An Interval Multi-level Monte Carlo Method for Reliability Analysis of Imprecise Probabilistic Systems

Douglas Brunson, Mohammad Motamed

Department of Mathematics and Statistics, University of New Mexico, United States

Abstract

In the reliability analysis of systems, we will in general encounter both aleatoric and epistemic uncertainty. To account for epistemic uncertainty, we need to consider an imprecise probabilistic framework in the calculation of quantities of interest. In this work, we present a method for accurate estimation of interval-valued probabilities by applying an implementation of the multi-level Monte Carlo (MC) algorithm with an iterative optimization strategy. We postulate this to be a more mathematically rigorous alternative to the use of probability boxes to estimate the unknown CDF for the system being analyzed. We present several numerical examples to demonstrate the superior efficiency of the multi-level MC approach over the more traditional MC apparoach for estimating failure probabilities.

Keywords: Reliability analysis, Uncertainty quantification, Imprecise probability, Failure probability, Multi-level Monte Carlo, Optimization

1. Introduction

Physical systems are often subject to uncertainty. The study of such systems requires uncertainty quantification (UQ), a process concerned with the characterization and propagation of uncertainty in the system. Most UQ methodologies are set in a probabilistic framework; see e.g. [22, 1, 8, 26, 6, 7, 13]. All these approaches assume that the uncertainty in the system is precisely known and can be described by precise probability density functions (PDFs). They may therefore be suitable for treating aleatoric (or random) uncertainty that arises from inherent randomness/variability in a system. Many physical systems may however exhibit a mixture of aleatoric and epistemic uncertainty, referred to as hybrid uncertainty, due to limited and/or in-

accurate information. For instance, the available data may be insufficient, or the mathematical model describing a system may be inaccurate. Such types of hybrid uncertainty may not be accurately represented by precise PDFs. Rather, one may need to consider an imprecise probabilistic framework; see e.g. [21, 23, 25, 24, 5]. One approach in this framework is to synthesize interval analysis and probability theory and build up interval-valued probability distributions [25]. This approach constructs a probability-box (or a *p-box*) consisting of a family of cumulative distribution functions (CDFs). The left and right envelopes of the family will form a *box* and bound the "unknown" distribution of the uncertain parameter from above and below.

In the present work we are concerned with the reliability analysis of systems that are subject to hybrid aleatoric-epistemic uncertainty. We particularly consider physical systems that are described by differential equations. Hence, the underlying mathematical models that we consider include sets of ordinary or partial differential equations with uncertain parameters represented by interval-valued probability distributions. An important quantity of interest (QoI) in reliability analysis is the *probability of failure*, measuring the likelihood that the system will fail at a given time or over a period of time. Examples include the probability that the maximum stress in a structure reaches a critical stress and the probability that the energy in a system exceeds a critical level. In an interval-valued probabilistic framework, as considered here, this quantity is an interval rather than a crisp real number. Our main goal is then the efficient computation of interval-valued failure probabilities in physical systems.

The computation of interval-valued failure probabilities requires interval arithmetic [12, 15], a numerical global optimization method; see e.g. [14, 20], and an algorithm for computing statistical moments; see e.g. [6, 4, 11, 10, 17, 8, 2, 27, 18]. Current computational methods are based on classical MC and quasi-MC teqniques coupled with a numerical global optimization method, referred to as interval MC and interval quasi-MC, respectively; see e.g. [28]. Although quasi-MC sampling may improve the slow convergence rate of classical MC sampling in some cases, for QoIs with low regularity, such as failure probabilities, the advantage of quasi-MC over classical MC sampling is not significant and cannot be guaranteed; see e.g. [3, 16]. Hence, a major problem with both methods is that they feature a slow convergence rate. In the context of differential equations, among the available MC sampling techniques [6, 4, 11, 10, 17], two algorithms are particularly advantagous in terms of improving the convergence of classical MC sampling in the absence

of stochastic regularity: 1) multi-level MC sampling [4], and 2) multi-order MC sampling [17]. Here, we consider and explore the former and present an interval multi-level MC sampling strategy for computing interval-valued failure probabilities. In particular, we combine multi-level MC sampling with an implementation of Nelder-Mead algorithm [19] and apply the method to the reliability analysis of imprecise probabilistic systems.

The rest of the paper is organized as follows. Section 2 provides the mathematical formulation of the problem, including the representation of interval-valued probability distributions and the probability of failure. In Section 3 we discuss and analyze an interval MC sampling approach for computing failure probability. We next present our interval multi-level MC sampling strategy in Section 4. Finally, in Section 5 we provide two numerical examples and demonstrate the performance of the proposed multi-level MC sampling technique.

2. Problem Formulation

In this section, we first provide the mathematical foundations of interval-valued probability. We next present the formulation of the problem in an interval-valued probabilistic framework and the ultimate goal of computations. We then address the numerical treatment of the problem. Throughout the present paper, we denote by $\mathbb{R}$ and $\mathbb{R}_+$ the set of all real numbers and the set of all strictly positive real numbers, respectively.

2.1. Interval-valued probability

An interval-valued random variable is a generalization of a random variable. A random variable $y : \Omega \to \mathbb{R}$ is a real-valued measurable function defined on a probability space (Ω, Σ, P), where Ω is a sample space, Σ is a non-empty *sigma*-field on Ω, and P is a probability measure assigned to each measurable subset of Ω satisfying Kolmogorov's axioms [9]. Every realization of a random variable $y(\omega)$, for some $\omega \in \Omega$, is a real number. If P is an absolutely continuous measure [9], it can be described by a single CDF denoted by F, or alternatively by a single PDF denoted by π through the following relations

$$F(y_0) = P(y \leq y_0) = \int_{-\infty}^{y_0} \pi(\tau)\, d\tau, \qquad y_0 \in \mathbb{R}.$$

The CDF and PDF are usually presented as functions of $y_0 \in \mathbb{R}$ and a set of n crisp parameters $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{R}^n$:

$$F(y_0; \mathbf{z}) = \int_{-\infty}^{y_0} \pi(\tau; \mathbf{z})\, d\tau, \qquad y_0 \in \mathbb{R}, \qquad \mathbf{z} \in \mathbb{R}^n. \tag{1}$$

This specifies a "parameterized CDF" for a random variable. For instance, for a normal random variable $y \sim \mathcal{N}(z_1, z_2^2)$ with two parameters being the mean $z_1 \in \mathbb{R}$ and standard deviation $z_2 \in \mathbb{R}_+$, the parameterized CDF reads:

$$F(y_0; \mathbf{z}) = \frac{1}{\sqrt{2\pi} z_2} \int_{-\infty}^{y_0} e^{\frac{-(\tau - z_1)^2}{2 z_2^2}}\, d\tau, \qquad y_0 \in \mathbb{R}, \qquad \mathbf{z} = (z_1, z_2) \in \mathbb{R} \times \mathbb{R}_+.$$

This concept can be generalized to define interval-valued random variables.

Definition 1. *Let $\mathcal{I}(\mathbb{R})$ be the set of all closed intervals on the real line. An interval-valued random variable $\tilde{y} : \Omega \to \mathcal{I}(\mathbb{R})$ is an interval-valued measurable function on the sample space Ω. Every realization of an interval-valued variable $\tilde{y}(\omega)$, for some $\omega \in \Omega$, is a closed interval rather than a real number.*

Throughout the present paper, we denote an interval-valued random variable by the superimposition of a tilde over a letter. Hence by y and $\tilde{y}$ we mean a random variable and an interval-valued random variable, respectively.

One type of interval-valued random variables that we consider is a random variable with interval-valued parameters. For example, a normal random variable with mean $Z_1 \in \mathcal{I}(\mathbb{R})$ and standard deviation $Z_2 \in \mathcal{I}(\mathbb{R}_+)$ being two closed intervals is a normal interval-valued random variable, denoted by $\tilde{y} \sim \mathcal{N}(Z_1, Z_2^2)$. It can be written in terms of a standard normal random variable $y \sim \mathcal{N}(0, 1)$ and the intervals Z_1 and Z_2 as $\tilde{y} = Z_1 + y\, Z_2$.

For this type of interval-valued random variables the *interval-valued CDF*, denoted by $\tilde{F}$, can be defined by generalizing the parameterized representation (1) of the CDF for random variables as follows.

Definition 2. *Let $Z_i \in \mathcal{I}(\mathbb{R})$, $i = 1, \ldots, n$, be n closed, bounded intervals. Consider an interval-valued random variable $\tilde{y}$, consisting of a random variable with n interval-valued parameters $Z_1, \ldots, Z_n$. For any fixed point $\mathbf{z} = (z_1, \ldots, z_n)$ in $\mathbf{Z} = (Z_1, \ldots, Z_n)$, the CDF of $\tilde{y}$ is a parameterized CDF $F(y_0; \mathbf{z})$ given as a function of $y_0 \in \mathbb{R}$ and $\mathbf{z}$ by (1). Let F_L and F_R be the extrema of the family of all parameterized CDFs over $\mathbf{Z}$:*

$$F_L(y_0) = \max_{\mathbf{z} \in \mathbf{Z}} F(y_0; \mathbf{z}), \qquad F_R(y_0) = \min_{\mathbf{z} \in \mathbf{Z}} F(y_0; \mathbf{z}). \tag{2}$$

Then the interval-valued CDF of $\tilde{y}$, evaluated at $y_0 \in \mathbb{R}$, is the interval:

$$\tilde{F}(y_0) = [F_L(y_0), F_R(y_0)]. \tag{3}$$

The sets $\{F_L(y_0),\, y_0 \in \mathbb{R}\}$ and $\{F_R(y_0),\, y_0 \in \mathbb{R}\}$ form two left (upper) and right (lower) envelopes encapsulating the family of all parameterized CDFs from above and below and form a p-box.

We next define interval-valued random functions by generalizing the notion of random functions.

Definition 3. *Let $\mathbf{y} = (y_1, \ldots, y_m) \in \Gamma \subset \mathbb{R}^m$ be a vector of m random variables with a bounded joint PDF $\pi = \pi(\mathbf{y}) : \Gamma \to \mathbb{R}_+$. Let $\mathbf{Z} = (Z_1, \ldots, Z_n)$ be a vector of n closed, bounded intervals $Z_i \in \mathcal{I}(\mathbb{R})$, $i = 1, \ldots, n$. An interval-valued random function is a crisp map with arguments $\mathbf{y}$ and $\mathbf{Z}$, generating an output interval-valued random variable, written as $\tilde{u} = u(\mathbf{y}, \mathbf{Z})$.*

We notice that an interval-valued random function can also be interpreted pointwise in $\mathbf{y} \in \Gamma$: at every realization, say $\mathbf{y}_0 \in \Gamma$, if the mapping u is continuous, the output $u(\mathbf{y}_0, \mathbf{Z})$ is an interval given by

$$u(\mathbf{y}_0, \mathbf{Z}) = [\min_{\mathbf{z}\in\mathbf{Z}} u(\mathbf{y}_0, \mathbf{z}),\, \max_{\mathbf{z}\in\mathbf{Z}} u(\mathbf{y}_0, \mathbf{z})], \qquad \forall \mathbf{y}_0 \in \Gamma.$$

If the mapping u is not continuous but is defined for all points $\mathbf{z} \in \mathbf{Z}$, we have:

$$\min_{\mathbf{z}\in\mathbf{Z}} u(\mathbf{y}_0, \mathbf{z}) \le u(\mathbf{y}_0, \mathbf{Z}) \le \max_{\mathbf{z}\in\mathbf{Z}} u(\mathbf{y}_0, \mathbf{z}), \qquad \forall \mathbf{y}_0 \in \Gamma.$$

Similarly, we can define interval-valued spatio-temporal random processes by generalizing the notion of spatio-temporal random processes.

Definition 4. *Let $D \subset \mathbb{R}^d$ be a compact spatial domain, with $d = 1, 2, 3$. Let further $[0, T]$ be a time interval between zero and a final time $T > 0$. Consider a spatial variable $\mathbf{x} = (x_1, \ldots, x_d) \in D$, a temporal variable $t \in [0, T]$, and let $\mathbf{y}$ and $\mathbf{Z}$ be as in Definition 3. An interval-valued spatio-temporal random processes, written as $\tilde{u}(\mathbf{x}, t) = u(\mathbf{x}, t, \mathbf{y}, \mathbf{Z})$, is an infinite collection of interval-valued random variables corresponding to all $t \in [0, T]$ and all $\mathbf{x} \in D$:*

$$\tilde{u} = \{u(\mathbf{x}, t, \mathbf{y}, \mathbf{Z}),\, \mathbf{x} \in D,\, t \in [0, T]\}.$$

Two particular types of interval-valued spatio-temporal random processes include interval-valued temporal random processes (in the absence of $\mathbf{x}$) and interval-valued spatial random processes (in the absence of t).

2.2. Problem statement

We are interested in computing the interval-valued failure probability of physical systems described by a set of ODEs or PDEs within an interval-valued probabilistic framework. In what follows, we consider the more general case of time-dependent PDEs with uncertain parameters described by interval-valued spatio-temporal random processes. Time-independent PDEs and ODEs will then be particular cases of this general case.

Let $u(\mathbf{x}, t, \mathbf{y}, \mathbf{Z})$ be the solution to a time-dependent PDE problem with uncertain parameters described by interval-valued spatio-temporal random processes. For example, we may consider the acoustic wave equation

$$u_{tt}(\mathbf{x}, t, \mathbf{y}, \mathbf{Z}) - \nabla \cdot (a^2(\mathbf{x}, \mathbf{y}, \mathbf{Z}) \nabla u(\mathbf{x}, t, \mathbf{y}, \mathbf{Z})) = f(\mathbf{x}, t, \mathbf{y}, \mathbf{Z}), \tag{4}$$

augmented with some initial-boundary conditions. Here, $\mathbf{x} = (x_1, \ldots, x_d) \in D \subset \mathbb{R}^d$ and $t \in [0, T]$ denote the d-dimensional spatial variables and time, respectively, $\mathbf{y} = (y_1, \ldots, y_m) \in \Gamma \subset \mathbb{R}^m$ is a vector of m random variables with a bounded joint PDF π, and $\mathbf{Z} = (Z_1, \ldots, Z_n)$ is a vector of n closed, bounded intervals $Z_i \in \mathcal{I}(\mathbb{R})$, $i = 1, \ldots, n$. In this example the PDE solution u represents acoustic pressure, and the hybrid uncertainty may be in the sound speed a and/or in the sound source f.

Let $g(\mathbf{y}, \mathbf{Z})$ be an interval-valued random function related to u, referred to as the *limit state function* [28]. The state function may be obtained by applying an algebraic, differential, or integral operator on u. Suppose that failure occurs when $g(\mathbf{y}, \mathbf{Z}) \leq 0$. For example, a mode of damage in a system may initiate when the PDE solution at a fixed point $(\mathbf{x}_0, t_0)$ in space and time reaches a critical value u_{cr}, that is, $g(\mathbf{y}, \mathbf{Z}) = u_{cr} - u(\mathbf{x}_0, t_0, \mathbf{y}, \mathbf{Z})$.

The main goal is to compute the probability of failure, denoted by P_f and given by

$$P_f(\mathbf{Z}) := \int_{[g(\mathbf{y}, \mathbf{Z}) \leq 0]} \pi(\mathbf{y}) d\mathbf{y} = \mathbb{E}[\mathbb{I}_{[g(\mathbf{y}, \mathbf{Z}) \leq 0]}]. \tag{5}$$

Here, $\mathbb{I}_{[.]}$ is the indicator function, taking the value 1 or 0 if the event [.] is "true" or "false", respectively. Introducing the QoI

$$Q(\mathbf{y}, \mathbf{Z}) = \mathbb{I}_{[g(\mathbf{y}, \mathbf{Z}) \leq 0]}, \tag{6}$$

we can express the failure probability (5) as the expected value of Q in (6):

$$P_f(\mathbf{Z}) = \mathbb{E}[Q(\mathbf{y}, \mathbf{Z})]. \tag{7}$$

We note that the failure probability, given by either (5) or (6)-(7), takes a range of values rather than being a crisp number. However, since Q in (6) may be discontinuous, the failure probability may not be a closed interval. Hence, we are interested in computing its lower and upper bounds, denoted by $\underline{P}_f$ and $\overline{P}_f$, respectively:

$$\underline{P}_f := \min_{\mathbf{z}\in\mathbf{Z}} \mathbb{E}[Q(\mathbf{y},\mathbf{z})] \leq P_f(\mathbf{Z}) \leq \max_{\mathbf{z}\in\mathbf{Z}} \mathbb{E}[Q(\mathbf{y},\mathbf{z})] =: \overline{P}_f. \tag{8}$$

2.3. Numerical approach

A general method to numerically compute the minimum and maximum failure probabilities in (8) consists of two parts: 1) a numerical global optimization procedure for computing global extrema, and 2) a sample-based numerical algorithm for computing the expectation of a QoI given by solutions to systems of differential equations. More precisely, we need to take the following two steps:

1. We first need to utilize a numerical optimization procedure which can find the global extrema of $P_f(\mathbf{z})$ for $\mathbf{z} \in \mathbf{Z}$. This can be done, for instance, by an iterative method; see e.g. [14, 20]. The choice of the method would depend on the dimension of $\mathbf{Z}$. In general, an iterative optimization algorithm requires M_o function evaluations $P_f(\mathbf{z}^{(k)})$ at some fixed points $\{\mathbf{z}^{(k)}\}_{k=1}^{M_o} \in \mathbf{Z}$, to be determined by the algorithm.

2. Each function evaluation $P_f(\mathbf{z}^{(k)})$ amounts to computing the expectation $\mathbb{E}[Q(\mathbf{y},\mathbf{z}^{(k)})]$, which may be done by a MC sampling strategy; see e.g. [6, 4, 11, 10, 17]. Using M_s samples, the sampling algorithm needs M_s realizations of $Q(\mathbf{y}^{(m)},\mathbf{z}^{(k)})$, with $m = 1,\ldots,M_s$, for each fixed k. The approximation of each realization $Q(\mathbf{y}^{(m)},\mathbf{z}^{(k)})$ can be done by a deterministic discretization method to compute the numerical solution to the underlying set of differential equations and thereby to obtain an approximation $Q_h(\mathbf{y}^{(m)},\mathbf{z}^{(k)})$, where h denotes the time step or the mesh/grid size in the discretization.

We notice that one may think that in step 2 a spectral stochastic method (see e.g. [8, 2, 27, 18]) would be a better alternative to a MC sampling approach. However, since Q is the indicator function and hence irregular with respect to $\mathbf{y}$, spectral stochastic methods that rely on the high regularity of QoIs will not work.

Now let $\underline{\mathcal{P}}_f$ and $\overline{\mathcal{P}}_f$ be the approximations to $\underline{P}_f = \min_{\mathbf{z}\in\mathbf{Z}} P_f(\mathbf{z})$ and $\overline{P}_f = \max_{\mathbf{z}\in\mathbf{Z}} P_f(\mathbf{z})$, respectively, obtained by the above two steps. In these approximations two major errors are incurred. The first error is due to the sample-based algorithm for estimating the expectation

$$P_f(\mathbf{z}) = \mathbb{E}[Q(\mathbf{y},\mathbf{z})] \approx \mathcal{P}_f(\mathbf{z}), \qquad \forall\, \mathbf{z} \in \mathbf{Z},$$

at any given point $\mathbf{z} \in \mathbf{Z}$. The second error is due to imperfect minimization or maximization

$$\min_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) \approx \underline{\mathcal{P}}_f, \qquad \max_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) \approx \overline{\mathcal{P}}_f.$$

Correspondingly, we may split the total error in the approximation into two parts, as follows. We write

$$\begin{aligned}\underline{\varepsilon} := |\underline{P}_f - \underline{\mathcal{P}}_f| &\le |\min_{\mathbf{z}\in\mathbf{Z}} P_f(\mathbf{z}) - \min_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z})| + |\min_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) - \underline{\mathcal{P}}_f| \\ &\le \max_{\mathbf{z}\in\mathbf{Z}} |P_f(\mathbf{z}) - \mathcal{P}_f(\mathbf{z})| + |\min_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) - \underline{\mathcal{P}}_f|,\end{aligned}$$

$$\begin{aligned}\overline{\varepsilon} := |\overline{P}_f - \overline{\mathcal{P}}_f| &\le |\max_{\mathbf{z}\in\mathbf{Z}} P_f(\mathbf{z}) - \max_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z})| + |\max_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) - \overline{\mathcal{P}}_f| \\ &\le \max_{\mathbf{z}\in\mathbf{Z}} |P_f(\mathbf{z}) - \mathcal{P}_f(\mathbf{z})| + |\max_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) - \overline{\mathcal{P}}_f|,\end{aligned}$$

where $\underline{\varepsilon}$ and $\overline{\varepsilon}$ are the error in the approximate lower and upper bounds, respectively. We now set

$$\varepsilon_s := \max_{\mathbf{z}\in\mathbf{Z}} |P_f(\mathbf{z}) - \mathcal{P}_f(\mathbf{z})|, \quad \varepsilon_o := \max(|\min_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) - \underline{\mathcal{P}}_f|, |\max_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z}) - \overline{\mathcal{P}}_f|), \tag{9}$$

where ε_s is the error in the sample-based algorithm, and ε_o is the error in the numerical optimization procesure. We then have

$$\varepsilon_{\text{TOTAL}} := \max(\underline{\varepsilon}, \overline{\varepsilon}) \le \varepsilon_s + \varepsilon_o. \tag{10}$$

In practical applications, we usually need to numerically compute the minimum and maximum failure probabilities in (8) to within a desired tolerance ε_{TOL}, that is, we need to have $\varepsilon_{\text{TOTAL}} \le \varepsilon_{\text{TOL}}$. Moreover, it is often desired to achieve this tolerance with minimum computational cost. This suggests that we may select the numerical optimization method and the numerical parameters in the sampling method, including M_s and h, by minimizing the computational cost subject to the error constraint $\varepsilon_s + \varepsilon_o = \varepsilon_{\text{TOL}}$.

3. Interval Monte Carlo sampling

In this section, we discuss and analyze an interval MC sampling approach for computing interval-valued failure probabilities.

3.1. General strategy

The goal is to compute both $\underline{P}_f$ and $\overline{P}_f$ to within a given tolerance ε_{TOL}, that is, we want

$$\underline{\varepsilon} = |\underline{P}_f - \underline{\mathcal{P}}_f| \leq \varepsilon_{\text{TOL}} \quad \text{and} \quad \overline{\varepsilon} = |\overline{P}_f - \overline{\mathcal{P}}_f| \leq \varepsilon_{\text{TOL}}.$$

For this purpose, we use (10) and require

$$\varepsilon_o + \varepsilon_s \leq \alpha\, \varepsilon_{\text{TOL}} + (1-\alpha)\, \varepsilon_{\text{TOL}} = \varepsilon_{\text{TOL}}, \qquad \alpha \in (0,1), \tag{11}$$

where α is a splitting parameter. As an example we may set $\alpha = 0.5$.

We consider an implementation of the Nelder-Mead direct simplex search algorithm [19] to numerically find both the minimum and maximum of $\mathcal{P}_f(\mathbf{z})$. The algorithm takes initial guesses for $\arg\min_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z})$ and $\arg\max_{\mathbf{z}\in\mathbf{Z}} \mathcal{P}_f(\mathbf{z})$ and iteratively finds a sequence of test points in $\mathbf{Z} \subset \mathbb{R}^n$ arranged as the vertices of a set of $(n+1)$-dimensional simplexes, until the following two conditions are satisfied:

- The maximum Euclidean distance between any pair of the last $n+1$ test points in the final simplex is within $\alpha\, \varepsilon_{TOL}$.
- The maximum distance between $\mathcal{P}_f(\mathbf{z})$ evaluated at any pair of the last $n+1$ test points in the final simplex is within $\alpha\, \varepsilon_{TOL}$.

We use the above two stopping criteria as indications of the convergence of the optimiaztion algorithm satisfying the requirement $\varepsilon_o \leq \alpha\, \varepsilon_{\text{TOL}}$ in (11). Assuming this can be achieved with M_o iterations, we need M_o evaluations of $\mathcal{P}_f(\mathbf{z})$ at the points $\mathbf{z}^{(k)}$, $k = 1, \ldots, M_o$. At any such fixed point, the Monte Carlo estimator reads

$$\mathcal{P}_f(\mathbf{z}^{(k)}) = \frac{1}{M_s} \sum_{m=1}^{M_s} \mathbb{I}_{[g_h(\mathbf{y}^{(m)}, \mathbf{z}^{(k)}) \leq 0]}, \qquad k = 1, \ldots, M_o. \tag{12}$$

Here, $g_h(\mathbf{y}^{(m)}, \mathbf{z}^{(k)})$ is the approximate value of the *limit state function* at the point $(\mathbf{y}^{(m)}, \mathbf{z}^{(k)})$ computed by a discretization method with a time step or a mesh/grid size h. The parameter h and the number of MC samples M_s are selected so that $|P_f(\mathbf{z}^{(k)}) - \mathcal{P}_f(\mathbf{z}^{(k)})| \leq (1-\alpha)\, \varepsilon_{TOL}$ for all $k = 1, \ldots, M_o$. This will guarantee that $\varepsilon_s \leq (1-\alpha)\, \varepsilon_{\text{TOL}}$ in (11) is achieved.

Remark 1. *Another MC formulation for minimum and maximum failure probability estimates is suggested in [28], given by*

$$\underline{\mathsf{P}}_f = \frac{1}{M_s}\sum_{m=1}^{M_s} \mathbb{I}_{[\max_{\mathbf{z}\in\mathbf{Z}} g_h(\mathbf{y}^{(m)},\mathbf{z})\le 0]}, \qquad \overline{\mathsf{P}}_f = \frac{1}{M_s}\sum_{m=1}^{M_s} \mathbb{I}_{[\min_{\mathbf{z}\in\mathbf{Z}} g_h(\mathbf{y}^{(m)},\mathbf{z})\le 0]}. \quad (13)$$

Obviously, compared to our formulation, these estimates are conservative. In addition, the estimators $\underline{\mathsf{P}}_f$ and $\overline{\mathsf{P}}_f$ in (13) do not necessarily converge to $\underline{P}_f$ and $\overline{P}_f$, respectively, as $M_s \to \infty$ and $h \to 0$. This latter issue is demonstrated for the two numerical examples in Section 5.

In the remainder of this section, we discuss a general strategy for the optimal selection of the numerical parameters, h and M_s, and apply the strategy to ODEs and time-dependent PDEs.

3.2. Cost-error analysis

We propose to optimally select the numerical parameters h and M_s in the MC estimator (12) by minimizing the computational cost subject to the error constraint $\varepsilon_s \le (1-\alpha)\,\varepsilon_{TOL}$. We use the method of Lagrange multipliers as follows.

We first note that $\varepsilon_s \ge |P_f(\mathbf{z}) - \mathcal{P}_f(\mathbf{z})|$, $\forall \mathbf{z} \in \mathbf{Z}$; see (9). We then split the error due to the sample-based algorithm into two parts:

$$|P_f(\mathbf{z})-\mathcal{P}_f(\mathbf{z})| \le \underbrace{|\mathbb{E}[Q(\mathbf{y},\mathbf{z})] - \mathbb{E}[Q_h(\mathbf{y},\mathbf{z})]|}_{\varepsilon_I} + \underbrace{|\mathbb{E}[Q_h(\mathbf{y},\mathbf{z})] - \mathcal{P}_f(\mathbf{z})|}_{\varepsilon_{II}} \le (1-\alpha)\varepsilon_{TOL}. \quad (14)$$

The first term ε_I in (14) is the discretization error of the deterministic solver that satisfies

$$\varepsilon_I \le c_1 h^q, \qquad \forall \mathbf{y} \in \Gamma, \quad (15)$$

where c_1 is a positive constant, and q is the order of the deterministic discretization solver in computing $\mathbb{E}[Q_h(\mathbf{y},\mathbf{z})]$. Note that we cannot actually calculate $\mathbb{E}[Q_h(\mathbf{y},\mathbf{z})]$ and therefore must choose a reasonably large number of random samples and use the same set of random samples for a set of different step sizes in order to estimate c_1 and q. Note also that for different parameters $\mathbf{z}$, there will be different values of c_1 and q. As the optimal step size h and sample size M_s will each depend on c_1 and q, we want to settle on one value of q and one value of c_1 which will work for any input $\mathbf{z} \in \mathbf{Z}$. This is done by estimating values of c_1 and q using least squares for a set

of step sizes for each of a set of values of $\mathbf{z}$. The maximum least squares estimator $\hat{c_1}(\mathbf{z})$ is taken to be c_1, and the minimum least squares estimator $\hat{q}(\mathbf{z})$ is taken to be q. This is to be conservative because, as shown later in this section, the optimal step size decreases for larger c_1 and smaller q. The optimal sample size M_s also increases for smaller q.

The second error term, ε_{II} is the statistical error. First, we know that the variance, $\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]$ of our QoI is bounded (since the expected value is bounded; see proof in Section 4). Then the Central Limit Theorem gives

$$\varepsilon_{II} = |\mathbb{E}[Q_h(\mathbf{y}, \mathbf{z})] - \frac{1}{M_s}\sum_{i=1}^{M_s} \mathbb{I}_{[g_h(\mathbf{y}_i, \mathbf{z}) \leq 0]}| \leq c_2 \frac{\sqrt{\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]}}{\sqrt{M_s}} \tag{16}$$

with the notation $\leq$ understood to mean that

$$P\big(\varepsilon_{II} \leq c_2 \frac{\sqrt{\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]}}{\sqrt{M_s}}\big) = 2\phi(c_2) - 1, \qquad \text{as} \quad M_s \to \infty \tag{17}$$

where P denotes a probability measure, and $\phi(c_2) = \int_{-\infty}^{c_2} \frac{1}{\sqrt{2\pi}} e^{-\tau^2/2} d\tau$ is the CDF for the standard normal random variable c_2. The computational cost, or work, of computing the Monte Carlo estimator for each input vector $\mathbf{z}$ is denoted by W_{MC}. The expression of computational cost will be different depending on whether we are working with an ODE or a PDE, and it is appropriate to break into two subsections at this point.

3.3. ODEs

For ODEs, because we are dealing with one independent variable, the work is given by $W_{MC} \propto M_s q h^{-1}$. To choose optimal h and M_s, we solve the following minimization problem, with $c_3 = c_2\sqrt{\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]}$.

$$\min_{h, M_s} M_s q h^{-1} \qquad \text{s.t.} \qquad c_1 h^q + c_3 M_s^{-1/2} = (1-\alpha)\varepsilon_{TOL} \tag{18}$$

The Lagrangian is given by

$$L(h, M_s, \lambda) = M_s q h^{-1} + \lambda(c_1 h^q + c_3 M_s^{-1/2} - (1-\alpha)\varepsilon_{TOL}) \tag{19}$$

We enforce $\partial_h L = \partial_{M_s} L = \partial_\lambda L = 0$. After some algebraic manipulations, the optimal step size and optimal sample size are found to be

$$\begin{aligned} h &= \big(\frac{(1-\alpha)\varepsilon_{TOL}}{c_1(1+2q)}\big)^{1/q} \\ M_s &= {c_2}^2 \mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]\big(\frac{1+2q}{2q}\big)((1-\alpha)\varepsilon_{TOL})^{-2} \end{aligned} \tag{20}$$

Finally, with the expressions given in (20), the optimal work is

$$W_{OPT} \propto \mathcal{O}(((1-\alpha)\varepsilon_{TOL})^{-(2+1/q)}) \tag{21}$$

3.4. Time-Dependent PDEs

For time-dependent PDEs, we consider the Method of Lines in which the PDE is discretized in space with the time variable considered continuous. This changes the PDE into an initial value ODE problem. Finite differencing is used for both spatial and temporal discretization. Assuming that the temporal step size may be computed from the grid size (e.g. the CFL condition for hyperbolic PDEs), the computational cost is estimated by $W_{MC} \propto M_s h^{-(1+d)}$, for d spatial dimensions where h is the characteristic grid size for the spatial discretization. For ODEs, $d = 0$. The optimization problem is given by

$$\min_{h,M_s} M_s h^{-(1+d)} \qquad \text{s.t.} \qquad c_1 h^q + c_3 M_s^{-1/2} = (1-\alpha)\varepsilon_{TOL} \tag{22}$$

The Lagrangian is given by

$$L(h, M_s, \lambda) = M_s h^{-(1+d)} + \lambda(c_1 h^q + c_3 M_s^{-1/2} - (1-\alpha)\varepsilon_{TOL}) \tag{23}$$

Enforcing $\partial_h L = \partial_{M_s} L = \partial_\lambda L = 0$ and performing the necessary algebraic manipulations, the optimal grid size h and sample size M_s are given by

$$\begin{aligned} h &= \Big(\frac{\varepsilon_s}{c_1(1+2q/(1+d))}\Big)^{1/q} \\ M_s &= {c_2}^2 \mathbb{V}[Q_h(\mathbf{y},\mathbf{z})]\Big(\frac{1+d+2q}{2q}\Big)((1-\alpha)\varepsilon_{TOL})^{-2} \end{aligned} \tag{24}$$

Using (24), the optimal work is

$$W_{OPT} \propto \mathcal{O}(((1-\alpha)\varepsilon_{TOL})^{-(2+1/q+d/q)}) \tag{25}$$

As expected, the cost for the Interval Monte Carlo sampling where the QoI depends on the solution to a PDE will be increase exponentially as the dimension of the problem increases.

4. Interval Multi-Level Monte Carlo Sampling

In this section, we discuss and analyze an interval multi-level Monte Carlo sampling approach for computing failure probability. The optimization approach is the same as in Section 3, We consider the problem of computing $\mathcal{P}_f(\mathbf{z})$ such that $|P_f(\mathbf{z}) - \mathcal{P}_f(\mathbf{z})| \leq (1-\alpha)\varepsilon_{TOL}$, but this time with the estimator $\mathcal{P}_f$ defined as

$$\mathcal{P}_f(\mathbf{z}) = \mathcal{A}_{MLMC}(\mathbf{z}), \tag{26}$$

where $\mathcal{A}_{MLMC}$ is the multi-level Monte Carlo (MLMC) estimator of $\mathbb{E}[Q(\mathbf{y}, \mathbf{z})]$. In this section, the goal is to achieve the same error condition (14).

4.1. General method for multi-level Monte Carlo sampling

To build up the MLMC estimator, we start with a decreasing sequence of mesh sizes $h_0 > h_1 > \cdots > h_L$ defined by

$$h_l = h_0\beta^{-l}, \qquad l = 0, 1, \ldots, L \qquad \beta \geq 2 \tag{27}$$

With a telescoping sum,

$$\mathbb{E}[Q_L(\mathbf{y}, \mathbf{z})] = \mathbb{E}[Q_0(\mathbf{y}, \mathbf{z})] + \sum_{i=1}^{L} \mathbb{E}[Q_l(\mathbf{y}_i, \mathbf{z}) - Q_{l-1}(\mathbf{y}_i, \mathbf{z})],$$

where $\mathbb{E}[Q_L(\mathbf{y}, \mathbf{z})] = \mathbb{E}[Q_h(\mathbf{y}, \mathbf{z})]$ with $h = h_0\beta^{-L}$. The MLMC estimator is an approximation of $\mathbb{E}[Q_L(\mathbf{y}, \mathbf{z})]$ computed by taking sample averages.

$$\mathcal{A}_{MLMC}(\mathbf{z}) = \frac{1}{M_0}\sum_{m_0=1}^{M_0} Q_0(\mathbf{y}^{m_0}, \mathbf{z}) + \sum_{l=1}^{L}\frac{1}{M_l}\sum_{m_l=1}^{M_l}[Q_l(\mathbf{y}^{m_l}, \mathbf{z}) - Q_{l-1}(\mathbf{y}^{m_l}, \mathbf{z})] \tag{28}$$

The term ε_I in (14) is the discretization error, and follows the relationship

$$\varepsilon_I \leq c_1 {h_L}^q \qquad \forall \mathbf{y} \in \Gamma \tag{29}$$

where $c_1 > 0$ and q is the order of accuracy of the DE solver in computing $\mathbb{E}[Q_L(\mathbf{y}, \mathbf{z}) - Q_{L+1}(\mathbf{y}, \mathbf{z})]$. For this implementation of MLMC, it is only necessary to have a good estimate of q. This estimate of q is obtained by estimating $\mathbb{E}[Q_L(\mathbf{y}, \mathbf{z}) - Q_{L+1}(\mathbf{y}, \mathbf{z})]$ with a common, large sample set and small set of step sizes for a set of vlues of $\mathbf{z}$ and performing least squares on the data for each input $\mathbf{z}$. Again, to be conservative, the minimum of

the estimators $\hat{q}(\mathbf{z})$ is taken to be q. From the Central Limit Theorem, the statistical error, ε_{II} can be written as

$$\begin{aligned}\varepsilon_{II} &= |\mathbb{E}[Q_L(\mathbf{y},\mathbf{z})] - \mathcal{A}_{MLMC}(\mathbf{z})| \\ &\leq c_2\sqrt{\frac{\mathbb{V}[Q_0(\mathbf{y},\mathbf{z})]}{M_0} + \sum_{l=1}^{L}\frac{\mathbb{V}[Q_l(\mathbf{y},\mathbf{z}) - Q_{l-1}(\mathbf{y},\mathbf{z})]}{M_l}} =: c_2\sqrt{\sum_{l=0}^{L}\frac{V_l}{M_l}} \end{aligned} \tag{30}$$

with the shorthand notation defined by $V_0 = \mathbb{V}[Q_0(\mathbf{y},\mathbf{z})]$ and $V_l = \mathbb{V}[Q_l(\mathbf{y},\mathbf{z})]$ for $l = 1, 2, \ldots, L$. As with Monte Carlo, the relation in (30) is interpreted by

$$P\big(\varepsilon_{II} \leq c_2\sqrt{\sum_{l=0}^{L}\frac{V_l}{M_l}}\big) = 2\phi(c_2) - 1, \qquad \text{as} \quad M_l \to \infty \tag{31}$$

where P denotes a probability measure, and $\phi(c_2) = \int_{-\infty}^{c_2}\frac{1}{\sqrt{2\pi}}e^{-\tau^2/2}d\tau$ is the CDF for the standard normal random variable c_2. In what follows, we also need to define the *strong error* given by

$$|\mathbb{V}[Q(\mathbf{y},\mathbf{z}) - Q_l(\mathbf{y},\mathbf{z})]| \leq |\mathbb{E}[(Q(\mathbf{y},\mathbf{z}) - Q_l(\mathbf{y},\mathbf{z}))^2]| \leq {c_1}' {h_l}^{q'} \qquad \forall \mathbf{y} \in \Gamma \tag{32}$$

where $c_2 > 0$ and q' is related to the order of accuracy of the DE solver. For the numerical examples in Section 5, the value of q' is estimated using least squares in the same way as was done in estimating q - taking the minimum least squares estimate of a power-law fit to the quantity $|\mathbb{E}[(Q(\mathbf{y},\mathbf{z}) - Q_l(\mathbf{y},\mathbf{z}))^2]|$ for a large common set of random samples, a set of grid sizes, and a set of $\mathbf{z}$-values. As a remark, it is typical for $q' \approx 2q$ [[CITE]]. However, this is not the case for estimating failure probabilities. This is important to point out because for the numerical examples in Section 5, it is seen that $q' \approx q$, and this has theoretical justification. For the QoI defined in (6), $q' \approx q$.

Proof. Begin by writing the QoI as
$\mu(\mathbf{z}) := \mathbb{E}[Q(\mathbf{y},\mathbf{z})] = \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} Q(\mathbf{y}_i,\mathbf{z}) = \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}_{[g(\mathbf{y}_i,\mathbf{z})\leq 0]}$.
Then, the variance, $\mathbb{V}[Q(\mathbf{y},\mathbf{z})] = \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}[\mathbb{I}_{[g(\mathbf{y}_i,\mathbf{z})\leq 0]} - \mu(\mathbf{z})]^2$
$= \lim_{N\to\infty}\frac{1}{N}[\sum_{i=1}^{N}\mathbb{I}^2_{[g(\mathbf{y}_i,\mathbf{z})\leq 0]} - 2\mu(\mathbf{z})\sum_{i=1}^{N}\mathbb{I}_{[g(\mathbf{y}_i,\mathbf{z})\leq 0]} + \sum_{i=1}^{N}\mu(\mathbf{z})^2]$
$= \lim_{N\to\infty}\frac{1}{N}[N\mu(\mathbf{z}) - 2\mu(\mathbf{z})\sum_{i=1}^{N}\mathbb{I}_{[g(\mathbf{y}_i,\mathbf{z})\leq 0]} + N\mu(\mathbf{z})^2]$
$= \lim_{N\to\infty}\frac{1}{N}[N\mu(\mathbf{z}) - N\mu(\mathbf{z})^2]$
$= \mu(\mathbf{z})(1-\mu(\mathbf{z}))$
Thus, if $\mu(\mathbf{z}) = \mathbb{E}[Q_l(\mathbf{y},\mathbf{z})] + \mathcal{O}({h_l}^q)$, then $\mu(\mathbf{z})(1-\mu(\mathbf{z})) = \mathbb{E}[Q_l(\mathbf{y},\mathbf{z})] +$

$\mathcal{O}(h_l{}^q)$ as well. Noting that $\mathbb{V}[Q(\mathbf{y},\mathbf{z})] = \mathbb{V}[Q_l(\mathbf{y},\mathbf{z})] + \mathcal{O}(h_l{}^{q'})$, it is clear that $q' \approx q$. □

In order to optimally select the number of samples M_l at each level and the final level L required to achieve ε_s, we introduce a splitting parameter, $\theta \in (0,1)$ and split the tolerance as follows:

$$\varepsilon_{MLMC} \leq \varepsilon_I + \varepsilon_{II} \leq (1-\theta)(1-\alpha)\varepsilon_{TOL} + \theta(1-\alpha)\varepsilon_{TOL} = (1-\alpha)\varepsilon_{TOL} \quad (33)$$

That is, we require $\varepsilon_I \leq (1-\theta)(1-\alpha)\varepsilon_{TOL}$ and $\varepsilon_{II} \leq \theta(1-\alpha)\varepsilon_{TOL}$. Note that the cost of computing $Q_l(\mathbf{y}^{m_l},\mathbf{z}) - Q_{l-1}(\mathbf{y}^{m_l},\mathbf{z})$ is dominated by the cost of computing $Q_l(\mathbf{y}^{m_l},\mathbf{z})$. Hence, the cost of computing the MLMC estimator (28) is given by

$$W_{MLMC} \propto \sum_{l=0}^{L} M_l W_l \quad (34)$$

Recall from Section 3 that for ODEs, $W_l \propto q{h_l}^{-1}$, and for time-dependent PDEs solved numerically via the Method of Lines, $W_l \propto {h_l}^{-(1+d)}$.

The MLMC algorithm implemented in this paper is iterative and consists of two main steps.

1. Optimal Sample Size for Each Level

We minimize W_{MLMC} subject to the accuracy constraint $\varepsilon_{II} \leq \theta\varepsilon_s$. The Lagrangian is given by

$$L(M_l,\lambda) = \sum_{l=0}^{L} M_l W_l + \lambda\Big(\sum_{l=0}^{L} \frac{V_l}{M_l} - (\frac{\theta(1-\alpha)\varepsilon_{TOL}}{c_2})^2\Big) \quad (35)$$

By requiring $\partial_{M_l} L = 0$ and $\partial_\lambda L = 0$, the optimal choice for M_l is given by

$$M_l = \Big\lceil (\frac{\theta(1-\alpha)\varepsilon_{TOL}}{c_2})^2 \sqrt{\frac{V_l}{W_l}} \sum_{l=0}^{L} \sqrt{V_l W_l} \Big\rceil \quad (36)$$

2. Stopping Criterion

We begin with $L = 2$ and iteratively add levels until $\varepsilon_I \leq (1-\theta)(1-\alpha)\varepsilon_{TOL}$ is achieved. For a practical stopping criterion, write

$$\begin{aligned} \mathbb{E}[Q(\mathbf{y},\mathbf{z}) - Q_L(\mathbf{y},\mathbf{z})] &= \sum_{l=L+1}^{\infty} \mathbb{E}[Q_l(\mathbf{y},\mathbf{z}) - Q_{l-1}(\mathbf{y},\mathbf{z})] \\ &= \mathbb{E}[Q_L(\mathbf{y},\mathbf{z}) - Q_{L-1}(\mathbf{y},\mathbf{z})] \sum_{l=L+1}^{\infty} \frac{\mathbb{E}[Q_l(\mathbf{y},\mathbf{z}) - Q_{l-1}(\mathbf{y},\mathbf{z})]}{\mathbb{E}[Q_L(\mathbf{y},\mathbf{z}) - Q_{L-1}(\mathbf{y},\mathbf{z})]} \end{aligned} \quad (37)$$

With the assumption that $\mathbb{E}[Q_l(\mathbf{y},\mathbf{z}) - Q_{l-1}(\mathbf{y},\mathbf{z})] \approx c{h_l}^q$, we have that

$$\frac{\mathbb{E}[Q_l(\mathbf{y},\mathbf{z}) - Q_{l-1}(\mathbf{y},\mathbf{z})]}{\mathbb{E}[Q_L(\mathbf{y},\mathbf{z}) - Q_{L-1}(\mathbf{y},\mathbf{z})]} \approx \frac{{h_l}^q}{{h_L}^q} = \beta^{(l-L)q} \tag{38}$$

Thus,

$$\varepsilon_{II} \leq \mathbb{E}[Q_L(\mathbf{y},\mathbf{z}) - Q_{L-1}(\mathbf{y},\mathbf{z})] \sum_{k=1}^{\infty} \beta^{-kq} = \frac{1}{\beta^q - 1}\mathbb{E}[Q_L(\mathbf{y},\mathbf{z}) - Q_{L-1}(\mathbf{y},\mathbf{z})] \tag{39}$$

From this, the condition we use for adding levels is

$$\max_{j\in\{0,1,2\}} \frac{\beta^{-jq}}{\beta^q - 1}\mathbb{E}[Q_{L-j}(\mathbf{y},\mathbf{z}) - Q_{L-j-1}(\mathbf{y},\mathbf{z})] \leq (1-\theta)(1-\alpha)\varepsilon_{TOL} \tag{40}$$

Now, assuming that we have made the assignment $L \leftarrow L+1$, in order to compute the new sample size M_l by (36), we must compute the variances $\{V_l\}_{l=0}^{L}$. Setting $Q_{-1}(\mathbf{y},\mathbf{z}) = 0$, the variances are approximated according to

$$\begin{aligned} V_l &\approx \frac{1}{M_l}\sum_{m_l=1}^{M_l}\left(\left(Q_l(\mathbf{y}^{m_l},\mathbf{z}) - Q_{l-1}(\mathbf{y}^{m_l},\mathbf{z})\right)^2 - \mu_l(\mathbf{z})^2\right) \\ \mu_l(\mathbf{z}) &= \frac{1}{M_l}\sum_{m_l=1}^{M_l}\left(Q_l(\mathbf{y}^{m_l},\mathbf{z}) - Q_{l-1}(\mathbf{y}^{m_l},\mathbf{z})\right) \end{aligned} \tag{41}$$

When a new level is added, the variance at the new level, V_L, cannot be calculated with (41). However, we can estimate the variance V_L by utilizing the strong error estimate (32) and assuming that $\mathbb{V}[Q_l(\mathbf{y},\mathbf{z}) - Q_{l-1}(\mathbf{y},\mathbf{z})] \approx c{h_l}^{q'}$. The variance at level L is thus approximated by

$$V_L \approx \beta^{-q'} V_{L-1} \tag{42}$$

Then, the number of samples $\{V_l\}_{l=0}^{L}$, including the newest level are updated with (36). The MLMC algorithm is outlined in

Algorithm 1 MLMC Algorithm

Start with L = 2 and generate a mesh hierarchy $\{h_l\}_{l=0}^{L}$ according to (27).
Choose an initial set $\{M_l\}_{l=0}^{L}$ of samples.
loop
Approximate $\{V_l\}_{l=0}^{L}$ by (41).
Update the optimal number of samples $\{M_l\}_{l=0}^{L}$ by (36).
if (40) is satisfied
Compute $\mathcal{A}_{MLMC}$ by (28) and terminate the loop.
else
Set $L := L + 1$ and $h_L = h_0\beta^{-L}$.
Approximate V_L by (42) and compute $\{M_l\}_{l=0}^{L}$ by (36).
end if
end loop

In [17], it is shown that the optimal cost of this implementation of multi-level Monte Carlo sampling is given by

$$W_{OPT} \propto \mathcal{O}(((1-\alpha)\varepsilon_{TOL})^{-2}) \tag{43}$$

5. Numerical Examples

In this section, we consider two examples. The first example involves an ODE, and the second example involves a second-order hyperbolic PDE.

For both examples, we compute optimal parameters for both Monte Carlo sampling and multi-level Monte Carlo sampling in accordance with Sections 3 and 4, run the sampling algorithms for a set of values of $\mathbf{z}$, which is a scalar in the first example, and a two-element vector in the second example, we verify that both methods are able to achieve specified tolerances, and compare the costs of the two methods for different tolerances. For both examples, we also show how minimum and maximum failure probabilities can be computed to within a specified tolerance by utilizing MLMC coupled with an implementation of the Nelder-Mead direct simplex search method [19]. Finally, we show approximate failure probability plots for both examples to confirm that the optimization method has found appropriate estimates for minimum and maximum failure probabilities.

5.1. An ODE with interval-valued random coefficient

Consider the following second-order linear ordinary differential equation with constant coefficients, which describes a free harmonic oscillator. The

initial position is 1, and the initial velocity is 0.

$$\begin{aligned} u''(t,\mathbf{y},\mathbf{z}) + 100a^2(\mathbf{y},\mathbf{z})u(t,\mathbf{y},\mathbf{z}) &= 0 \text{ on } [0,1]\times[1-\mathbf{z},1+\mathbf{z}] \\ u(0,\mathbf{y},\mathbf{z}) &= 1 \text{ on } \{0\}\times[1-\mathbf{z},1+\mathbf{z}] \\ u'(0,\mathbf{y},\mathbf{z}) &= 0 \text{ on } \{0\}\times[1-\mathbf{z},1+\mathbf{z}] \end{aligned} \tag{44}$$

Here, $\mathbf{y}$ is a uniform random variable between 0 and 1, and $\mathbf{z}$ is a fixed, deterministic parameter defining the distribution of the interval-valued random variable a, which is given by $a(\mathbf{y},\mathbf{z}) = 1 + \mathbf{z}(2\mathbf{y}-1)$. Thus, a is a uniform random variable between $1-\mathbf{z}$ and $1+\mathbf{z}$. Let the probability box for this example to be the set of all CDFs for which $\mathbf{z} \in [0.3, 1]$. The quantities of interest are the lower and upper bounds of the failure probability of the system. Define the limit state function to be $g(\mathbf{y},\mathbf{z}) = u(t=1,\mathbf{y},\mathbf{z}) + 0.9$ with $g \leq 0$ denoting failure.

The exact interval-valued solution of (44) is given by

$$\tilde{u}(t,\mathbf{y},\mathbf{Z}) = \cos(10\tilde{a}(\mathbf{y},\mathbf{Z})t) \tag{45}$$

For solving numerically each realization of (44), we use the 4^{th}-order Runge-Kutta method with parameter h denoting the time step size.

In order to estimate failure probabilities with the Monte Carlo sampling approach, we must first obtain accurate estimates of the parameters c_1 and q shown in (15) as well as $\mathbb{V}[Q_h(\mathbf{y},\mathbf{z})]$, which is the variance computed using the optimal step size. We estimate these values by taking 11 linearly-spaced values of $\mathbf{z}$ between 0.3 and 1, and for each value of $\mathbf{z}$, using the same set of time step sizes $\{h\}_{i=1}^5 = [\frac{1}{10}, \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}]$ and the same set of 20×10^6 random samples $\mathbf{y}$, compute the errors given by $|\mathbb{E}[Q_{h_i}(\mathbf{y},\mathbf{z}) - Q(\mathbf{y},\mathbf{z})]|$ for $i = 1,2,3,4,5$. For each value of $\mathbf{z}$, linear regression is used to estimate c_1 and q. To be conservative, the largest value of c_1 is taken to be c_1, and the smallest value of q is taken to be q. We take $\mathbb{V}[Q_h(\mathbf{y},\mathbf{z})]$ to be the value of $\max_{\mathbf{z},\mathbf{h}} \mathbb{V}[Q_h(\mathbf{y},\mathbf{z})]$ over these 11 values of $\mathbf{z}$ and over the step sizes $\{h\}_{i=1}^5 = [\frac{1}{10}, \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}]$ rounded up to the ceiling of the second decimal place. Finally, we must decide on the parameter c_2, given in (17), which defines the confidence for the error given in (16). The parameters for implementing Monte Carlo Sampling for this example are summarized in Table 1. The value of $c_2 = 1.96$ in Table 1 corresponds to a 95% confidence level.

To apply the interval multi-level Monte Carlo method, we must estimate the quantities q and q' in (29) and (32), respectively. We estimate the value

Table 1: Parameters for Monte Carlo sampling.

Quantity	Value
c_1	2.2×10^4
q	3.9
c_2	1.96
$\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]$	0.15

of q by taking 11 linearly-spaced values of $\mathbf{z}$ between 0.3 and 1, and for each value of $\mathbf{z}$, using the same set of time step sizes $\{h\}_{i=1}^{5} = [\frac{1}{10}, \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}]$ and the same set of 10^6 random samples $\{\mathbf{y}\}$, compute the errors given by $|\mathbb{E}[Q_{h_i}(\mathbf{y}, \mathbf{z}) - Q_{h_{i+1}}(\mathbf{y}, \mathbf{z})]|$ for $i = 1, 2, 3, 4$. Similar computations are done in order to estimate q', except that the errors are given by $|\mathbb{V}[Q_{h_i}(\mathbf{y}, \mathbf{z}) - Q_{h_{i+1}}(\mathbf{y}, \mathbf{z})]|$ for $i = 1, 2, 3, 4$. For each value of $\mathbf{z}$, linear regression is used to estimate q and q'. Again, to be conservative, we take the smallest value of q to be the value of q, and the smallest value of q' to be q'. The error-splitting parameter, θ is chosen to be 0.9, which means that most of the error will be absorbed by the statistical error, ϵ_{II}. Shown below in Table 2 is a summary of the parameters used for the multi-level Monte Carlo computations for this example.

Quantity	Value
q	3.9
q'	3.7
c_2	1.96
β	2
θ	0.9
h_0	0.1
M_0	1000

Table 2: Parameters for multi-level Monte Carlo sampling.

We compute failure probabilities using a set of tolerances $(1 - \alpha)\varepsilon_{TOL}$. To verify that failure probabilities are calculated within these tolerances, we compute reference failure probabilities for the same set of 11 input parameters $\mathbf{z}$. The reference failure probabilities are computed using the exact solution of one realization (45) with tolerance of 5×10^{-5} via Monte Carlo sampling. In computing these reference values, the same parameters in Table 1 are used.

Using 11 linearly-spaced values of $\mathbf{z}$ between 0.3 and 1, failure probabilities are computed for each of the tolerances using both Monte Carlo sampling and multi-level Monte Carlo sampling. For each tolerance, the maximum difference over the 11 values of $\mathbf{z}$ between the reference failure probability and that calculated using either the Monte Carlo sampling or the multi-level Monte Carlo sampling is recorded. Because we are using reference values, each with tolerance $\delta = 5 \times 10^{-4}$, in order to verify that the true errors are less than $(1-\alpha)\varepsilon_{TOL}$, we must verify that the absolute difference between the test failure probabilities and the references values are less than $(1-\alpha)\varepsilon_{TOL} - \delta$. This is indeed the case, as shown in Figure 1. This plot shows how for both methods, the desired tolerances are achieved.

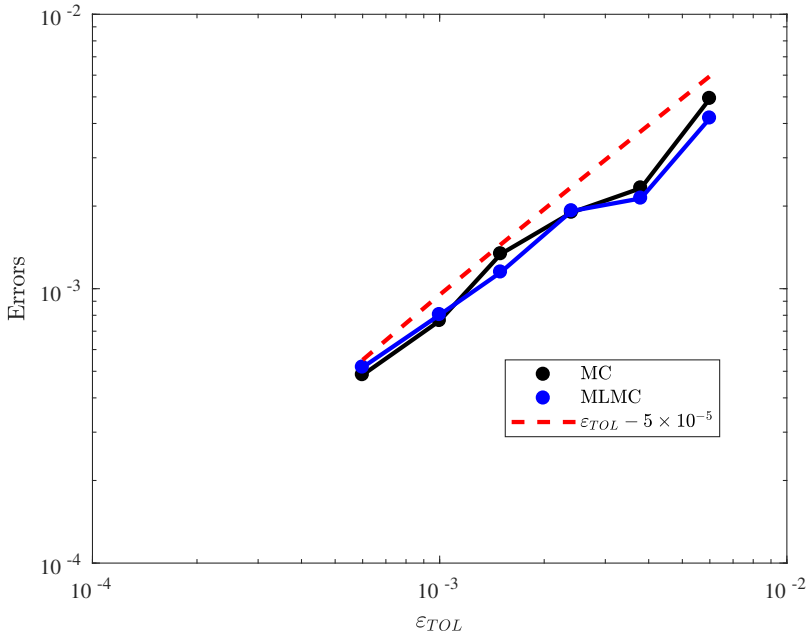

Figure 1: Verification that desired tolerances are achieved using both Monte Carlo and multi-level Monte Carlo sampling methods.

Figure 2 compares the costs of Monte Carlo sampling with multi-level Monte Carlo sampling. The numerical experiments are seen to agree with what is expected from the numerical analysis.

If we can have a reasonably nice failure probability function which we can roughly estimate and thus have some knowledge of the function and where its extrema are located, we can consider using a smart optimization

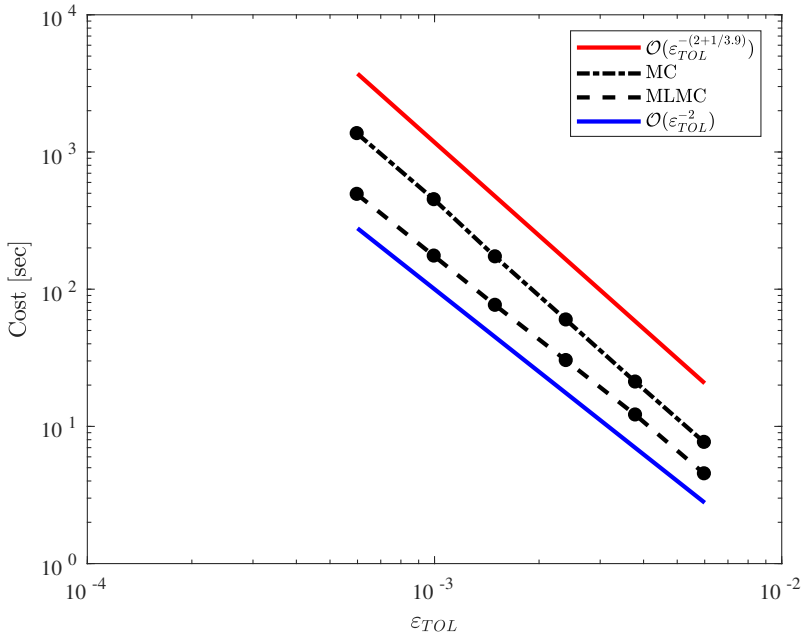

Figure 2: Computational cost is seen to be in agreement with what is predicted from numerical analysis.

strategy instead of a grid search over a large set of $\mathbf{z}$-values, which would be highly expensive. We use an implementation of the Nelder-Mead direct simplex search for both examples in this paper. To compute minimum and maximum failure probabilities, we use the initial guess $\mathbf{z}_0 = 0.4$ for the minimum and $\mathbf{z}_0 = 0.8$ for the maximum. The Nelder-Mead optimization is terminated when both (1) the maximum Euclidean distance between two points in the final simplex is within 4×10^{-4} and (2) the maximum difference in absolute value between failure probability results in the final simplex is within 4×10^{-4}. The chosen minimum is taken to be the point $\mathbf{z}$ in the final simplex which minimizes the failure probability. The maximization problem is handled by minimizing the negative of the failure probability function and changing the sign of the result found. For the one-dimensional optimization in this example, the simplexes are line segments. In general, for an optimization over n-dimensional space, the simplexes will each contain $n+1$ points. Figure 3 shows the failure probability plot with 1001 linearly-spaced values of $\mathbf{z}$, where, for each value of $\mathbf{z}$, the failure probability is computed with tolerance of 10^{-3}. Shown on the plot are the minimum and maximum failure

probability results computed using the MLMC-Nelder-Mead optimization approach with $\varepsilon_I + \varepsilon_{II} \leq 6 \times 10^{-4} = (1-\alpha)\varepsilon_{TOL}$ and $\alpha\varepsilon_{TOL} 4 \times 10^{-4}$, so that the minimum and maximum failure probabilities are calculated within a tolerance of $(1-\alpha)\varepsilon_{TOL} + \alpha\varepsilon_{TOL} = 10^{-3}$. For this example then, $\alpha = 0.4$.

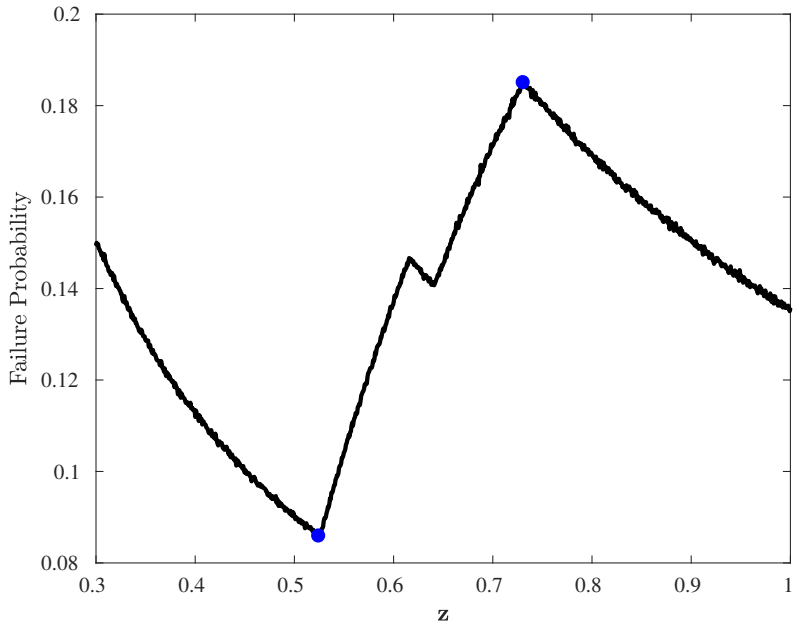

Figure 3: The MLMC strategy coupled with the Nelder-Mead direct simplex search produces the estimates shown in blue for which $\epsilon_{TOL} = 10^{-3}$.

For this example, we can easily contrast the results for upper and lower failure probability bounds for a given tolerance using the estimators in either (12) or (26) with the results obtained using the excessively conservative estimators (13). We choose a tolerance of $\epsilon_{TOL} = 6 \times 10^{-4}$ and use the same optimal time step size and sample size for computing (13) as were used for computing (12). In computing the estimators (13), a grid search is done over 11 linearly-spaced value of $\mathbf{z}$ between 0.3 and 1 for each random input sample $\mathbf{y}$. In computing the estimators according to (12) and (26), a grid search is done over the same values of z for each expected value estimate. The results are summarized in Table 3.

As seen in Table 3, the failure probability bounds computed from (13) are much less accurate than those computed using the other methods. The upper bound is highly overestimated, and the lower bound is underestimated.

p_f Estimator	$\underline{\mathcal{P}}_f \times 10^2$	$\overline{\mathcal{P}}_f$
(12)	8.8	0.180
(26)	8.8	0.180
(13)	5.1	0.581

Table 3: Comparison of failure probability estimates using three different approaches.

Having demonstrated the superior performance of the multi-level Monte Carlo sampling for computing accurate failure probability bounds for this example, we now apply a similar analysis to a time-dependent PDE problem.

5.2. A Time-Dependent PDE with interval-valued random variables

Consider the following partial differential equation with constant coefficients.

$$u_{tt}(\mathbf{x}, t, \mathbf{y}, \mathbf{z}) - \nabla^2 u(\mathbf{x}, t, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}, t, \mathbf{y}, \mathbf{z}) \text{ in } [0, 1] \times D \times \Gamma, \tag{46}$$

with $t \in [0, 1]$, $D = \mathbf{x} = [x_1, x_2]^T \in [-\frac{1}{2}, \frac{1}{2}]^2$, and $\mathbf{y} \in \Gamma \subset \mathbb{R}^2$. Here, $\mathbf{y} = [y_1, y_2]^T$, where y_1 and y_2 are standard uniform random variables. For each realization, the vector $\mathbf{z} = [z_1, z_2]^T$ is a fixed, deterministic vector defining the uniform distribution of the vector $\mathbf{b} = [b_1, b_2]^T$ defined by $b_1 = z_1 + 0.5(2y_1 - 1)$ and $b_2 = z_2 + 0.5(2y_2 - 1)$. That is, b_1 varies uniformly between $z_1 - 0.5$ and $z_1 + 0.5$, and b_2 varies uniformly between $z_2 - 0.5$ and $z_2 + 0.5$. The probability box for this example is the set of all CDFs for which $z_1, z_2 \in [1, 1.8]$. The forcing function, f, is given by

$$f(\mathbf{x}, t, b_1, b_2) = -50 b_1 b_2 \cos(5b_1(t + x_1)) \sin(5b_2(t + x_2)) \tag{47}$$

Finally, define the limit state function to be $g(\mathbf{y}, \mathbf{z}) = u(t = 1, \mathbf{y}, \mathbf{z}) + 0.5$ with $g \leq 0$ denoting failure.

The exact interval-valued solution of (46) is given by

$$\tilde{u}(\mathbf{x}, t, \mathbf{y}, \mathbf{Z}) = u(\mathbf{x}, t, \tilde{b_1}, \tilde{b_2}) = \sin(5\tilde{b_1}(t + x_1)) \cos(5\tilde{b_2}(t + x_2)) \tag{48}$$

For solving numerically each realization of (46), we use 2^{nd}-order finite differencing with parameter h, denoting the grid size. For example, using a grid size of $\frac{1}{12}$ equates to setting the number of grid points in both x_1 and x_2-directions to be 13.

As for the previous example, we must obtain accurate estimates of the parameters c_1 and q shown in (15) as well as $\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]$ in order to estimate

failure probabilities with the Monte Carlo sampling approach. We estimate these values by taking 7 linearly-spaced values of both z_1 and z_2, each between 1 and 1.8, thereby forming a grid of 49 sample vectors $\mathbf{z}$. For each value of $\mathbf{z}$, using the same set of grid step sizes $\{h\}_{i=1}^{5} = [\frac{1}{6}, \frac{1}{12}, \frac{1}{24}, \frac{1}{48}]$ and the same set of 20×10^4 random sample vectors $\mathbf{y}$, we compute the error estimates given by $|\mathbb{E}[Q_{h_i}(\mathbf{y}, \mathbf{z}) - Q(\mathbf{y}, \mathbf{z})]|$ for $i = 1, 2, 3, 4$. For each value of $\mathbf{z}$, linear regression is used to estimate c_1 and q. The largest regression estimate of c_1 is taken to be c_1, and the smallest regression estimate of q is taken to be q. We take $\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]$ to be the value of $\max_{\mathbf{z},\mathbf{h}} \mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]$ over these 49 values of $\mathbf{z}$ and over the step sizes $\{h\}_{i=1}^{4} = [\frac{1}{6}, \frac{1}{12}, \frac{1}{24}, \frac{1}{48}]$ rounded up to the ceiling of the second decimal place. We use the same value of the confidence parameter, c_2, as was used in the previous example. The parameters for implementing Monte Carlo sampling for this example are summarized in Table 4.

Table 4: Parameters for Monte Carlo sampling.

Quantity	Value
c_1	4.2
q	1.6
c_2	1.96
$\mathbb{V}[Q_h(\mathbf{y}, \mathbf{z})]$	0.21

To apply the interval multi-level Monte Carlo method, we must estimate the quantities q and q' in (**??**) and (32), respectively. We estimate the value of q by taking the same set of 49 values of the vector $\mathbf{z}$, and for each value of $\mathbf{z}$, using the same set of grid sizes $\{h\}_{i=1}^{5} = [\frac{1}{6}, \frac{1}{12}, \frac{1}{24}, \frac{1}{48}, \frac{1}{96}]$ and the same set of 2×10^4 random sample vectors $\mathbf{y}$, compute the errors given by $|\mathbb{E}[Q_{h_i}(\mathbf{y}, \mathbf{z}) - Q_{h_{i+1}}(\mathbf{y}, \mathbf{z})]|$ for $i = 1, 2, 3, 4$. Similar computations are done in order to estimate q', except that the errors are given by $|\mathbb{V}[Q_{h_i}(\mathbf{y}, \mathbf{z}) - Q_{h_{i+1}}(\mathbf{y}, \mathbf{z})]|$ for $i = 1, 2, 3, 4$. For each value of $\mathbf{z}$, linear regression is used to estimate q and q'. Again, to be conservative, we take the smallest value of q to be the value of q, and the smallest value of q' to be q'. As for the first example, we choose the error-splitting parameter, θ to be 0.9. Shown in Table 5 is a summary of the parameters used for the Multi-Level Monte Carlo computations for this example.

We compute failure probabilities using different sets of tolerances for Monte Carlo sampling than for multi-level Monte Carlo sampling. To verify that failure probabilities are calculated within these tolerances, we compute

Quantity	Value
q	1.3
q'	1.6
c_2	1.96
β	2
θ	0.9
h_0	$\frac{1}{12}$
M_0	100

Table 5: Parameters for Multi-Level Monte Carlo sampling.

reference failure probabilities for the same set of 49 input parameter vectors $\mathbf{z}$. The reference failure probabilities are computed using the exact solution of one realization (48) with tolerance of 5×10^{-5} via Monte Carlo sampling with the same parameters that appear in Table 4. Using the 49 values of $\mathbf{z}$ with both $\mathbf{z_1}$ and $\mathbf{z_2}$ between 1 and 1.8, failure probabilities are computed using both Monte Carlo sampling and multi-level Monte Carlo sampling. For each tolerance, the maximum difference over the 49 values of $\mathbf{z}$ between the reference failure probability and that calculated using either the Monte Carlo sampling or the multi-level Monte Carlo sampling is recorded. As for the previous example, we subtract the reference tolerance $\delta = 5 \times 10^{-4}$ from the test tolerances and plot the results in Figure 4. This plot shows how for both methods, the desired tolerances are achieved.

Figure 5 compares the costs of Monte Carlo sampling with multi-level Monte Carlo sampling. The numerical experiments are seen to agree with what is expected from the numerical analysis.

To compute minimum and maximum failure probabilities with the Nelder-Mead direct simplex search, we use the initial guess $\mathbf{z}_0 = [1.55, 1.4]$ for the minimum and the initial guess $\mathbf{z}_0 = [1.25, 1.4]$ for the maximum. The Nelder-Mead optimization is terminated with both (1) the maximum Euclidean distance between two points in the final simplex is within 4×10^{-3} and (2) the maximum difference in absolute value between failure probability results in the final simplex is within 4×10^{-3}. For this example, the simplexes are triangles. Figure 6 shows the failure probability plot with $51 \times 51 = 2601$ values of $\mathbf{z}$, where, for each value of $\mathbf{z}$, the failure probability is computed with tolerance of 10^{-2}. Shown on the plot are the minimum and maximum failure probability results computed using the MLMC-Nelder-Mead optimization approach with $\varepsilon_I + \varepsilon_{II} \leq 6 \times 10^{-3} = (1-\alpha)\varepsilon_{TOL}$ and $\alpha\varepsilon_{TOL} = 4 \times 10^{-3}$, so

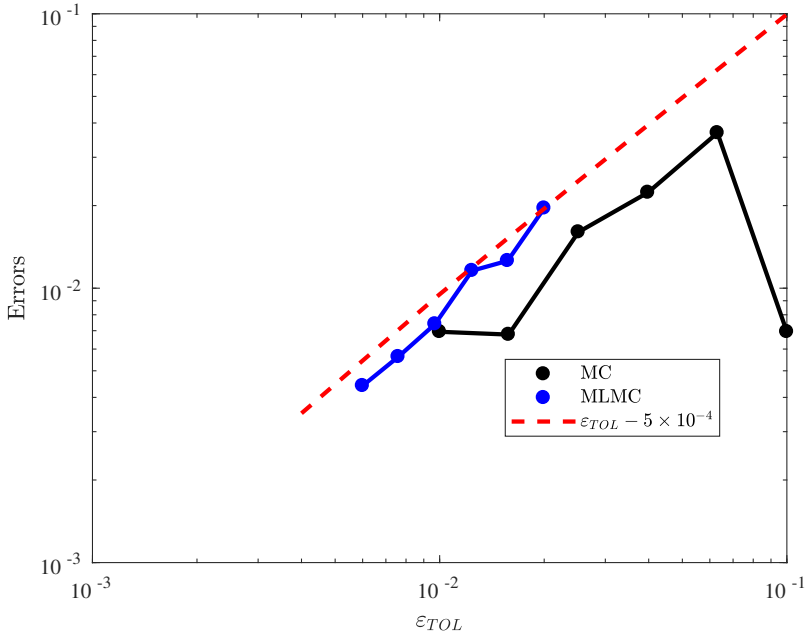

Figure 4: Verification that desired tolerances are achieved using both Monte Carlo and multi-level Monte Carlo sampling methods.

that the minimum and maximum failure probabilities are calculated within a tolerance of $\varepsilon_{TOL} = (1-\alpha)\varepsilon_{TOL} + \alpha\varepsilon_{TOL} = 10^{-2}$. For this example as well, we chose $\alpha = 0.4$.

We can again contrast the results for upper and lower failure probability bounds for a given tolerance using the estimators in either (12) or (26) with the results obtained using the excessively conservative estimators (13). We choose a tolerance of 6×10^{-3} and use the same optimal time step size and sample size for computing (13) as were used for computing (12). In computing the estimators (13), a grid search is done over the same 49 values of $\mathbf{z}$ for each random input sample $\mathbf{y}$. In computing (12) and (26), a grid search is done over the same values of $\mathbf{z}$ for each expected value estimate. The results are summarized in Table 6.

As seen in Table 6, although the tolerance of 10^{-2} is reasonably small, the failure probability bounds computed from (13) are as wide as possible and provide essentially no information about the true failure probabilities.

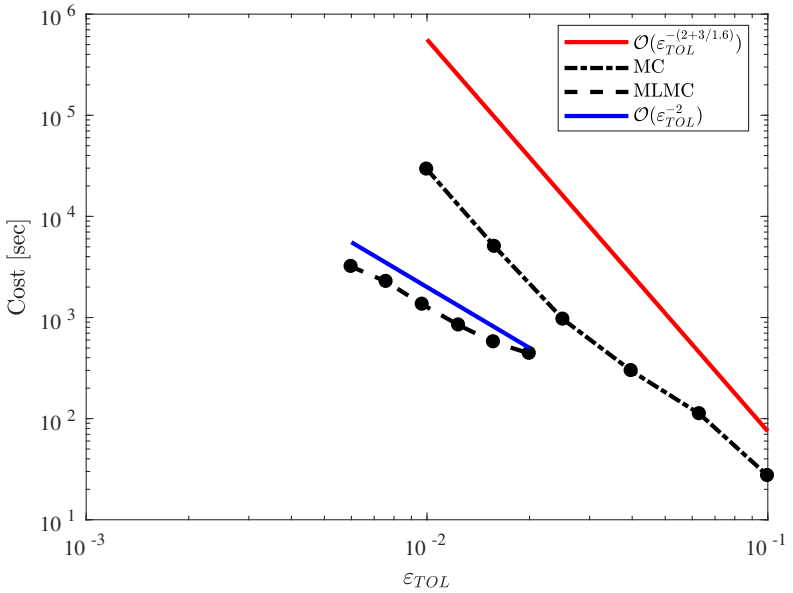

Figure 5: Computational cost is seen to be in agreement with what is predicted from numerical analysis.

P_f Estimator	$\underline{\mathcal{P}}_f \times 10^2$	$\overline{\mathcal{P}}_f$
(12)	8	0.26
(26)	9	0.27
(13)	0	1

Table 6: Comparison of failure probability estimates using three different approaches.

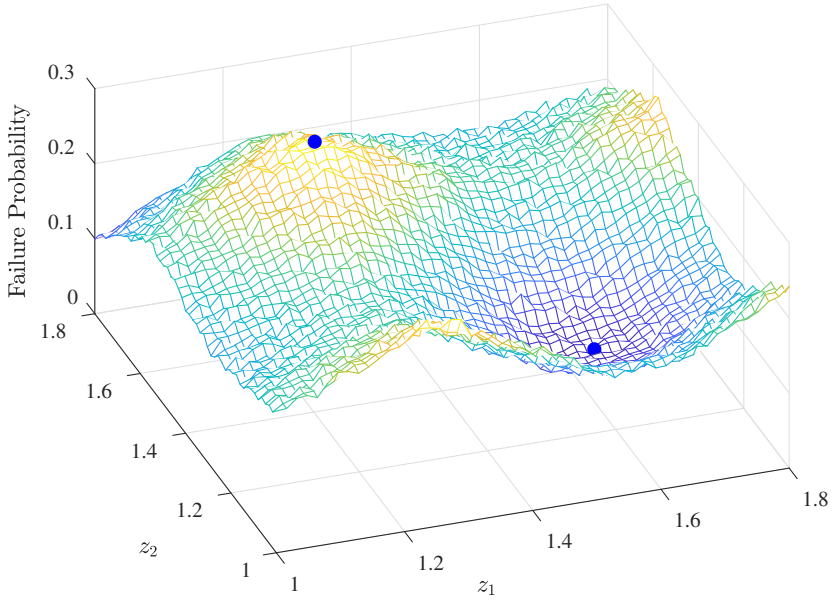

Figure 6: The MLMC strategy coupled with the Nelder-Mead direct simplex search produces the estimates shown in blue for which $\epsilon_{TOL} = 10^{-2}$.

6. Conclusions

We have presented the problem of computing interval-valued probabilities, with one application being the calculation of failure probabilities for physical systems. Two methods were discussed in detail; the use of either standard MC sampling or multi-level MC sampling, coupled with a global optimization method. These two methods were analyzed for their computational cost in estimating interval valued probabilites to within a specified tolerance. Several numerical examples were presented in order to verify the theoretical cost estimates. We have demonstrated that the use of the multi-level MC allows for much more efficient computation of interval-valued probabilities, with the improvement in efficiency over MC increasing with decreasing error tolerance.

7. References

[1] I. Babuška and J. T. Oden. Verification and validation in computational engineering and science: Part I, basic concepts. *Computer Methods in Applied Mechanics and Engineering*, 193:4047–4068, 2004.

[2] I. Babuška, R. Tempone, and G. E. Zouraris. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Computer Methods in Applied Mechanics and Engineering*, 194:1251–1294, 2005.

[3] M. Berblinger, Ch. Schlier, and T. Weiss. Monte carlo integration with quasi-random numbers: experience with discontinuous integrands. *Computer Physics Communications*, 99(2):151 – 162, 1997.

[4] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Visual Sci.*, 14:3–15, 2011.

[5] D. Dubois and H. Prade. Formal representations of uncertainty. In D. Bouyssou, D. Dubois, H. Prade, and M. Pirlot, editors, *Decision Making Process: Concepts and Methods*, pages 85–157. Wiley-ISTE, 2010.

[6] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications.* Springer- Verlag, New York, 1996.

[7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2004.

[8] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: A spectral approach*. Springer, New York, 1991.

[9] M. Grigoriu. *Stochastic Calculus, Applications in Sciences and Engineering*. Birkhäuser, Boston, 2002.

[10] A.-L. Haji-Ali, F. Nobile, and R. Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numerische Mathematik*, 132:767–806, 2016.

[11] T. Y. Hou and X. Wu. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *J. Comput. Phys.*, 230:3668–3694, 2011.

[12] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.

[13] J. Kaipo and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, New York, 2005.

[14] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: new perspectives on classical and modern methods. *SIAM Review*, 45:385–482, 2003.

[15] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. SIAM Press, Philadelphia, Pennsylvania, 2009.

[16] B. Moskowitz and R.E. Caflisch. Smoothness and dimension reduction in quasi-monte carlo methods. *Mathematical and Computer Modelling*, 23(8):37 – 54, 1996.

[17] M. Motamed and D. Appelö. A multi-order discontinuous Galerkin Monte Carlo method for hyperbolic problems with stochastic parameters. *SIAM J. Numer. Anal.*, 56:448–468, 2018.

[18] M. Motamed, F. Nobile, and R. Tempone. A stochastic collocation method for the second order wave equation with a discontinuous random speed. *Numer. Math.*, 123:493–536, 2013.

[19] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[20] A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13:15–83, 2004.

[21] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, NJ, 1976.

[22] T. J. Sullivan. *Introduction to Uncertainty Quantification*. Springer, 2015.

[23] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

[24] P. Walley. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning*, 24:125–148, 2000.

[25] K. Weichselberger. The theory of interval probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, 24:149–170, 2000.

[26] D. Xiu. *Numerical Methods For Stochastic Computations: A spectral method approach*. Princeton University Press, 2010.

[27] D. Xiu and J. S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27:1118–1139, 2005.

[28] H. Zhang, H. Dai, M. Beer, and W. Wang. Structural reliability analysis on the basis of small samples: An interval quasi-monte carlo method. *Mechanical Systems and Signal Processing*, 37(2):137–151, 2013.