\theta|D_r)$ is not proper until we have at least three data points, we do not begin regularly updating σ in the DRAM algorithm, which controls the variability of the parameter chains, until $r = 4$. Figure 3 shows the parameter chains after several iterations of the algorithm, illustrating that they are well-mixed in the sense that they appear as white noise with no discernible patterns. Figure 4 shows the final joint posterior

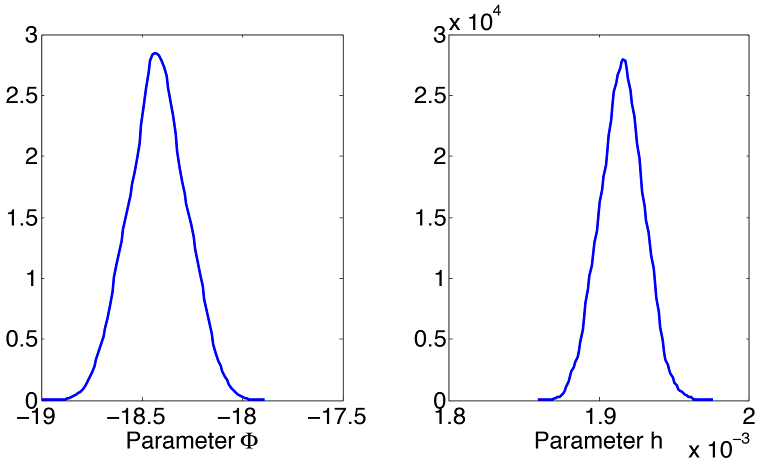


Figure 2: High-fidelity parameter distributions used to construct "experimental data".

Method	**Sequence**
kNN	14 7 1 5 2 6 3 8 4 9 13 10 12 11

Table 3: Design choice sequence for the kNN algorithm for 5000 iterations in the steady state heat equation example.

distributions after all fifteen points have been selected. Each pairwise combination of the parameters $\theta = [A, B, C]$ is highly correlated, but not to the point where parameter identifiability becomes an issue.

Figure 5 displays the evolution of the parameter distributions over the course of the 14-cycle process. In the early stages, there is a uncertainty in the parameter behavior due to the fact that at Stage 3, we have the minimal number of observations required to have a proper posterior density. As the algorithm progresses and the amount of data is increased, the posterior distributions exhibit the expected Gaussian behavior.

Having confirmed that all parameter chains are well-mixed and each parameter is identifiable with a marginal distribution that is Gaussian, we compare the final quadratic model, obtained using all fifteen calibration points, with the analytic solution in Figure 6, for the ordering of points selected via the kNN algorithm. A listing of L^2 errors is included in Table 4 to illustrate the convergence of the low-fidelity model to the high-fidelity model. Errors are large in the selection of the first few points, when there is not enough information for the low-fidelity model to "know" the correct shape. With the addition of a few extra points, there is enough data for the low-fidelity model to mimic the shape of the high-fidelity model. We note that around Step 8, the errors no are longer decreasing by any appreciable amount—our low-fidelity model is about as well-calibrated as it can be. At this point, we could choose to terminate the process and save the computation time of evaluating the high-fidelity model at the final six points.

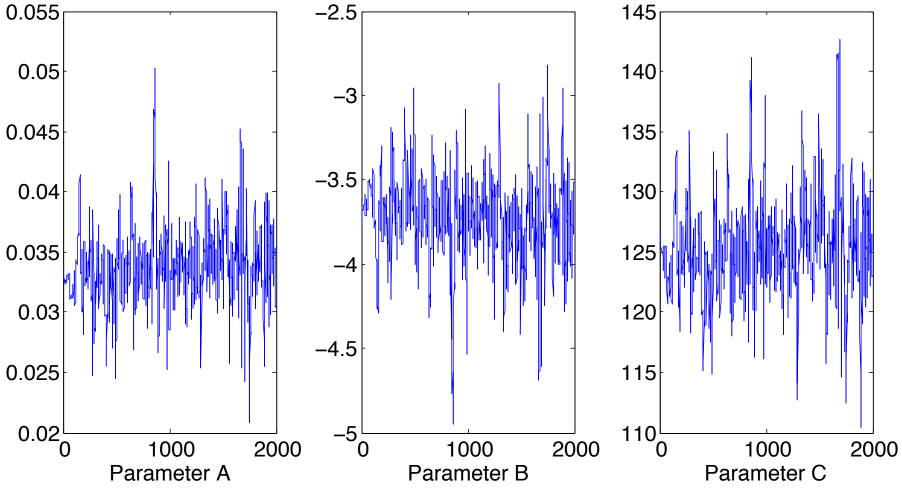


Figure 3: Well-mixed parameter chains after the selection of the first five points.

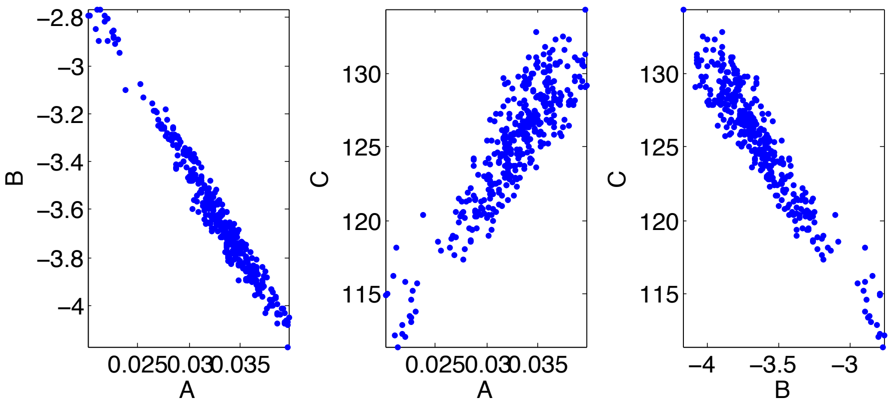


Figure 4: Joint posterior distributions for pairwise combinations of $\theta = [A, B, C]$.

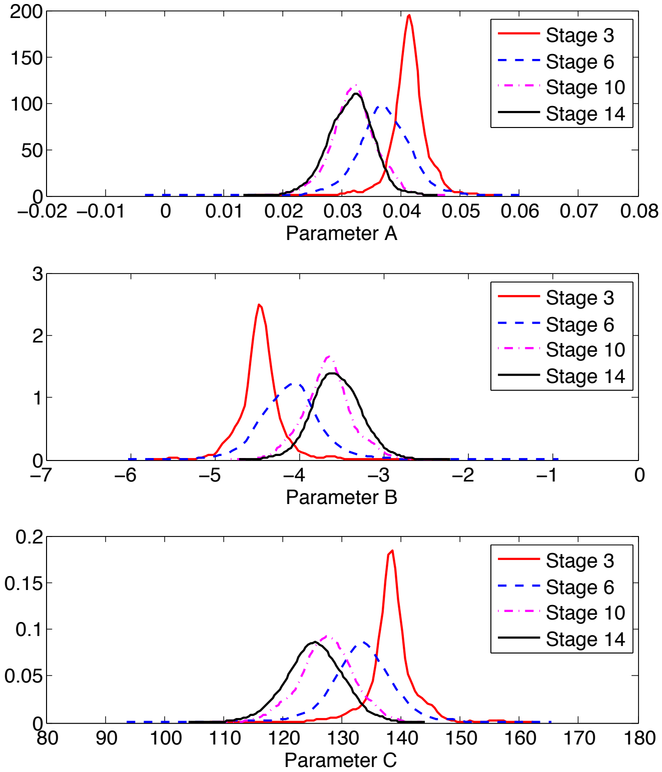


Figure 5: Evolution of parameter distributions for the steady state heat equation over 14 cycles of the kNN algorithm.

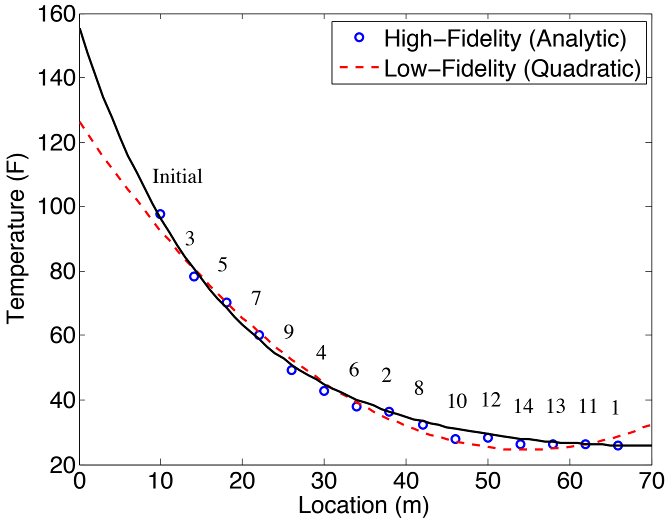


Figure 6: Fit of quadratic model to the steady state heat equation analytic solution for 15 calibration points, with ordering selected by the kNN algorithm.

Step	1	2	3	4	5	6	7
L^2 **Error**	6.31e-4	436.35	25.28	17.11	11.56	10.29	9.47
Step	8	9	10	11	12	13	14
L^2 **Error**	9.10	9.14	9.16	8.82	8.93	9.19	9.61

Table 4: L^2 errors for the steady state heat equation at each of the 14 calibration steps.

3.2 Time-Dependent Diffusion Model

We now consider the time-dependent diffusion model

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{10}$$

subject to defined initial and boundary conditions

$$\begin{aligned} u(t,0) &= u(t,2) = 0, \\ u(0,x) &= \sin(\pi x/2). \end{aligned}$$

The analytic solution of (10) is

$$u(t,x) = e^{-\alpha\pi^2 t/4} \sin\left(\frac{\pi x}{2}\right), \tag{11}$$

obtained using separation of variables. To maintain generality, we consider non-dimensional variables and employ the prior distribution $\mathcal{N}(0.7, 0.1)$ for the diffusivity α in the high-fidelity model (11). Output from this solution is used to calibrate a lower-fidelity model constructed via a finite difference approximation of (10) obtained using backward differences in time and centered differences in space. As in previous examples, we add noise

Design	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Time	1.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	3.0	3.0	3.0
Location	0.0	0.4	0.8	1.2	1.6	2.0	0.0	0.4	0.8	1.2	1.6	2.0	0.0	0.4	0.8

Design	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Time	3.0	3.0	3.0	4.0	4.0	4.0	4.0	4.0	4.0	5.0	5.0	5.0	5.0	5.0	5.0
Location	1.2	1.6	2.0	0.0	0.4	0.8	1.2	1.6	2.0	0.0	0.4	0.8	1.2	1.6	2.0

Table 5: Possible design choices for the time-dependent heat equation example.

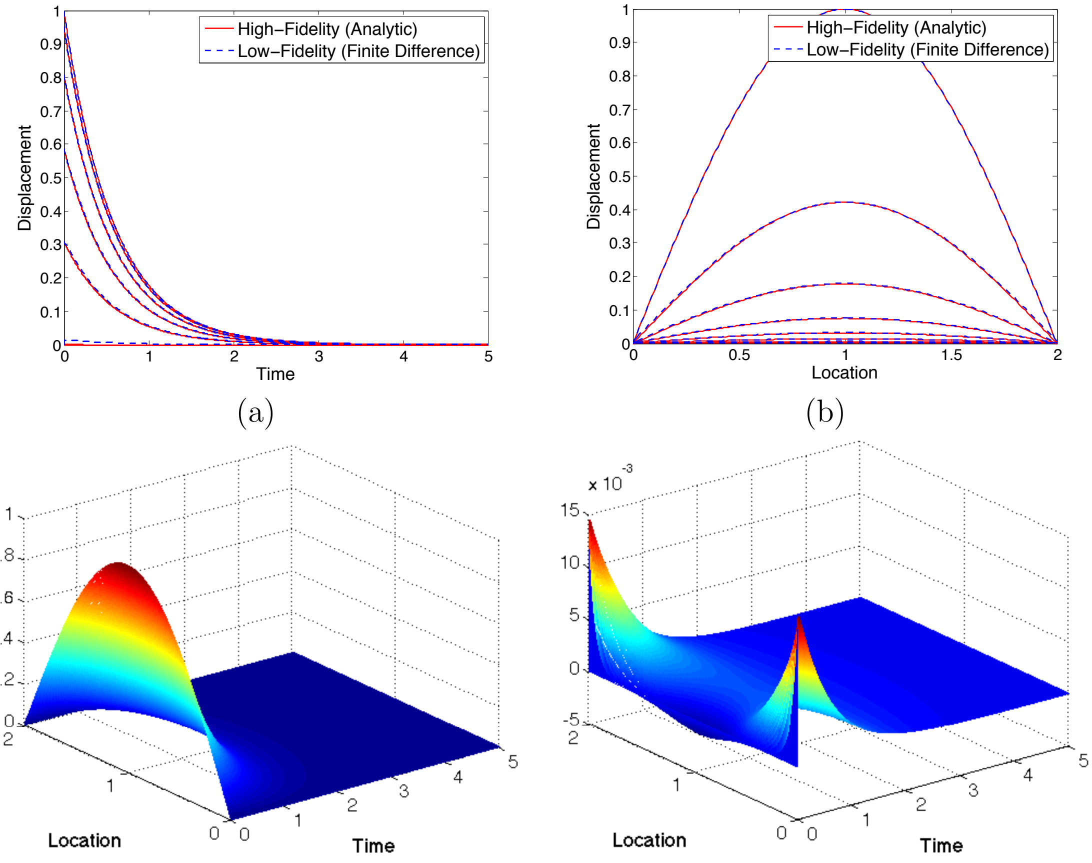
$\tilde{\varepsilon}_n(\xi_n) \sim \mathcal{N}(0, \tilde{\sigma}^2)$, where $2\tilde{\sigma}$ is 10% of $\max_{i=1,\cdots,n-1} u(t_{\xi_i}, x_{\xi_i})$, to the simulated data produced with the high-fidelity model at each design ξ_n.

The model is analyzed on a time domain of $[0, 5]$ and a spatial domain of length $[0, 2]$. Stepsizes are 5×10^{-3} in the t direction and 1×10^{-2} in the x direction. Using a grid of 30 possible design conditions—see Table 5—the points used for model calibration are chosen one-by-one based on the maximum mutual information between the parameter α and the model predictions given design inputs (t_{ξ_n}, x_{ξ_n}). The design sequence is given in Table 6 for the time dependent model, and the final fit of the model after parameter calibration with a parameter value of $\alpha = 0.7117$ is shown in Figure 7. Since this example has only a 1-dimensional parameter space, we include results from the direct quadrature of the Monte Carlo estimate (5) to further verify the kNN algorithm.

The design sequence chosen by both mutual information algorithms follows the behavior that we would expect; the points are chosen in order of increasing t values, corresponding to the exponential temporal decay of the solution. Within each time classification, the points are generally chosen from the ends of the spatial interval in towards the center, again following the leveling off of the slope in the spatial domain. Points occurring in a region with a larger slope value in magnitude are more likely to be chosen because of the larger discrepancy in their output values which contributes a more significant information gain relative to other points. This concurs with the engineering practice of sampling in regions having large gradients to observe increased sensitivity.

Method	**Sequence**
MC	1 6 2 5 3 4 7 12 8 11 9 10 13 18 14 17 15 16 19 24 20 23 21 22 30 25 26 29 27 28
kNN	1 6 2 5 3 4 7 12 8 11 9 10 13 18 14 17 15 16 19 24 20 23 21 22 30 25 26 29 27 28

Table 6: Comparison of design sequences provided by the Monte Carlo and kNN methods for the time-dependent heat equation example.



(a) (b)

(c) (d)

Figure 7: High-fidelity solution versus calibrated low-fidelity time-dependent heat models in (a) time domain [0, 5] for eleven uniformly spaced values of x on the interval [0, 2], (b) spatial domain [0, 2] for eleven uniformly spaced values of t on the interval [0, 5], and (c) 3-dimensional domain. (d) Difference between the low- and high-fidelity solutions.

3.3 Neutron Diffusion Model

In this example, we employ the 1-D kinetic neutron diffusion equation

$$
\begin{aligned}
&\frac{1}{\nu_{th}}\frac{\partial \phi}{\partial t} - \frac{1}{3\Sigma_{sc}}\frac{\partial^2 \phi}{\partial x^2} = \Sigma_f \phi n_f - \Sigma_a \phi n_a \quad , \; 0 < x < L, t > 0 \\
&\phi(t,0) = \phi(t,L) = 0 \quad , \; t > 0 \\
&\phi(0,x) = 1 \quad , \; 0 < x < L
\end{aligned}
\tag{12}
$$

as the high-fidelity model. Here ϕ (neutrons/s·m^3) is the flux and Σ_{sc}, Σ_f, and Σ_a respectively represent the scattering, fission, and absorption cross sections with units of (1/m), (fission/m), and (absorption/m). The constants ν_{th}, n_f, and n_a have units of (m/s), (neutrons/fission), and (neutrons/absorption). We employ the values

$$
\begin{aligned}
\nu_{th} &= 2 \times 10^3 \ , \ n_f = 2.2 \ , \quad n_a = 1 \\
\Sigma_{sc} &= 8 \times 10^2 \ , \ \Sigma_a = 4.2 \ , \quad \Sigma_f = 23
\end{aligned}
$$

from [10].

Because we cannot easily obtain an analytic solution for (12), we approximate the solution using a finite-element discretization of the weak model formulation. To construct a weak formulation, we multiply (12) by test functions $\eta(x) \in H_0^1(0,L)$, the Sobolev space of functions that are at least once differentiable on in the interval $[0,L]$, and integrate in space to obtain

$$\frac{1}{\nu_{th}}\int_0^L \frac{\partial \phi}{\partial t}\eta(x)dx - \frac{1}{3\Sigma_{sc}}\int_0^L \frac{\partial^2 \phi}{\partial x^2}\eta(x)dx = (\Sigma_f n_f - \Sigma_a n_a)\int_0^L \phi\eta(x)dx, \tag{13}$$

which must hold for all $\eta(x) \in H_0^1(0,L)$.

We approximate the solution $\phi(t,x)$ using the representation

$$\phi^N(t,x) = \sum_{j=1}^{N-1} \varphi_j(t)\eta_j(x) \tag{14}$$

where we employ the piecewise splines

$$\eta_j(x) = \frac{1}{h}\begin{cases} x - x_{j-1} & , \quad x_{j-1} \leqslant x < x_j \\ x_{j+1} - x & , \quad x_j \leqslant x < x_{j+1} \\ 0 & , \quad \text{else} \end{cases}$$

as spatial basis functions. For $L = 2$, we obtained converged solutions with $N = 25$ which yields the stepsize $h = 2/25$. Substitution of (14) into (13), use of the basis functions as test functions, and integration by parts yields the system

$$\frac{d\vec{\varphi}}{dt} = V_0^{-1}\left[\nu_{th}\left((\Sigma_f - \Sigma_a)V_0 - \frac{1}{3\Sigma_{sc}}V_1\right)\right]\vec{\varphi}(t) \tag{15}$$

of ordinary differential equations where $\vec{\varphi}(t) = [\varphi_1(t), \cdots, \varphi_{N-1}(t)]$. The $(N-1)\times(N-1)$ matrices V_0 and V_1 are given by

$$V_0 = \int_0^L \eta_i\eta_j dx = \frac{h}{6}\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \end{bmatrix}$$

and

$$V_1 = \int_0^L \eta_i'\eta_j' dx = \frac{1}{h}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \end{bmatrix}.$$

We numerically integrate the system (15) to obtain $\vec{\varphi}(t)$ and evaluate (14) at $\bar{x} = 1$ to obtain our high-fidelity solution. Observations are given by $\tilde{d}_n = \phi^N(t_{\xi_n}, \bar{x}) + \tilde{\varepsilon}_n(\xi_n)$, where $\tilde{\varepsilon}_n(\xi_n) \sim \mathcal{N}(0, \tilde{\sigma}^2)$ with $2\tilde{\sigma}$ taken to be 10% of $\max_{i=1,\cdots,n-1} \phi^N(t_{\xi_i}, \bar{x})$.

Design #	1	2	3	4	5	6	7	8	9	10
Time (s)	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00

Table 7: Possible design conditions for the neutronics example.

We employ the point kinetic equation

$$\frac{dn}{dt} = \frac{(r-1)}{\beta} n(t), \tag{16}$$

as a low-fidelity model where r represents the reactivity, β is a function of shape, and n has units of (neutrons/m^3). Model (16) is compared to the high-fidelity model (15) to calibrate the parameters $\theta = [r, \beta]$. The prior distributions for r and β were taken to be $r \sim \mathcal{U}(0, 2)$ and $\beta \sim \mathcal{U}(0, 250)$.

As before, we identify a set of design conditions from which our calibration points will be selected. The remaining independent variable, time, is discretized into a set of ten values listed in Table 7.

Choosing these calibration points one at a time in order of decreasing mutual information, we obtain the parameter values $r = 1.63$ and $\beta = 172.65$ that most accurately fit the low-fidelity model to the high-fidelity model. The resulting fit is shown in Figure 8. Table 8 contains the estimated mutual information values for the kNN method.

We show the evolution of the parameter distributions as additional calibration points are selected in Figure 9. As the number of calibration points increases, the parameter distributions become more stable and increasingly more Gaussian in shape. It is observed that the visual fit in Figure 8 is highly accurate. We are currently investigating the use of energy statistics to quantitatively test the null hypothesis that two independent samples of parameters are derived from the same probability distribution [17].

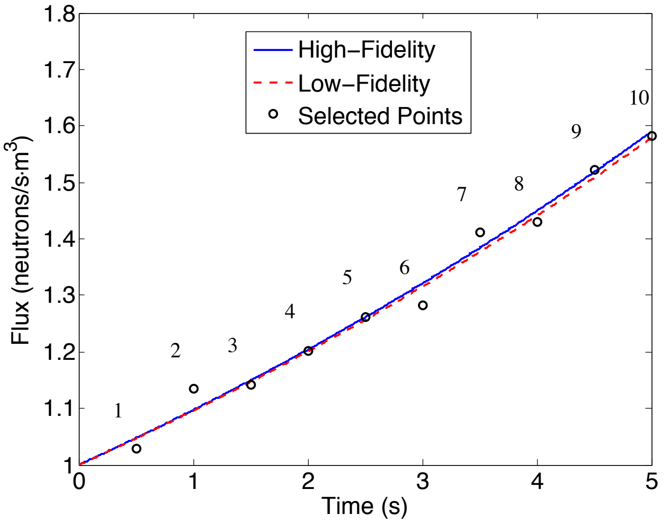


Figure 8: High-fidelity versus low-fidelity models for the neutronics example.

Time	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0.5	**2.899**	—	—	—	—	—	—	—	—	—
1.0	2.881	**1.606**	—	—	—	—	—	—	—	—
1.5	2.852	1.598	**2.743**	—	—	—	—	—	—	—
2.0	2.809	1.589	2.741	**1.374**	—	—	—	—	—	—
2.5	2.755	1.583	2.737	1.372	**0.890**	—	—	—	—	—
3.0	2.692	1.576	2.739	1.371	0.889	**0.665**	—	—	—	—
3.5	2.621	1.569	2.736	1.369	0.889	0.664	**0.468**	—	—	—
4.0	2.545	1.563	2.736	1.367	0.888	0.664	0.467	**0.368**	—	—
4.5	2.463	1.554	2.734	1.365	0.887	0.664	0.467	0.367	**0.253**	—
5.0	2.375	1.543	2.733	1.363	0.886	0.663	0.467	0.367	0.252	**0.359**

Table 8: Estimated mutual information values and sequential design sequence for the neutronics example from the kNN method.

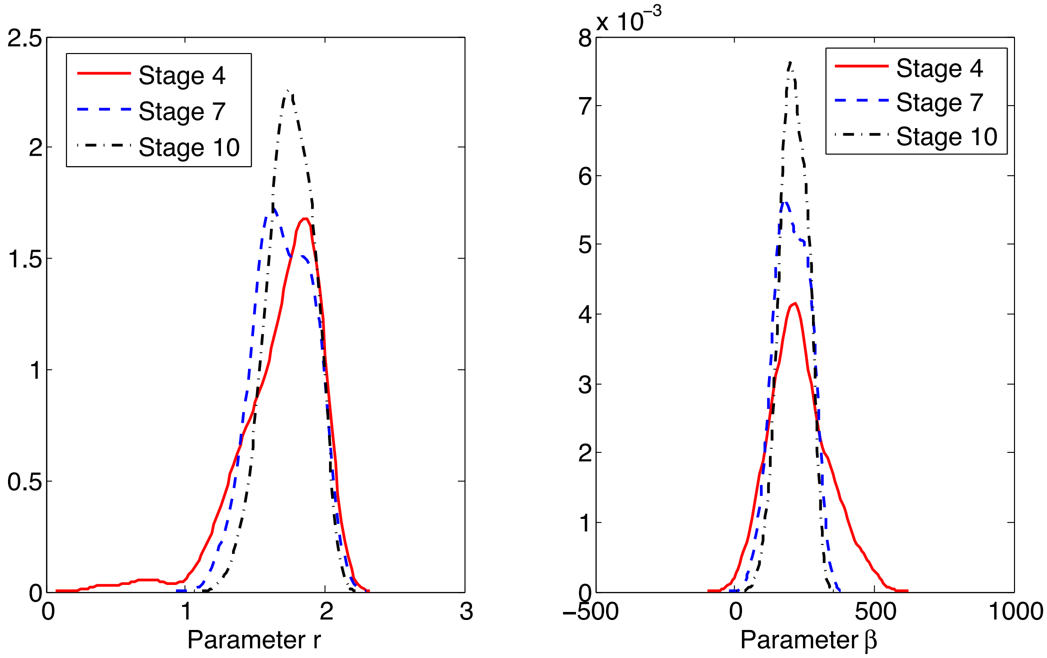


Figure 9: Evolution of parameter distributions for the neutronics example over 10 cycles.

3.4 Particle Transport Model

Finally, we consider an example investigated in the context of the method of manufactured universes with examination of a particle transport “universe”. These types of particle transport calculations are important in many applications, including nuclear reactors or high energy-density laboratory experiments [16]. In these applications, it is often prohibitively expensive to obtain experimental measurements. We employ a high-fidelity model to represent the “reality” of our universe—from which “experimental measurements” are obtained—and use this to calibrate the parameters in a low-fidelity model for production of simulated results, which serves as an approximation to the high-fidelity model in situations where evaluation of the high-fidelity model is too expensive.

For our high-fidelity model, we assume that the particle transport universe behaves according to the S_N discrete ordinates method—we choose $N = 8$—whose governing equation in 1D slab geometry with no volumetric source is [2]

$$\mu_m \frac{d\Psi_m(x)}{dx} + \Sigma_t \Psi_m(x) - \frac{c\Sigma_t}{2} \sum_{n=1}^{N} \Psi_n(x) w_n = 0 \tag{17}$$

where $\Psi_m(x)$ represents the angular flux in the mth quadrature point at slab length x, and the cosines μ and weights w are given by the Gauss-Legendre quadrature set of N points on the interval $(-1, 1)$. The total cross-section and scattering ratio are respectively taken to be $\Sigma_t = 1.00$ cm^{-1} and $c = \Sigma_s/\Sigma_t = 0.99$.

The analytic solution of model (17) is

$$\Psi_m(x) = \sum_{k=1}^{N} A_k \frac{\nu_k}{\nu_k - \mu_m} \exp\left(\frac{-\Sigma_t x}{\nu_k}\right) \tag{18}$$

where each ν_k satisfies the condition

$$\frac{2}{c} = \sum_{m=1}^{N} w_m \frac{\nu}{\nu - \mu_m}.$$

The constant coefficients A_k, $k = 1, ..., N$, are obtained via satisfaction of the boundary and interface conditions

$$\Psi_m(0) = 1.0 \ , \ \mu_m > 0$$
$$\Psi_m(L_s) = 0.0 \ , \ \mu_m < 0$$

where L_s is the thickness of the slab.

Using the analytic solution (18), we produce "experimental data," with measurement noise $\tilde{\varepsilon}$, at a set of six different slab lengths, $x = [1, 2, 4, 8, 16, 32]$ cm. Our quantity of interest is the reflected particle flow rate,

$$Y_{exp} = j^-(x = 0) = \sum_{\substack{m \\ \mu_m < 0}} |\mu_m| \Psi_m(0) w_m. \tag{19}$$

For our low-fidelity model, we utilize the diffusion equation in a finite 1D slab. The governing equation

$$\frac{d^2\Phi(x)}{dx^2} - \frac{1}{L^2}\Phi(x) = 0 \ , \ 0 \leqslant x \leqslant L_s \tag{20}$$

where $L = \sqrt{D/\Sigma_a}$ is the diffusion length for diffusion coefficient D and absorption cross-section Σ_a, has analytic solution

$$\Phi(x) = C_1 \sinh\left(\frac{x}{L}\right) + C_2 \cosh\left(\frac{x}{L}\right). \tag{21}$$

The C coefficients are chosen to satisfy the known incident particle flow rates on each slab boundary from the high-fidelity model,

$$\begin{aligned} j^+(0) &= \frac{\Phi(0)}{4} + \frac{J(0)}{2} = \sum_{\substack{m \\ \mu_m>0}} \Psi_m(0)\mu_m w_m \\ j^-(L_s) &= \frac{\Phi(L_s)}{4} - \frac{J(L_s)}{2} = \sum_{\substack{m \\ \mu_m<0}} \Psi_m(L_s)|\mu_m| w_m \end{aligned}$$

where $J(x) = -Dd\Phi/dx$.

Again, we measure the reflected particle flow rate at each of the six slab lengths as our quantity of interest, given by the low-fidelity model as

$$Y_{sim} = j^-(x=0) = \frac{\Phi(0)}{4} - \frac{J(0)}{2}. \tag{22}$$

The high-fidelity experimental measurements obtained from (19) are used to calibrate the parameter set $\theta = [D, \Sigma_a]$ in the low-fidelity model (21). For calibration, we use the DRAM algorithm with an initial parameter guess of $\theta = [0.333, 0.010]$ and 10,000 iterations. We throw out the first 7,000 iterations during which the parameter chains are "burned in" and use the final 3,000 iterations as a sample from the joint posterior distribution of our parameters. Figure 10 illustrates the fit between the high and low-fidelity models, comparing experimental and simulated reflected particle flow rates (19) and (22), respectively, for each of the six slab widths. The 95% prediction and credible intervals

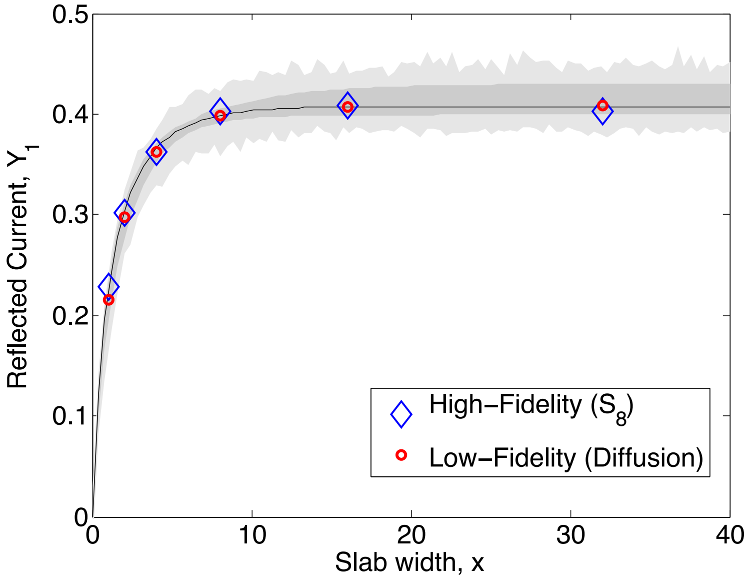


Figure 10: High-fidelity simulated experimental data versus low-fidelity measurements for the particle transport model. Note the widening of the 95% credible interval (indicated in dark gray) and the 95% prediction interval (indicated in light gray) as the distance between measurements increases.

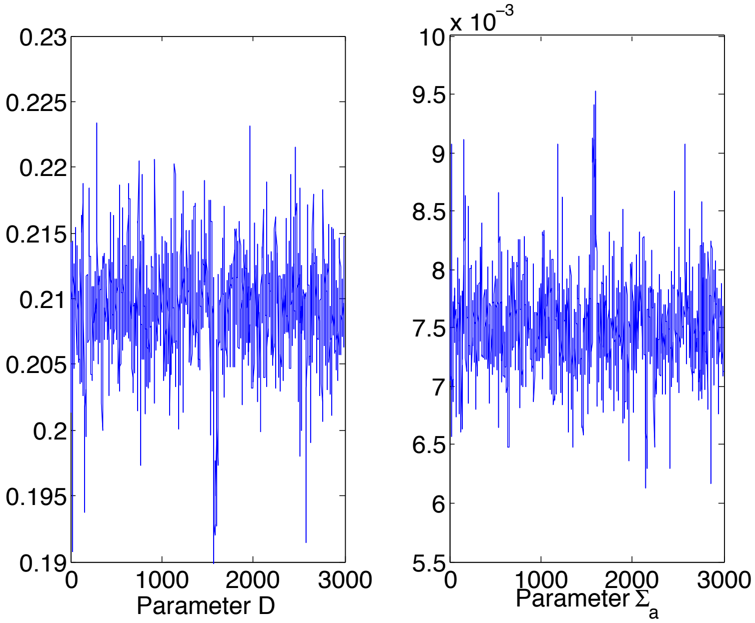


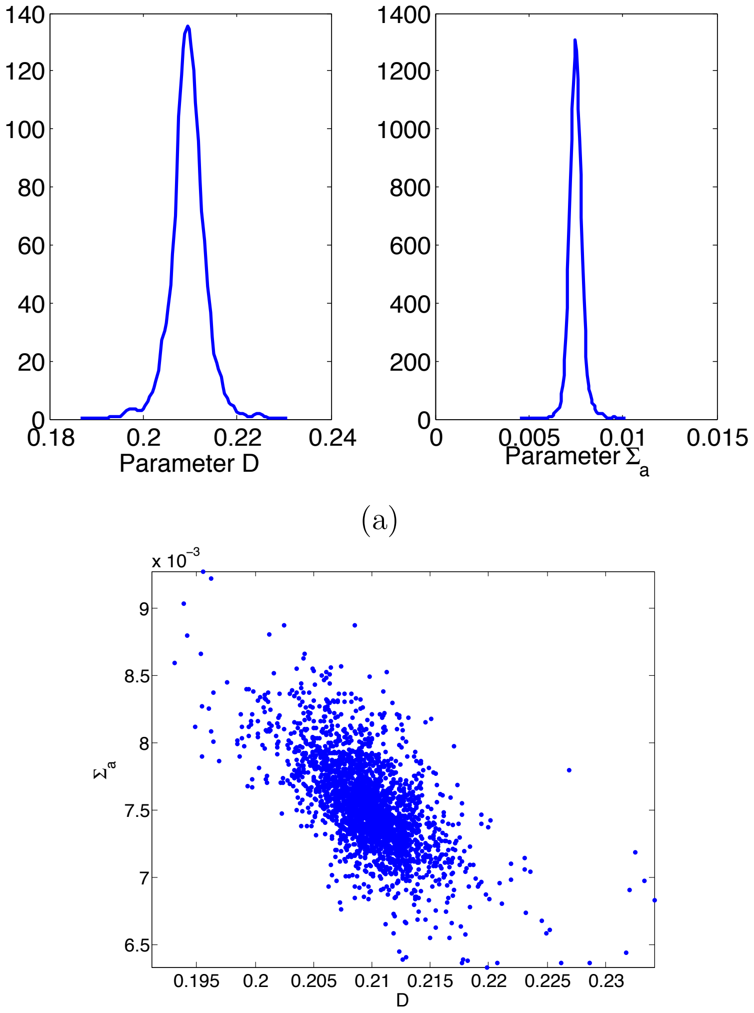
Figure 11: Well-mixed parameter chains for the particle transport example.

are plotted with the data to illustrate the likely locations of future model observations taken at slab widths other than those used for "training" the low-fidelity model. Measurement noise ε_n is added to the statistical model, causing a widening of the prediction interval as the slab width increases and information becomes more scarce. In Figures 11 and 12, we show the well-mixed DRAM chains and marginal posterior densities, respectively, for the parameters D and Σ_a.

Having calibrated our parameter set $\theta = [D, \Sigma_a]$ so that the low-fidelity model is able to accurately quantify the behavior of the high-fidelity model at our "training" set of slab widths, $x = [1, 2, 4, 8, 16, 32]$ cm, we can now use (21) in place of (18) to predict experimental data at slab widths outside of our calibration set, cutting down on our computational expenses for future predictions. For this same example, [16] discusses the construction of an emulator for the reflected particle flow rate that can be used to construct 95% prediction intervals for future observations taken at a slab width x outside of our training set.

4 Conclusion

The focus of this investigation has been to study an information-theoretic approach to calibrate low-fidelity models using a limited number of high-fidelity code evaluations. As a problem becomes more complex in nature, it proves more likely that the measurement of experimental data or the evaluation of the model that represents the physicality of the system will prove to be computationally expensive or infeasible. Therefore, we emphasize the importance of developing lower-fidelity, cost-efficient models that accurately depict the characteristics of the original system. To that end, we also explore more efficient methods of experimental design in the Bayesian framework, using a measure of statistical dependence between a set of model parameters and the corresponding prediction outputs to select



(a)

(b)

Figure 12: (a) Marginal parameter densities for the particle transport model, calculated from the final 3,000 iterations in a 10,000 iteration DRAM run, and (b) joint posterior distribution for the parameter set $\theta = [D, \Sigma_a]$.

designs in an order that will maximize the information gain while minimizing the number of evaluations required for calibration purposes. Whereas in this investigation we chose to specify a fixed number of design conditions to be used for calibration of the low-fidelity model, one could choose to terminate the iteration procedure when the information gain decreases below a specified tolerance. While one can begin the process with no high-fidelity "data" available, the mutual information estimates are much more accurate towards the beginning of the process if several data points are already present—a number exceeding the dimensionality of the parameter space is preferred.

The result of this process is the construction of a low-fidelity model that exhibits the primary characteristics of the high-fidelity model necessary to fit the provided data. The evaluation of this low-fidelity model is much more efficient in both time and computational costs. The uncertainties present in the new model may be quantified so that future observations may be predicted with known levels of uncertainty as illustrated by the prediction intervals in Figure 10.

We note that the kNN analysis requires repeated evaluation of the low-fidelity model to determine optimal evaluation conditions ξ_n. For computationally intense models, this direct evaluation may be infeasible. In such cases, surrogate models may be required to determine the optimal evaluation designs at which one subsequently evaluates the high-fidelity model.

We note that by using a validated high-fidelity code to calibrate parameters in the low-fidelity model, one can cover the state space that is relevant to the quantities of interest that we ultimately want to predict. Whereas experimental data is preferable, we often do not have the option of collecting it throughout the state space.

Appendix A: DRAM Algorithm

The Delayed Rejection Adaptive Metropolis algorithm (DRAM) [3, 15] was developed in response to a need for a more efficient means of sampling a posterior density. In a basic Metropolis algorithm, a Markov chain is constructed such that its stationary distribution is the posterior distribution of a parameter by proposing certain parameter values that explore the geometry of the distribution. Haario et al. [3] furthered this approach by including both an adaptive step that updates the proposal covariance matrix as further information is gained about the parameter space and a "delayed rejection" step that constructs an alternative parameter candidate which will induce greater mixing if the first candidate is rejected rather than retaining the initial estimate; this is the foundation of the DRAM algorithm. The basic algorithm for DRAM is given in Algorithms 3 and 4; for more details on the implementation of DRAM see [3, 15].

Note: The choice of $\gamma_2 = \frac{1}{5}$ in Algorithm 4 is common, but other values are reasonable. The choice of $\gamma_2 < 1$ ensures that the second-stage proposal function is narrower than the first, increasing the mixing of the chain.

Appendix B: Derivation of the kNN Algorithm

Here we present the derivation of the kNN approximation to mutual information (7) as developed in [8] by Kraskov et al. We generalize to any two random variables X and Y—this can easily be adapted to measure the information between a random variable and a set of model parameters, as done in the above investigation.

We begin with the Shannon entropy [13] of a random variable X with density $\mu(x)$,

$$H(X) = -\int \mu(x)\log(\mu(x))dx \tag{23}$$

where the log function is taken to be the natural log for the remainder of this appendix. We can view (23), up to the minus sign, as the average of $\log(\mu(x))$ over the domain of

Algorithm 3 Delayed Rejection Adaptive Metropolis [3, 15]

```
(1) Set design parameters $n_s$,$s_p$, $\sigma_s^2$, $k_0$ and number of chain iterates $M$
(2) Determine $\theta^0$ =arg $\min_\theta \sum_{i=1}^n [v_i - f_i(\theta)]^2$
(3) Set $SS_{\theta^0} = \sum_{i=1}^n [v_i - f_i(\theta^0)]^2$
(4) Compute initial variance estimate $s_0^2 = \frac{SS_{\theta^0}}{n-p}$ for $n$ the number of data points and $p$
    the number of parameters
(5) Construct covariance estimate $V = s_0^2 \left[\chi^T(\theta^0)\chi(\theta^0)\right]^{-1}$ and $R = \text{chol}(V)$ where the
    sensitivity matrix has components $\chi_{ik}(\theta^0) = \frac{\partial f_i(\theta^0)}{\partial \theta_k}$
(6) For $k = 1, ..., M$
    (a) Sample $z_k \sim \mathcal{N}(0, I)$
    (b) Construct candidate $\theta^* = \theta^{k-1} + R^T z_k$
    (c) Sample $u_\alpha \sim \mathcal{U}(0, 1)$
    (d) Compute $SS_{\theta*} = \sum_{i=1}^n [v_i - f_i(\theta^*)]^2$
    (e) Compute $\alpha(\theta^*|\theta^{k-1}) = \min\left(1, e^{-[SS_{\theta*} - SS_{\theta^{k-1}}]/2s_{k-1}^2}\right)$
    (f) If $u_\alpha < \alpha$,
            Set $\theta^k = \theta^*$, $SS_{\theta^k} = SS_{\theta*}$
        else
            Enter Delayed Rejection Algorithm 4
        endif
    (g) Update $s_k^2 \sim$ Inv-gamma$(a_{val}, b_{val})$, where
            $a_{val} = 0.5(n_s + n)$ , $b_{val} = 0.5(n_s \sigma_s^2 + SS_{\theta^k})$
    (h) If  mod $(k, k_0) = 1$
            Update $V_k = s_p \text{cov}(\theta^0, \theta^1, ..., \theta^k)$
        else
            $V_k = V_{k-1}$
        endif
    (i) Update $R_k = \text{chol}(V_k)$
```

the x variable. For a set of unbiased estimators $\widehat{\log(\mu(x_i))}$, we create the estimator

$$\hat{H}(X) = -\frac{1}{N}\sum_{i=1}^{N} \widehat{\log(\mu(x_i))},$$

Algorithm 4 Delayed Rejection Component of DRAM [3, 15]

```
(1) Set the design parameter $\gamma_2 = \frac{1}{5}$
(2) Sample $z_k \sim \mathcal{N}(0, I)$
(3) Construct second-stage candidate $\theta^{*2} = \theta^{k-1} + \gamma_2 R_k^T z_k$
(4) Sample $u_\alpha \sim \mathcal{U}(0, 1)$
(5) Compute $SS_{\theta^{*2}} = \sum_{i=1}^n [v_i - f_i(\theta^{*2})]^2$
(6) Compute
        $\alpha_2(\theta^{*2}|\theta^{k-1}, \theta^*) = \min\left(1, \frac{\pi(\theta^{*2}|v)J_1(\theta^*|\theta^{*2})[1-\alpha(\theta^*|\theta^{*2})]}{\pi(\theta^{k-1}|v)J_1(\theta^*|\theta^{k-1})[1-\alpha(\theta^*|\theta^{k-1})]}\right)$,
    where $J_1(\theta_p|\theta_c)$ is the first stage proposal distribution from (6a-b) of Algorithm 3 for
    the move from $\theta_c$ to $\theta_p$
(7) If $u_\alpha < \alpha$,
        Set $\theta^k = \theta^{*2}$ , $SS_{\theta^k} = SS_{\theta^{*2}}$
    else
        Set $\theta^k = \theta^{k-1}$ , $SS_{\theta^k} = SS_{\theta^{k-1}}$
    endif
```

which estimates $H(X)$ from a random sample $\{x_1, x_2, ..., x_N\}$ of N realizations of X. To obtain the $\widehat{\log(\mu(x_i))}$ estimators, consider a probability distribution $P_k(\epsilon)$ for the distance between the point x_i and it's kth-nearest neighbor. The probability $P_k(\epsilon)d\epsilon$ is the chance that one point falls within a distance $r \in [\epsilon/2, \epsilon/2 + d\epsilon/2]$ from x_i, $k-1$ points are at distances smaller than $\epsilon/2$ from x_i, and $N-k-1$ points are at distances larger than $\epsilon/2 + d\epsilon/2$ from x_i. We denote the mass of the ϵ-ball centered at x_i by $p_i(\epsilon)$; that is,

$$p_i(\epsilon) = \int_{||\xi - x_i|| < \epsilon/2} \mu(\xi) d\xi.$$

Then the probability density of the distance ϵ between x_i and its kth nearest neighbor can be expressed as

$$P_k(\epsilon) = k \binom{N-1}{k} \frac{dp_i(\epsilon)}{d\epsilon} p_i(\epsilon)^{k-1} (1 - p_i(\epsilon))^{N-k-1}. \tag{24}$$

Note that

$$\int P_k(\epsilon) d\epsilon = 1;$$

that is, $P_k(\epsilon)$ is correctly normalized. Using (24), we compute the expectation of the mass of an ϵ-ball centered at x_i with respect to the distribution of the distance to its kth nearest neighbor

$$\begin{aligned} \mathbb{E}(\log(p_i(\epsilon))) &= \int_0^\infty P_k(\epsilon)\log(p_i(\epsilon))d\epsilon \\ &= k\binom{N-1}{k}\int_0^1 \log(p)p^{k-1}(1-p)^{N-k-1}dp \\ &= \psi(k)-\psi(N), \end{aligned} \tag{25}$$

where $\psi(x)$ is the digamma function, $\psi(x) = \frac{1}{\Gamma(x)}\frac{d\Gamma(x)}{dx}$. To obtain the estimator for $\log(\mu(x_i))$, we assume that $\mu(x_i)$ is constant on the entire ϵ-ball. The mass of the ϵ-ball can then be estimated by

$$p_i(\epsilon) \approx c_d\epsilon^d\mu(x_i)$$

where d is the dimensionality of x and c_d is the volume of the d-dimensional unit ball. By taking the expectation of $\log(p_i(\epsilon))$, and isolating $\log(\mu(x_i))$, we obtain

$$\begin{aligned} \log(\mu(x_i)) &\approx \mathbb{E}[\log(p_i(\epsilon))] - \log(c_d) - d\mathbb{E}(\log(\epsilon)) \\ &= \psi(k)-\psi(N)-\log(c_d)-d\mathbb{E}(\log(\epsilon)). \end{aligned}$$

We note that the expectation of $\log(\epsilon)$ can be estimated by the mean of the N sampled values $\log(\epsilon(i))$,

$$\mathbb{E}[\log(\epsilon)] \approx \frac{1}{N}\sum_{i=1}^N \log(\epsilon(i)).$$

Therefore,

$$\hat{H}(X) = -\psi(k)+\psi(N)+\log(c_d)+\frac{d}{N}\sum_{i=1}^N \log(\epsilon(i)). \tag{26}$$

Equation (26) is known as the Kozachenko-Leonenko estimator for the Shannon entropy of a random variable X [7]. We can adapt this for multiple random variables as follows. Let $Z = (X, Y)$ be a joint random variable with maximum norm,

$$||z-z'|| = \max\{||x-x'||, ||y-y'||\}.$$

Equations (24) and (25) still hold. We change only the dimensionality d to $d_z = d_x + d_y$ and the volume of the d_z-dimensional 1-ball to $c_{d_z} = c_{d_x}c_{d_y}$, a result of Z having the maximum norm. It then follows that

$$\hat{H}(X,Y) = -\psi(k)+\psi(N)+\log(c_{d_x}c_{d_y})+\frac{d_x+d_y}{N}\sum_{i=1}^N \log(\epsilon(i)). \tag{27}$$

Suppose the kth nearest neighbor of z_i is furthest in one of the x directions. Then, $\epsilon(i)/2$ is the distance to the $(n_x(i)+1)$-st neighbor of x_i and

$$\hat{H}(X) = -\frac{1}{N}\sum_{i=1}^{N}\psi(n_x(i)+1) + \psi(N) + \log(c_{d_x}) + \frac{d_x}{N}\sum_{i=1}^{N}\log(\epsilon(i)). \tag{28}$$

Similarly, we take

$$\hat{H}(Y) = -\frac{1}{N}\sum_{i=1}^{N}\psi(n_y(i)+1) + \psi(N) + \log(c_{d_y}) + \frac{d_y}{N}\sum_{i=1}^{N}\log(\epsilon(i)). \tag{29}$$

We note that (29) is biased since the kNN of z_i was assumed to be furthest in one of the x directions, the distance $\epsilon(i)/2$ is actually larger than the distance to the $(n_y(i)+1)$-st neighbor. However, (29) is still considered a good approximation for $H(Y)$ since it becomes exact as $N \to \infty$ [8]. Combining (27), (28), and (29), we obtain our estimate for the mutual information between random variables X and Y:

$$\begin{aligned} I(X;Y) &\approx \hat{H}(X) + \hat{H}(Y) - \hat{H}(X,Y) \\ &= \psi(k) - \frac{1}{N}\left[\sum_{i=1}^{n}\psi(n_x(i)+1) + \sum_{i=1}^{n}\psi(n_y(i)+1)\right] + \psi(N). \end{aligned}$$

Acknowledgements

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (`http://www.casl.gov`), an Energy Innovation Hub (`http://www.energy.gov/hubs`) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725. North Carolina State University (NCSU) and Los Alamos National Laboratory (LANL) are core CASL partners.

References

[1] Bryant, C., and Terejanu, G., “An Information-Theoretic Approach to Optimally Calibrate Approximate Models”, *American Institute of Aeronautics and Astronautics*, 2012.

[2] Case, K.M., and Zweifel, P.F., 1967, *Linear Transport Theory*, Addison-Wesley, Reading, MA.

[3] Haario, H., Laine, M., Mira, A., and Saksman, E., “DRAM: Efficient adaptive MCMC”, *Statistics and Computing*, 16(4): 339-354, 2006.

[4] Huan, X., and Marzouk, Y. M., “Simulation-based optimal Bayesian experimental design for nonlinear systems”, *Journal of Computational Physics*, 232(1): 288-317, 2013.

[5] Kennedy, M.C., and O’Hagan, A., ”Predicting the output from a complex computer code when fast approximations are available”, *Biometrika*, 87: 1-13, 2000.

[6] Kennedy, M.C., and O'Hagan, A., "Bayesian calibration of computer models", *Journal of the Royal Statistical Society*, Series B 63: 425-450, 2001.

[7] Kozachenko, L.F., and Leonenko, N.N., "Sample Estimate of the Entropy of a Random Vector", *Problems of Information Transmission*, 23(2): 95-101, 1987.

[8] Kraskov, A., Stogbauer, H., and Grassberger, P., "Estimating mutual information", *Physical Review E*, 69(6, 2004.

[9] Liepe, J., Filippi, S., Komorowski, M., and Stumpf, M.P.H., "Maximizing the Information Content of Experiments in Systems Biology", *PLOS Computational Biology*, 9(1), Jan. 2013.

[10] Mousseau, V.A., "A fully implicit, second order in time, simulation of a nuclear reactor core", *Proceedings of ICONE 14, International Conference on Nuclear Engineering*, 2014.

[11] Schafer, J.P., and Abdel-Khalik, H.S., "Subspace Multi-Scale Approach for Reactor Analysis, Part 1: Energy Variable", *International Conference on Mathematics, Computational Methods & Reactor Physics*, 2009.

[12] Sebastini, P., and Wynn, H.P., "Maximum entropy sampling and optimal Bayesian experimental design", *Journal of the Royal Statistical Society*, Series B 62(1): 145-157, 2000.

[13] Shannon, C.E., "A Mathematical Theory of Communication", *Bell System Technical Journal*, Vol. 27: 379-423, 1948.

[14] Smith, K.S., "Assembly Homogenization Techniques for Light Water Reactor Analysis", *Progress in Nuclear Energy*, Vol. 17(3): 303-335, 1986.

[15] Smith, R.C., 2014, *Uncertainty Quantification: Theory, Implementation, and Applications*, SIAM, Philadelphia, PA.

[16] Stripling, H.F., Adams, M.L., McClarren, R.G., and Mallick, B.K., "The Method of Manufacturing Universes for validating uncertainty quantification methods", *Reliability Engineering and System Safety*, 96(9): 1242-1256, Jan. 2011.

[17] Székely, G.L., and Rizzo, M.L., "Energy Statistics: A class of statistics based on distances," *Journal of Statistical Planning and Inference*, Vol. 143: 1249-1272, 2013.

[18] Terejanu, G., Oliver, T., and Simmons, C., "Application of Predictive Model Selection to Coupled Models", Proceedings of the World Congress on Engineering and Computer Science, Vol. 2, Oct. 2011.

[19] Terejanu, G., Upadhyay, R.R., and Miki, K., "Bayesian experimental design for the active nitration of graphite by atomic nitrogen", *Experimental Thermal and Fluid Science*, Vol. 36, Jan. 2012, pp. 178-193.