International Journal for Uncertainty Quantification, 4 (2): 133–149 (2014)

GAUSSIAN PROCESS ADAPTIVE IMPORTANCE SAMPLING

Keith Dalbey & Laura Swiler*

Sandia National Laboratories, P.O. Box 5800, M.S. 0670, Albuquerque, New Mexico 87185, USA

Original Manuscript Submitted: 09/26/2012; Final Draft Received: 06/24/2013

The objective is to calculate the probability, P_{F}, that a device will fail when its inputs, $\mathbf{x}$, are randomly distributed with probability density, $p(\mathbf{x})$, e.g., the probability that a device will fracture when subject to varying loads. Here failure is defined as some scalar function, $y(\mathbf{x})$, exceeding a threshold, T. If evaluating $y(\mathbf{x})$ via physical or numerical experiments is sufficiently expensive or P_{F} is sufficiently small, then Monte Carlo (MC) methods to estimate P_{F} will be unfeasible due to the large number of function evaluations required for a specified accuracy. Importance sampling (IS), i.e., preferentially sampling from "important" regions in the input space and appropriately down-weighting to obtain an unbiased estimate, is one approach to assess P_{F} more efficiently. The inputs are sampled from an importance density, $p'(\mathbf{x})$. We present an adaptive importance sampling (AIS) approach which endeavors to adaptively improve the estimate of the ideal importance density, $p^*(\mathbf{x})$, during the sampling process. Our approach uses a mixture of component probability densities that each approximate $p^*(\mathbf{x})$. An iterative process is used to construct the sequence of improving component probability densities. At each iteration, a Gaussian process (GP) surrogate is used to help identify areas in the space where failure is likely to occur. The GPs are not used to directly calculate the failure probability; they are only used to approximate the importance density. Thus, our Gaussian process adaptive importance sampling (GPAIS) algorithm overcomes limitations involving using a potentially inaccurate surrogate model directly in IS calculations. This robust GPAIS algorithm performs surprisingly well on a pathological test function.

KEY WORDS: *uncertainty quantification, probability theory, Monte Carlo, mixture models*

1. INTRODUCTION

The objective is to calculate the probability, P_{F}, that a device will fail when its inputs, $\mathbf{x}$, are randomly distributed with probability density, $p(\mathbf{x})$, e.g. the probability that a device will fracture when subject to varying loads. Failure is defined as some deterministic scalar function, $y(\mathbf{x})$, exceeding a threshold, T. We assume that $y(\mathbf{x})$ can be evaluated by performing physical or numerical experiments. Specifically, this method has been designed to work when $y(\mathbf{x})$ is calculated by a black-box simulation code about which we have limited information: we assume the user has no *a priori* knowledge about where important regions reside in the input space.

The probability of failure can be thought of as the mean rate of occurrence of failure. The Monte Carlo (MC) estimate of P_{F} is therefore the sample mean of the indicator function, $I(\mathbf{x})$,

$$P_{\mathrm{MC}} = \frac{1}{N}\sum_{i=1}^{N} I(\mathbf{x}_i) \quad \mathbf{x} \sim p(\mathbf{x}), \tag{1}$$

where N samples, $\mathbf{x}_i$, are drawn from $p(\mathbf{x})$, and the indicator function, $I(\mathbf{x})$, is 1 if failure occurs and zero otherwise. For example, if failure occurs when $y(\mathbf{x}) > T$, then

$$I(\mathbf{x}) = \begin{cases} 1 & \text{if } y(\mathbf{x}) > T \\ 0 & \text{otherwise} \end{cases}.$$

*Correspond to Keith Dalbey, E-mail: kdalbey@sandia.gov

2152–5080/14/$35.00

Similarly, if failure occurs when $y(\mathbf{x}) < T$, then

$$I(\mathbf{x}) = \begin{cases} 1 & \text{if } y(\mathbf{x}) < T \\ 0 & \text{otherwise} \end{cases}.$$

The central limit theorem states that, for sufficiently large N, the error in a mean of a function g computed by MC is normally distributed about zero with standard deviation $\sigma_{\mathrm{err_{MC}}} = \sqrt{\sigma_g^2/N}$. Here σ_g^2 is the variance of the function g. When the probability of failure P_{F} is the quantity of interest and P_{F} is small, this works out to

$$\sigma_{\mathrm{err_{MC}}} = \sqrt{\frac{\sigma_I^2}{N}} = \sqrt{\frac{P_{\mathrm{F}} - P_{\mathrm{F}}^2}{N}} = \sqrt{\frac{P_{\mathrm{F}}^2 - P_{\mathrm{F}}^3}{NP_{\mathrm{F}}}} \approx \frac{P_{\mathrm{F}}}{\sqrt{N_{\mathrm{F}}}}. \tag{2}$$

Here N_{F} is the number of samples that hit the failure region. Thus, to get two significant figures of accuracy in the estimate of P_{F}, MC requires roughly $N = 10^4/P_{\mathrm{F}}$ samples. The cost of MC is prohibitive if evaluating $y(\mathbf{x})$ is sufficiently expensive or P_{F} is sufficiently small. MC's error converges to zero at a rate of $\mathcal{O}\left(N^{-1/2}\right)$. Latin hypercube sampling (LHS), in which the dimensions have been paired through random permutation, gives an estimator for a function mean that has lower variance than MC for any function having finite second moment [1, 2]. Further, the convergence behavior of LHS improves if the function is additively separable. For many problems, however, the cost of LHS is still prohibitive.

The outline of the paper is as follows: Section 2 discusses importance sampling and the use of nonparametric methods in importance sampling, Section 3 presents the Gaussian process adaptive importance sampling algorithm that we have developed (including motivation, proofs, and implementation details), Section 4 presents the results of our approach applied to a variety of test problems, and Section 5 provides conclusions.

2. IMPORTANCE SAMPLING

Importance sampling (IS) is a technique to reduce the error variance of Monte Carlo by drawing more samples from "important" regions and appropriately down-weighting them to obtain an unbiased estimate [3, 4]. Instead of taking the sample mean of the indicator function as in Eq. (1), where the samples are drawn from the nominal probability density $p(\mathbf{x})$, IS draws samples from the importance density $p'(\mathbf{x})$ and scales the sample mean by the importance density:

$$P_{\mathrm{IS}} = \frac{1}{N}\sum_{i=1}^{N}\left(I(\mathbf{x}_i)\frac{p(\mathbf{x}_i)}{p'(\mathbf{x}_i)}\right) \quad \mathbf{x} \sim p'(\mathbf{x}). \tag{3}$$

This reduces the asymptotic error variance from $\sigma^2_{\mathrm{err_{MC}}} = \mathrm{E}\left[(I(\mathbf{x}) - P_{\mathrm{F}})^2\right]/N$ to

$$\sigma^2_{\mathrm{err_{IS}}} = \frac{\mathrm{E}\left[\left(I(\mathbf{x})\frac{p(\mathbf{x})}{p'(\mathbf{x})} - P_{\mathrm{F}}\right)^2\right]}{N}. \tag{4}$$

Inspection of Eq. 4 reveals $\sigma^2_{\mathrm{err_{IS}}} = 0$ if $p'(\mathbf{x})$ equals the ideal importance density $p^*(\mathbf{x})$,

$$p^*(\mathbf{x}) = \frac{I(\mathbf{x})p(\mathbf{x})}{P_{\mathrm{F}}} = \alpha I(\mathbf{x})p(\mathbf{x}), \tag{5}$$

where $\alpha = 1/P_{\mathrm{F}}$. However, $p^*(\mathbf{x})$ is unknown *a priori* because $I(\mathbf{x})$ is only known where it has been evaluated. Therefore, the required P_{F} in the denominator (or α in the numerator) is also unknown; P_{F} is what we are trying to estimate. This is a serious drawback of the traditional approach to importance sampling.

If importance sampling is to be effective, the practitioner must be able to choose a good $p'(\mathbf{x})$ without already knowing $I(\mathbf{x})$ everywhere. There is a danger though; a poor choice for $p'(\mathbf{x})$ can put most of the samples in unimportant regions and make $\sigma^2_{\mathrm{err_{IS}}}$ much greater than $\sigma^2_{\mathrm{err_{MC}}}$. It can be very challenging to generate an importance sampling

probability density for a general black-box application for which one cannot exploit any specific knowledge. Richard and Zhang [5] state, "The construction of importance samplers clearly constitutes the Achilles heel of importance sampling... importance samplers have to be carefully tailored to the problem under consideration. This has proved to be a significant obstacle to routine applications of importance sampling."

A classical approach is to assume that $p'(\mathbf{x})$ belongs to a parametric distribution family. Then, the problem is determining the values of the parameters governing that distribution [for example, determining the mean and variance for $p'(\mathbf{x})$ if p' is assumed to be normal]. Often these parameters are obtained by optimizing the variance of the importance sampling estimator, but this implies that one can calculate an analytic expression or approximation for this estimator. Oh and Berger [6] used a mixture distribution (a set of weighted individual distributions), where the individual distributions were multivariate-t distributions. They then had to determine the weights, location, and scale parameters of the t-distributions, which they did by numerical minimization of the estimate of the squared variation coefficient of the weight function.

The use of nonparametric approaches in importance sampling is fairly recent. The first papers were mid-1990s, and include Bayesian approaches (e.g., Givens and Raftery [7]) and kernel density estimators (Zhang [8]). Zhang outlines the rationale for going to nonparametric approaches and highlights the trade-offs of the increased convergence but higher computation of using nonparametric methods: "The most difficult part in parametric importance sampling is choosing a suitable distribution family to start with. There is no general recipe, and the issue remains largely a matter of art in the literature. Most parametric distributions fail to include the optimal importance sampling density as a member" [8].

The literature indicates that mixture importance sampling and adaptive importance sampling (AIS) are promising approaches for choosing a good $p'(\mathbf{x})$. Owen and Zhou [9] state that mixture importance sampling "is asymptotically not much worse than importance sampling from the best of the mixture components." Zhang [8] developed one of the first nonparametric importance sampling approaches based on kernel density estimators (KDE). Swiler and West [10] expanded this idea and presented an AIS approach meant to be used after an initial set of Latin hypercube samples has been taken to help refine a failure probability estimate. This approach to estimating the importance density $p^*(\mathbf{x})$ provides "a quick way to generate more samples in the failure region," but requires that one or more of the initial LHS samples hit each failure region. Thus, KDE-based AIS can perform poorly when P_{F} and the number of allowed samples is small and/or there are multiple disjoint failure regions. We will refer to this as Limitation 1 when we address it below.

Swiler and West [10] also explored fitting Gaussian process (GP) surrogates and calculating the probability that the surrogate exceeded the threshold. They effectively used the GP's expectation of $y(\mathbf{x})$, i.e., its adjusted mean, to make a binary approximation of $I(\mathbf{x})$. They discovered that such an approximation of the indicator function can perform poorly when the surrogate is inaccurate for some parts of the domain. We will refer to this as Limitation 2 below.

3. GAUSSIAN PROCESS ADAPTIVE IMPORTANCE SAMPLING

We propose a Gaussian process adaptive importance sampling (GPAIS) algorithm that combines ideas from mixture importance sampling with a nonparametric approach to AIS. Note that in the technical sense, it uses an ensemble rather than a "mixture" of distributions. Our approach is adaptive in the sense that the distributions in the ensemble are a sequence of improving GP approximations of the ideal importance density. Our algorithm also use a novel estimator with advantages over traditional estimator.

3.1 GPAIS Estimator

Let $\mathcal{E}_{\mathbf{x}} = \{p'_i(\mathbf{x}) \ \forall i = 1, 2, ..., N\}$ be a set or "ensemble" of N probability distributions defined over the domain of $\mathbf{x}$. Further, let one point be drawn randomly from each $p'_i(\mathbf{x})$. However, the N component distributions, the $p'_i(\mathbf{x})$s, are not required to be unique; i.e., there can be multiple copies of the "same distribution" in the ensemble which thus permits more than one sample to be drawn from a particular distribution. Then the *ensemble distribution*,

$$p^{\mathcal{E}_{\mathbf{x}}}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} p_i'(\mathbf{x}), \tag{6}$$

is the single distribution from which an arbitrary set of points $\mathbf{x}_i$ would be selected with the same probability as if they were drawn from the ensemble in the manner described above.

GPAIS estimates the probability of failure using:

- the N samples drawn from the ensemble, and
- the ensemble distribution as the sole importance distribution, $p'(\mathbf{x}) = p^{\mathcal{E}_{\mathbf{x}}}(\mathbf{x})$, i.e.,

$$P_{\text{GPAIS}} = \frac{1}{N}\sum_{i=1}^{N}\left(I(\mathbf{x}_i)\frac{p(\mathbf{x}_i)}{p^{\mathcal{E}_{\mathbf{x}}}(\mathbf{x}_i)}\right) \quad \mathbf{x} \sim \mathcal{E}_{\mathbf{x}}. \tag{7}$$

This GPAIS estimator should be compared to the following "traditional" importance sampling estimator for an ensemble of N importance distributions:

$$P_{\text{Trad}} = \frac{1}{N}\sum_{i=1}^{N}\left(I(\mathbf{x}_i)\frac{p(\mathbf{x}_i)}{p_i'(\mathbf{x}_i)}\right) \quad \mathbf{x}_i \sim p_i'(\mathbf{x}). \tag{8}$$

If the $\mathbf{x}_i \sim p_i'(\mathbf{x})$ for $i = 1, ..., N$ are independent, then the GPAIS estimator, Eq. (7), is unbiased with a variance that is less than or equal to the traditional estimator, Eq. (8). This is most easily seen in the discrete case where there is a large but finite number, M, values of $\mathbf{x}$. Taking the limit as M goes to infinity and as the probability mass of each $\mathbf{x}$ goes to zero will show that Eq. (7) is also unbiased and has smaller variance than Eq. (8) in the continuous case. We give proofs of these statements.

3.1.1 Proof that the GPAIS Estimator is Unbiased

Let $\mathcal{R}$ be the set of all rare event states such that

$$\sum_{\mathbf{x}\in\mathcal{R}} p(\mathbf{x}) = \sum_{x\in\mathcal{X}} I(\mathbf{x})\,p(\mathbf{x}) = P \ll 1 \quad \text{and} \quad I(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \mathcal{R} \\ 0 & \forall \mathbf{x} \notin \mathcal{R} \end{cases}.$$

Then

$$\begin{aligned}
\mathrm{E}[P_{\text{GPAIS}}] &= \mathrm{E}_{\{\mathbf{x}_i\sim p_i'\}}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{I(\mathbf{x}_i)\,p(\mathbf{x}_i)}{p^{\mathcal{E}_x}(\mathbf{x}_i)}\right] = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{E}_{\{\mathbf{x}_i\sim p_i'\}}\left[\frac{I(\mathbf{x}_i)\,p(\mathbf{x}_i)}{p^{\mathcal{E}_x}(\mathbf{x}_i)}\right]\right) \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{M} p_i'(\mathbf{x}_k)\frac{I(\mathbf{x}_k)\,p(\mathbf{x}_k)}{p^{\mathcal{E}_x}(\mathbf{x}_k)}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{\mathbf{x}_k\in\mathcal{R}} p_i'(\mathbf{x}_k)\frac{p(\mathbf{x}_k)}{p^{\mathcal{E}_x}(\mathbf{x}_k)}\right) \\
&= \sum_{\mathbf{x}_k\in\mathcal{R}}\frac{p(\mathbf{x}_k)}{p^{\mathcal{E}_x}(\mathbf{x}_k)}\left(\frac{1}{N}\sum_{i=1}^{N} p_i'(\mathbf{x}_k)\right) = \sum_{\mathbf{x}_k\in\mathcal{R}}\frac{p(\mathbf{x}_k)}{p^{\mathcal{E}_x}(\mathbf{x}_k)}\left(p^{\mathcal{E}_x}(\mathbf{x}_k)\right) \\
&= \sum_{\mathbf{x}_k\in\mathcal{R}} p(\mathbf{x}_k) = P.
\end{aligned} \tag{9}$$

Thus, assuming the $\mathbf{x}_i$s are independent, the GPAIS estimator is unbiased.

3.1.2 Proof that the GPAIS Estimator Reduces Variance

Lemma: If the $\mathbf{x}_i$s are independent, then

$$\mathrm{E}\left[\sum_{i=1}^{N}\left(\frac{I\left(\mathbf{x}_i\right)p\left(\mathbf{x}_i\right)}{N\,p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right)^2\right] \leq \mathrm{E}\left[\sum_{i=1}^{N}\left(\frac{I\left(\mathbf{x}_i\right)p\left(\mathbf{x}_i\right)}{N\,p'_i\left(\mathbf{x}_i\right)}\right)^2\right]. \tag{10}$$

Proof: The harmonic mean inequality states that a harmonic mean is less than or equal to the arithmetic mean. This implies that for all $\mathbf{x}_k$

$$\frac{1}{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{p'(\mathbf{x}_k)}} \leq \frac{1}{N}\sum_{i=1}^{N}p'_i\left(\mathbf{x}_k\right),$$

and hence that

$$\frac{1}{p^{\mathcal{E}_x}\left(\mathbf{x}_k\right)} \leq \frac{1}{N}\sum_{i=1}^{N}\frac{1}{p'_i\left(\mathbf{x}_k\right)}. \tag{11}$$

The equality holds only when all $p'_i\left(\mathbf{x}\right)$ are the same. It then follows that

$$\begin{aligned}
\mathrm{E}\left[\sum_{i=1}^{N}\left(\frac{I\left(\mathbf{x}_i\right)p\left(\mathbf{x}_i\right)}{N\,p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right)^2\right] &= \sum_{i=1}^{N}\mathrm{E}\left[\left(\frac{I\left(\mathbf{x}_i\right)p\left(\mathbf{x}_i\right)}{N\,p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right)^2\right] \\
&= \sum_{i=1}^{N}\left\{\sum_{k=1}^{M}p'_i\left(\mathbf{x}_k\right)\left(\frac{I\left(\mathbf{x}_k\right)p\left(\mathbf{x}_k\right)}{N\,p^{\mathcal{E}_x}\left(\mathbf{x}_k\right)}\right)^2\right\} \\
&= \sum_{k=1}^{M}\left(\frac{I\left(\mathbf{x}_k\right)p\left(\mathbf{x}_k\right)}{N\,p^{\mathcal{E}_x}\left(\mathbf{x}_k\right)}\right)^2\sum_{i=1}^{N}p'_i\left(\mathbf{x}_k\right) \\
&= \sum_{k=1}^{M}\left(\frac{I\left(\mathbf{x}_k\right)p\left(\mathbf{x}_k\right)}{N\,p^{\mathcal{E}_x}\left(\mathbf{x}_k\right)}\right)^2 N\,p^{\mathcal{E}_x}\left(\mathbf{x}_k\right) \\
&= \sum_{k=1}^{M}\frac{\left(I\left(\mathbf{x}_k\right)p\left(\mathbf{x}_k\right)\right)^2}{N}\left(\frac{1}{p^{\mathcal{E}_x}\left(\mathbf{x}_k\right)}\right) \\
&\leq \sum_{k=1}^{M}\frac{\left(I\left(\mathbf{x}_k\right)p\left(\mathbf{x}_k\right)\right)^2}{N}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{p'_i\left(\mathbf{x}_k\right)}\right) \quad [\text{by Eq. (11)}] \\
&= \sum_{i=1}^{N}\left\{\sum_{k=1}^{M}p'_i\left(\mathbf{x}_k\right)\left(\frac{I\left(\mathbf{x}_k\right)p\left(\mathbf{x}_k\right)}{N\,p'_i\left(\mathbf{x}_k\right)}\right)^2\right\} \\
&= \sum_{i=1}^{N}\mathrm{E}\left[\left(\frac{I\left(\mathbf{x}_i\right)p\left(\mathbf{x}_i\right)}{N\,p'_i\left(\mathbf{x}_i\right)}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{i=1}^{N}\left(\frac{I\left(\mathbf{x}_i\right)p\left(\mathbf{x}_i\right)}{N\,p'_i\left(\mathbf{x}_i\right)}\right)^2\right].
\end{aligned}$$

Theorem: If the $\mathbf{x}_i$s are independent, then

$$\mathrm{Var}\left[P_{\mathrm{GPAIS}}\right] \leq \mathrm{Var}\left[P_{\mathrm{Trad}}\right] \tag{12}$$

Proof: Define

$$\mu_i \equiv \mathrm{E}\left[\frac{1}{N}\,\frac{I\left(\mathbf{x}_i\right)p\left(\mathbf{x}_i\right)}{p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right].$$

Then Eq. (9) shows that

$$\sum_{i=1}^{N} \mu_i = P,$$

which (because variance cannot be negative) implies that

$$\sum_{i=1}^{N} \mu_i^2 \geq \frac{P^2}{N}. \tag{13}$$

Then

$$\begin{aligned}
\mathrm{Var}\left[P_{\mathrm{GPAIS}}\right] &= \mathrm{Var}\left[\sum_{i=1}^{N} \frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{N\, p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right] \\
&= \sum_{i=1}^{N} \mathrm{Var}\left[\frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{N\, p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right] \\
&= \sum_{i=1}^{N} \mathrm{E}\left[\left(\frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{N\, p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right)^2 - \mu_i^2\right] \\
&= \mathrm{E}\left[\sum_{i=1}^{N} \left(\frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{N\, p^{\mathcal{E}_x}\left(\mathbf{x}_i\right)}\right)^2\right] - \sum_{i=1}^{N} \mu_i^2 \\
&\leq \mathrm{E}\left[\sum_{i=1}^{N} \left(\frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{N\, p_i'\left(\mathbf{x}_i\right)}\right)^2\right] - \frac{P^2}{N} \quad [\text{ From Eq. (10) and Eq. (13)}] \\
&= \sum_{i=1}^{N} \mathrm{E}\left[\left(\frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{N\, p_i'\left(\mathbf{x}_i\right)} - \frac{P}{N}\right)^2\right] \\
&= \sum_{i=1}^{N} \mathrm{Var}\left[\frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{N\, p_i'\left(\mathbf{x}_i\right)}\right] \\
&= \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} \frac{I\left(\mathbf{x}_i\right) p\left(\mathbf{x}_i\right)}{p_i'\left(\mathbf{x}_i\right)}\right] \\
&= \mathrm{Var}\left[P_{\mathrm{Trad}}\right].
\end{aligned}$$

3.2 GPAIS Algorithm

We have shown that the GPAIS importance density will yield an unbiased estimator for a failure probability, and an estimate which has lower variance than a traditional importance sampling approach. This section provides details about the algorithm implementation, specifically how to generate the component densities in the ensemble. In practice, the first component importance density to approximate $p^*\left(\mathbf{x}\right)$ will be based on a GP emulator that is built from an set of multiple points drawn from an initial (zeroth) component distribution. We use GP models in this work because they have been shown to be effective emulators or surrogates for black-box functions [11] and they have the capability to estimate the error in their prediction.

The following summary of our GPAIS algorithm will hopefully provide the reader with context when we subsequently describe the components in greater detail.

- We generate an initial set of samples that is "optimal" for both constructing the initial GP and estimating the failure probability. In the absence of prior information about the unknown true function,

  - the nominal distribution is the optimal initial distribution for estimating the probability of failure, and
  - it is optimal to build GPs on a bounded domain from uniformly spaced data points.

  We therefore map the simulator's D input variables, $\mathbf{x}$, through their cumulative distribution functions (CDFs), to "χ space" where the domain of the GP is the $[0,1]^D$ unit hypercube. If the input variables are independent, then the mapped to χ space D-dimensional nominal distribution will be uniform and the one-dimensional marginals of the nominal distributions are guaranteed to be uniform regardless of their dependence/independence. We build the first (and all subsequent) GPs in χ space using an initial set of data points that are generated by LHS in χ space.

- The GP's spatially varying, real valued, expectation of the indicator function, hereafter referred to as the *expected indicator function*, is the portion of its Gaussian distribution in the "vertical" direction that is on the failing side of the failure threshold. The expected indicator function is used in place of the unknown true indicator function to approximate the unknown ideal importance density.

- Sampling the GP a large number of times is used to estimate the normalization constant of the approximated ideal importance density. This large number of emulator evaluations is reused to draw one or more importance samples from the approximated ideal importance density.

- The simulator is evaluated at the new importance samples which are added to the existing set of data points and used to rebuild the GP. This rebuilding of the GP as new data points are added adaptively updates/refines our approximation of the unknown true ideal importance density.

- At the end of the importance sampling, the GPAIS estimator is used to calculate the probability of failure.

Note that the preceding was a simplified explanation, and the actual implementation is far more detailed. For example, to take advantage of available computing resources, one might wish to concurrently evaluate the simulator at batches of points drawn from each of the component distributions. We now redefine the ensemble and ensemble distribution to facilitate the presentation of the GPAIS algorithm in these contexts.

Let $\mathcal{E}_{\mathbf{x}} = \{p'_j(\mathbf{x}) \ \forall j = 0,1,2,...,J\}$ be a set or "ensemble" of $J+1$ different distributions defined over the domain of $\mathbf{x}$ from which a total of N points are to be drawn so that n_j points are drawn from distribution $p'_j(\mathbf{x})$ and $N = \sum_{j=0}^{J} n_j$. Also let $\mathbf{x}_i$ for $i = 1,2,3,...,N$ be an arbitrary set of N points that are drawn from this ensemble of distributions. Then the *ensemble distribution* is

$$p^{\mathcal{E}_{\mathbf{x}}}(\mathbf{x}) = \frac{1}{N}\sum_{j=0}^{J} n_j p'_j(\mathbf{x}). \tag{14}$$

The definition of the GPAIS estimator in Eq. (7) is unchanged.

The adaptive part of our GPAIS algorithm is that the full ensemble of distributions, $\mathcal{E}_{\mathbf{x}}$, is not known *a priori*, instead the jth component distribution, $p'_j(\mathbf{x})\ j > 0$, is an estimate of the ideal importance distribution $p^*(\mathbf{x})$ found by fitting a GP to the subset of points already drawn from the $0,...,j-1$ component distributions. The GP's adjusted mean, $\mathrm{E}[y(\mathbf{x})]$, is its "best guess" for the function $y(\mathbf{x})$ whose true value is only known where it has been sampled.

Note that there may be ambiguity about the word "adaptive" in adaptive importance sampling. Zhang [8] refers to AIS as a technique for simultaneously estimating the parameters governing the importance sampling distribution and estimating the quantity of interest (the expectation) as compared to performing the computations in a staged approach. Another interpretation of AIS is that the importance sampling probability density changes or is updated iteratively during the course of estimating the expectation. We use the latter meaning.

Note that although the drawing of the different $\mathbf{x}_i$ from their respective $p'_i(\mathbf{x})$ is independent, the component *distributions* later in the sequence do depend on the earlier draws. Strictly speaking, this violates the assumption of independent $\mathbf{x}_i$ used to prove unbiasedness and reduced variance relative to the traditional estimator. Despite this, the GPAIS algorithm performed admirably on test problems.

If any of the one-dimensional marginal distributions of the D-dimensional nominal distribution, $p(\mathbf{x})$, is not the uniform distribution, then the original $\mathbf{x}$ should be mapped to χ through their one-dimensional cumulative distribution functions, $\chi_d = \mathrm{CDF}_d(x_d)\ \ d = 1, 2, ..., D$. The GP is then constructed to approximate $y(\chi)$ rather than $y(\mathbf{x})$. Here and in what follows, we use p to denote distributions defined over $\mathbf{x}$ and ρ to denote distributions defined over χ. We also abuse notation so that $y(\chi)$ and $I(\chi)$ denote mapping χ to $\mathbf{x}$ and then, respectively, evaluating the true function and indicator function at the corresponding $\mathbf{x}$. Similarly $\mathrm{E}[y(\chi)]$ and $\mathrm{Var}[y(\chi)]$, denote evaluating the emulator's adjusted mean and variance in terms of χ; however, since the emulator is natively defined in χ space, evaluating $\mathrm{E}[y(\chi)]$ and $\mathrm{Var}[y(\chi)]$ does not involve mapping χ to $\mathbf{x}$. This has several advantages.

GPs tend to be best conditioned and most accurate when they are built from uniformly spaced points on a bounded domain and the GP only needs to extrapolate a very short distance beyond the build points. Even if the marginals of $p(\mathbf{x})$ have infinite support (e.g. Gaussian distributions), $\rho(\chi)$ is defined over the bounded domain $[0, 1]^D$. If the D dimensions are independent, then the nominal probability density in χ space is $\rho(\chi) = \prod_{d=1}^{D} u(0, 1) = 1$. If the dimensions are not independent, then a copula will be needed to express that dependence. This is still somewhat convenient because copulas are defined in terms of χ (the marginal CDFs). However, handling dependent dimensions using a copula is beyond the scope of the current work. For the remainder of this paper we will assume that the dimensions are independent and therefore that $\rho(\chi) = 1$.

Sampling χ from $\rho(\chi)$ is equivalent to sampling $\mathbf{x}$ from $p(\mathbf{x})$. Because $\rho(\chi)$ is the D-dimensional uniform distribution we can select an initial set of samples that is "optimal" for both constructing the GP and estimating the failure probability. It also greatly simplifies our algorithm. Expressed in terms of χ rather than $\mathbf{x}$, P_{GPAIS} is

$$P_{\mathrm{GPAIS}} = \frac{1}{N} \sum_{i=1}^{N} \left(I(\chi_i) \frac{1}{\rho^{\mathcal{E}_\chi}(\chi_i)} \right) \quad \chi \sim \mathcal{E}_\chi, \tag{15}$$

where

$$\mathcal{E}_\chi = \{\rho'_j(\chi)\ \forall j = 0, 1, 2, ..., J\} \tag{16}$$

and

$$\rho^{\mathcal{E}_\chi}(\chi) = \frac{1}{N} \sum_{j=0}^{J} n_j \rho'_j(\chi). \tag{17}$$

The "Gaussian" in "Gaussian process" refers to the GP's estimated normal distribution of possible true surfaces about the adjusted mean. The portion of the Gaussian cumulative distribution function that exceeds the threshold, T, is the GP's expected indicator function, $\mathrm{E}[I(\chi)]$. If failure occurs when $y(\chi) > T$, then we define

$$\mathrm{E}[I(\chi)] = \frac{1}{2} \left(1 + \mathrm{erf} \left(\frac{(\mathrm{E}[y(\chi)] - T)}{\sqrt{2\mathrm{Var}[y(\chi)]}} \right) \right). \tag{18}$$

Alternatively, if failure occurs when $y(\chi) < T$, then

$$\mathrm{E}[I(\chi)] = \frac{1}{2} \left(1 + \mathrm{erf} \left(\frac{(T - \mathrm{E}[y(\chi)])}{\sqrt{2\mathrm{Var}[y(\chi)]}} \right) \right). \tag{19}$$

The expected indicator function is illustrated in Fig. 1, where it is shown as the solid red line at the bottom of the subplots. The explanation is given in the caption.

Note that the GPs are not used to *directly* calculate the probability of failure; the true indicator function is used for that purpose as indicated in Eq. (15). The GP is only used to approximate ρ^*, i.e., KDE's role in the approach of Swiler and West [10]. This avoids Limitation 1 and shifts Limitation 2 to the determination of $\rho'_j(\chi)$.

We use the expected indicator function to determine the component densities used in GPAIS. The jth GP's estimate, $j > 0$, of the ideal importance density is then

$$\rho'_j(\chi) = \alpha_j \mathrm{E}_j[I(\chi)]\, \rho(\chi) = \alpha_j \mathrm{E}_j[I(\chi)], \tag{20}$$

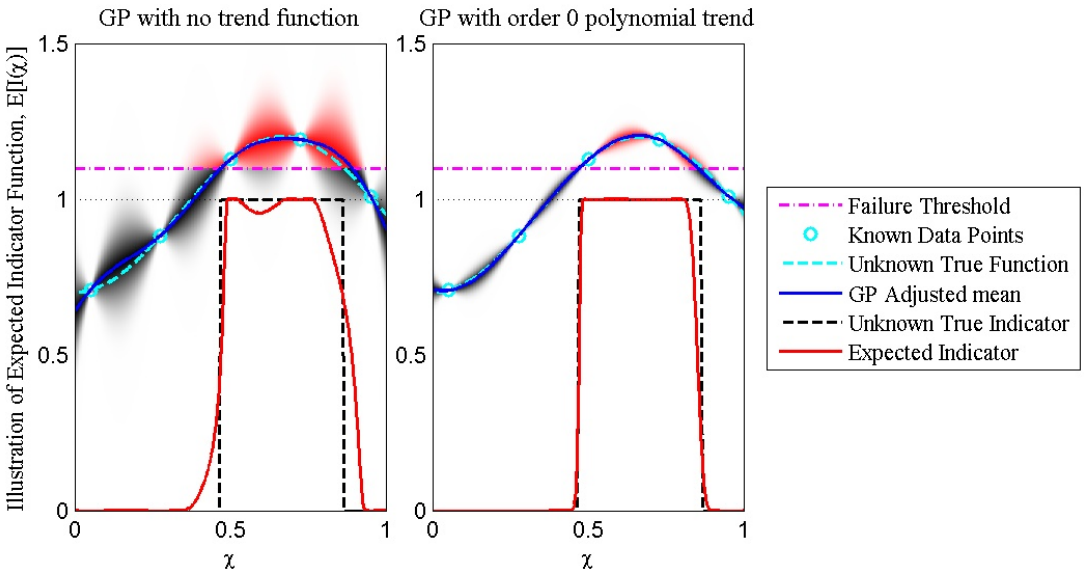

FIG. 1: Illustration of the expected indicator function, $\mathrm{E}\left[I\left(\chi\right)\right]$, which is shown as the solid red line at the bottom of the subplots. The dashed black line is the unknown true indicator function. The left subplot uses a deliberately bad trend function (i.e., no trend function) to illustrate the behavior of the GP's expected indicator function under large uncertainty. The right subplot uses an appropriate trend function to illustrate that the expected indicator function closely follows the unknown true indicator function. The GP was built from the 5 data points represented by the cyan circles. The unknown true function is plotted as the dashed cyan line. The GP's adjusted mean is plotted as the solid blue line. The GP's Gaussian distribution in the "vertical direction" is represented by the intensity of the red/black shaded region where the red portion signifies failure and the black portion signifies nonfailure. In both regions, the intensity of the shading signifies how much of the distribution is further from the adjusted mean than the current point. In the left subplot, the slight dip in the expected indicator function at about $\chi = 0.55$ is due to the estimated uncertainty about the unknown true function's value locally increasing faster than the distance of the adjusted mean above the failure threshold. In this figure, failure occurs above the threshold which is plotted as the dash-dotted magenta line.

and its normalization constant α_j is given by

$$\alpha_j = \frac{1}{\int \mathrm{E}_j\left[I\left(\chi\right)\right] \rho\left(\chi\right) d\chi} = \frac{1}{\int \mathrm{E}_j\left[I\left(\chi\right)\right] d\chi} = \frac{1}{\hat{P}_j}. \tag{21}$$

Note that $\mathrm{E}\left[I\left(\chi\right)\right]$ is real valued instead of binary. If one adds a nugget to handle ill-conditioning in the GP construction, the slight loss in accuracy of the expected indicator is inconsequential rather than critical. One could also add a nugget to account for either the threshold, T, or $y(\chi)$ being stochastic rather than deterministic.

GPAIS avoids or otherwise overcomes many of the traditional limitations of importance sampling approaches. However, it has two new challenges. The first is how to determine α_j; it is challenging because the integral in the denominator of Eq. (21) cannot be performed analytically. The second is how to draw from component importance densities $\rho_j'(\mathbf{x})$ that are only implicitly defined. We found a joint solution to both problems, namely to evaluate the jth GP emulator at a large number, K, of points ξ_k $k = 1, 2, ..., K$ where $\xi \sim \rho(\xi)$. From this ensemble of emulator evaluations we estimate α_j and draw n_j importance samples. The true function $y\left(\mathbf{x}\right)$ is then evaluated at these n_j points which are added to the set of points used to construct GP_{j+1}.

Note that α_j being approximate rather than exact does not mean that the samples are drawn from the wrong distribution since the distribution implicitly defined by the GP is the distribution that we are drawing from. Rather,

approximate α_js only mean that the ensemble distribution, $\rho^{\mathcal{E}_X}(\chi)$ used in Eq. (15) is approximate. And fortunately, the error in $\rho^{\mathcal{E}_X}(\chi)$ is fairly easy to control.

We accomplish the estimation of α_j by evaluating a quantity $f_k = \mathrm{E}_j\left[I\left(\xi_k\right)\right]\rho\left(\xi_k\right) = \mathrm{E}_j\left[I\left(\xi_k\right)\right]$ for each ξ_k where $\xi \sim \prod_{d=1}^{D} u(0,1)$. Equation (2) shows that the Monte Carlo (MC) standard deviation of error for probability of failure is roughly $P_{\mathrm{F}}/\sqrt{N_{\mathrm{F}}}$, so if we continue evaluating the emulator until $S_j = \sum_{k=1}^{K} f_k \geq 25$, then the standard deviation of integration error in $\hat{P}_j \approx S_j/K$ should be at most about $\hat{P}_j/5$. At first glance an error of 20% in $\hat{P}_j$ might seem unacceptable, but if 25 component GP approximations of $\rho^*\left(\chi\right)$ are employed, then the overall integration error in $\rho^{\mathcal{E}_X}\left(\chi\right)$ should be about 4%, which is often acceptable. Likewise, using 100 component GP approximations should result in 2% or less integration error and having n_j = 1 for $j > 0$ minimizes the integration error. However, if evaluating $y\left(\xi\right)$ is sufficiently "expensive," then it will be preferable to use a larger S_j; e.g., if we continue sampling the emulator until $S_j = \sum_{k=1}^{K} f_k \geq 400$ then the integration error in $\hat{P}_j$ will be about $\hat{P}_j/20$.

We then select the kth evaluation, ξ_k, of the jth GP to be the ith importance sample, χ_i, with probability $P\left(\chi_i = \xi_k\right) = f_k/S_j$. This can be done by normalizing the cumulative sum of the f_k terms to 1 and randomly drawing one real number from the uniform distribution over $[0,1]$. That draw selects one of the f_k terms and its associated ξ_k. If $S_j = 0$ after a very large K, one can simply draw χ_i from $\rho(\chi)$, which is equivalent to all of the emulator evaluations having equal probability of being selected and thus is equivalent to drawing $\mathbf{x}$ from the nominal distribution $p\left(\mathbf{x}\right)$. If the S_j is very small but nonzero when the upper limit on K is reached, then selecting the point via the cumulative sum will still generate "good" (i.e., more likely to fail) samples, but the previously stated probabilistic bound on the overall integration error may not apply.

Drawing batches of $n_j > 1$ importance samples is more complicated, but is not conceptually more difficult than selecting a single importance sample. One just needs to ensure that individual ξ_ks are not selected more than once without altering the probability of any bin being selected. This can be accomplished by

- using a one-dimensional (1D) LHS to select multiple samples from the cumulative sum of f_ks,
- reordering the f_ks so that smallest values are located closest to the LHS bin edges,
- requiring a significantly larger S_k, and
- when an individual f_k is selected more than once, discarding and redrawing the whole 1D LHS.

If S_k is small but nonzero when the upper limit on K is reached, then a portion, perhaps all but one sample, could be drawn from $\rho(\chi)$. Drawing batches of points also means there are fewer component importance densities in the mixture approximation. For these reasons, batch GPAIS can perform noticeably worse than when points are drawn one at a time.

Our GPAIS algorithm actually builds two GPs at each step; the hyper-parameters for both are selected by using global optimization to maximize the likelihood. The first candidate uses the exponential correlation function,

$$r\left(\chi_1, \chi_2\right) = \exp\left(-\sum_{d=1}^{D} \theta_d \left|\chi_{1,d} - \chi_{2,d}\right|\right), \tag{22}$$

which produces C^0 continuous GPs. The exponential correlation function is useful for capturing pathological problems with discontinuities because it allows for greater localization of uncertainties. The second candidate uses the squared exponential (also known as "Gaussian") correlation function,

$$r\left(\chi_1, \chi_2\right) = \exp\left(-\sum_{d=1}^{D} \theta_d \left(\chi_{1,d} - \chi_{2,d}\right)^2\right), \tag{23}$$

which produces C^∞ continuous GPs. The squared exponential correlation function is useful for smooth problems where it has smaller uncertainty at interpolated points.

Our algorithm calculates $\hat{P}_j$ for both GPs using the same 10^4 samples of ξ_k, selects the GP with the smaller $\hat{P}_j$ and, if necessary (i.e., if S_j is smaller than desired), generates additional ξ_k for the selected GP to refine its estimate of $\hat{P}_j$. This heuristic method of GP selection is based on the assumption that the true failure probability is "small" and $\hat{P}_j$ for the selected GP is smaller because it has less uncertainty about, and therefore is more representative of, the unknown true function. This assumption deserves a bit of explanation.

Relative to other smart sampling techniques such as space-filling LHS [12–14], importance sampling is only beneficial if the probability of failure is very small (or very close to 1, in which case it can accurately estimate the very small probability of not-failing and that can be subtracted from 1). Thus, the assumption that the probability of failure is "small" is likely valid given that importance sampling is being applied. Suppose that

- there are two GP emulators designated as "A" and "B,"
- these two emulators predict identical adjusted means which are fairly close to the true function,
- emulator A is correctly confident (has small adjusted variance) about where the true function is far from the failure threshold, although it is not confident about if or where it does fail, and
- emulator B is unconfident (has large variance) about where it correctly predicts the true function.

Then

- GP A will estimate a small $\hat{P}$ and accurately approximate $\rho^*(\chi)$, meaning it correctly favors regions of the domain where the failure occurs or is close to occurring; and
- GP B will estimate a much larger $\hat{P}$ and poorly approximate $\rho^*(\chi)$, meaning it will be closer to $\rho(\chi)$ (i.e., cause GPAIS to be closer to conventional MC) than GP A.

Moreover, GPs with small variance tend to have adjusted means closer to truth than those with large variance.

Note that the exponential and squared exponential correlation functions are the respective lower and upper bound assumptions on the true function's degree of smoothness. Taken in combination, they provide good coverage for functions with intermediate degrees of smoothness.

To summarize, GPAIS has the following steps:

- Take an initial set of LHS samples from $\rho_0'(\chi) = \prod_{d=1}^{D} u(0,1)$
- For $j = 1, 2, ..., J$
  1. Build two candidates (over χ) for GP_j
  2. Evaluate both candidate GPs at 10^4 random samples and estimate $\hat{P}_j$ from Eq. (21)
  3. Select the GP with the smaller $\hat{P}_j$ as GP_j
  4. If necessary (i.e., if S_j is smaller than desired), evaluate GP_j at additional random samples to refine the estimate of $\hat{P}_j$
  5. Generate n_j (typically 1) draws of χ from $\rho_j'(\chi)$ and evaluate the true function $y(\chi)$ at these draws
  6. Add the new sample points to the set of build points for GP_{j+1}
- Use Eq. (15) to calculate the failure probability, where the importance density is calculated by Eq. (17).

4. RESULTS

We present the results of the GPAIS algorithm on four test problems with a range of attributes. Thresholds were chosen to produce desired failure probabilities and are listed in Table 1.

The Herbie test function [15] is

$$y_{\text{herbie}}(\mathbf{x}) = -\prod_{d=1}^{D}\left(\exp\left(-(x_d-1)^2\right)+\exp\left(-0.8(x_d+1)^2\right)-0.05\sin\left(8(x_d+0.1)\right)\right). \tag{24}$$

When $P_{\text{F}} \approx 1.5\times10^{-2}$, the 2D Herbie function has five disjoint failure regions in the $[-2,2]^2$ square. Row 1 of Table 1 and the left subplot of Fig. 2 show the results for this case. With 50 training points and 150 points adaptively chosen one at a time, GPAIS estimated P_{F} to be 1.460×10^{-2}; discovered all five failure regions, and placed a significant number of points in the neighborhood of $x_1 = x_2 = 1$ which is close to failing (see the left subplot of Fig. 2).

For the Herbie function, when estimating the probability of failure for the first threshold (row 1 of Table 1), GPAIS chose the the exponential correlation function for the first 34 adaptively chosen points and the squared exponential for the remaining 116. When $P_{\text{F}} \approx 10^{-4}$, there is one failure region in the $[-2,2]^2$ square. The results for this case are shown in row 2 of Table 1 and the right subplot of Fig. 2. GPAIS put 128 of the 150 adaptively chosen points in the failure region and a significant number in three other regions that are close to failing. Note that in this case, the squared exponential function was always chosen.

For the ten-dimensional circular parabola test function,

$$y_{\text{circparab}}(\mathbf{x}) = \sum_{d=1}^{D}(2x_d-1)^2, \tag{25}$$

the GPAIS algorithm performs much better when allowed to choose between the exponential and squared exponential correlation function vs. being forced to use the exponential correlation function in the GP, both in terms of number of hits and accuracy of failure estimate (see rows 3 and 4 of Table 1). The results indicate that only 82 of the 300 adaptively chosen points using the exponential correlation function hit the failure region. However, when the algorithm was allowed to choose, it selected the exponential correlation function for the first adaptively chosen point and the squared exponential for the remaining 299. In this case, 282 of these points hit the failure region. This improvement is because this is a smooth function and well approximated with the squared exponential.

We have developed a particularly challenging, arbitrary dimensional test problem defined over the unit hypercube. We call it the "planar cross" function due to the shape of its failure region, which is shown in the left subplot of Fig. 3 for $T = 0.02$ and 3 dimensions.

TABLE 1: The "Exp GPs" and "Exp^2 GPs" columns contain the number of times the GP using the exponential and squared exponential correlation function, respectively, was selected. A "—" entry means GPAIS was forced to use the other correlation function for all adaptively chosen samples. A zero means GPAIS chose not to use the correlation function. The threshold T is approximate for the circular parabola and circular square root of diameter problems and exact for Herbie and planar cross. The true P_{F} is exact for the circular parabola and circular square root of diameter problems and approximate for Herbie and planar cross

Row ID	Test problem	T	$P(y<T)$	P_{GPAIS}	Training samples	Adaptive Samples	Exp GPs	Exp^2 GPs	No. of hits
1	2D Herbie	–1.065	1.5×10^{-2}	1.460×10^{-2}	50	150	34	116	120
2	2D Herbie	–1.12565	10^{-4}	1.079×10^{-4}	50	150	0	150	128
3	10D Circ. Parab.	0.5257	10^{-4}	1.004×10^{-4}	100	300	1	299	282
4	10D Circ. Parab.	0.5257	10^{-4}	9.50×10^{-5}	100	300	300	—	83
5	6D Planar Cross	0.003262	10^{-4}	8.82×10^{-5}	100	650	650	0	7
6	6D Planar Cross	0.003262	10^{-4}	5.63×10^{-5}	100	1500	—	1500	2
7	10D Circ. Diam. Root	0.8515	10^{-4}	9.36×10^{-5}	100	300	82	218	189

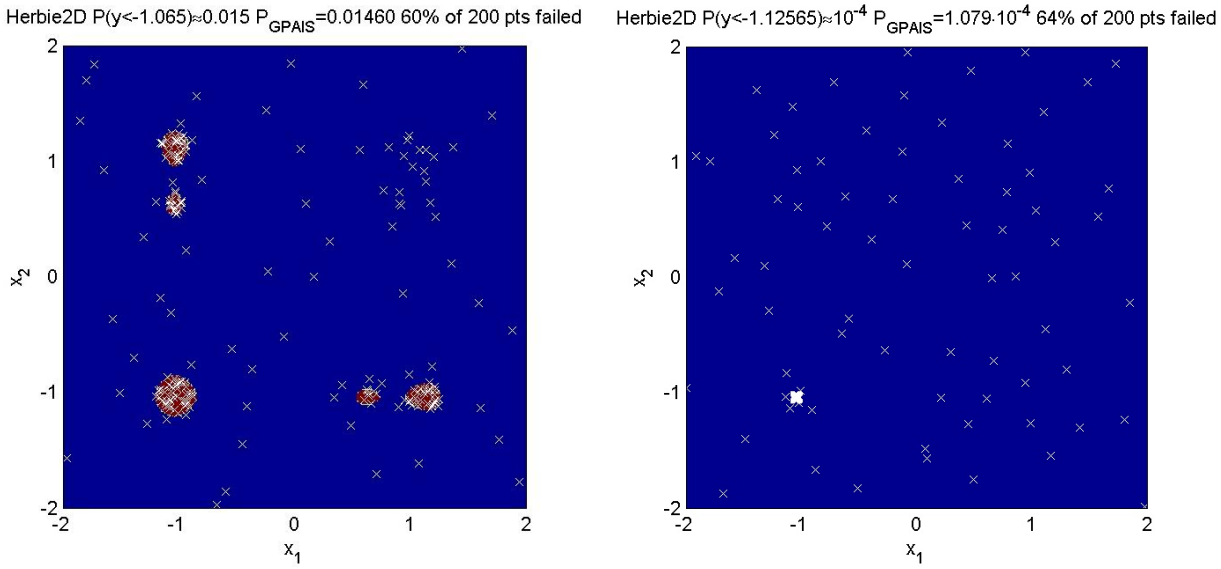

FIG. 2: For the 2D Herbie test function (left) with failure probability, $P_{\mathrm{F}} \approx 1.5\%$, the GPAIS estimate of P_{F} is 1.460%; (right) with a failure probability, $P_{\mathrm{F}} \approx 10^{-4}$, the GPAIS estimate of P_{F} is 1.079×10^{-4}. The red area indicates the failure region. Importance samples are plotted as white x's.

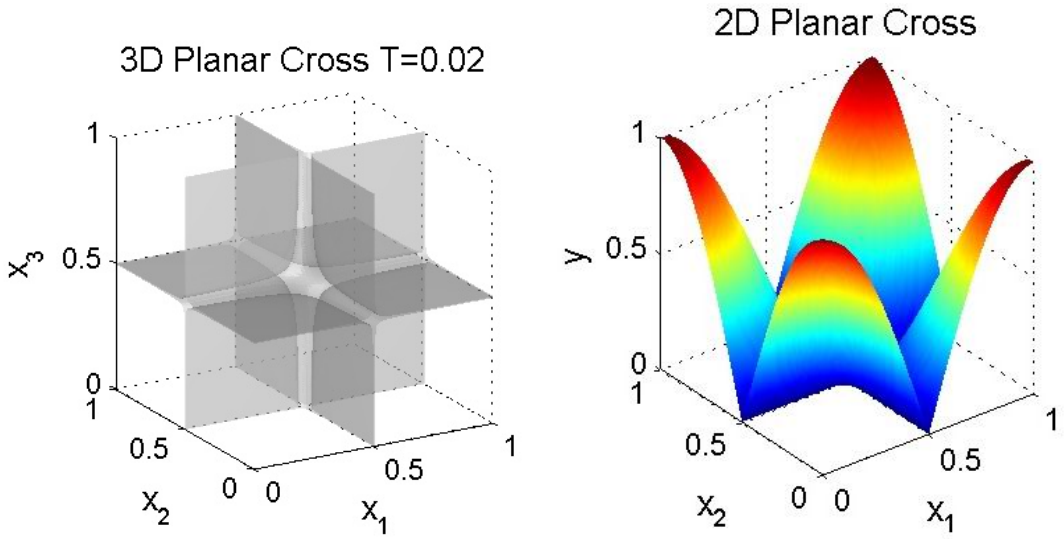

FIG. 3: Left: Isosurface of the failure region of the 3D planar cross function for $T = 0.02$. Right: The planer cross function for 2 inputs.

$$y_{\mathrm{planarcross}}(\mathbf{x}) = \left(\prod_{d=1}^{D} \left(\frac{1}{2} \left(1 + \cos\left(2\pi x_d \right) \right) \right) \right)^{1/D} \tag{26}$$

The 2D planar cross function is shown in the right subplot of Fig. 3. For $D = 1$, the planar cross function is C^{∞} continuous. For $D = 2$, the planar cross function has a finite magnitude discontinuity in its first derivative at $x_d = 0.5$ for any d. For $D \geq 3$, the first derivative discontinuity has infinite magnitude; this is what makes the planar cross function a pathologically challenging problem.

This test problem deserves a few comments. When GPAIS was allowed to use the exponential correlation function (row 5 of Table 1), the final $\hat{P}_j$ was 3.28×10^{-3}. This differs from the true P_{F} by roughly 3128%. From this it is apparent that the final GP was not a very good representation of reality. But, because the GP was only used to approximate the ideal importance density and the true indicator function was used to estimate P_{F}, the GPAIS estimate of 8.82×10^{-5} differs from truth by less than 12%. Also, GPAIS correctly determined that the C^0 continuous exponential correlation function was a much better choice than the C^∞ continuous squared exponential correlation function (row 6 of Table 1). For the sake of comparison, when GPAIS was required to use the squared exponential correlation function, with 100 training points, about 1100 adaptively chosen samples were required to hit the failure region. This is still substantially better than MC.

We present a test problem which highlights the usefulness of choosing the correlation function. We call this test function the circular square root of diameter function,

$$y_{\mathrm{circdiamroot}}(\mathbf{x}) = \sum_{d=1}^{D} ((2x_d - 1)^2)^{0.5/R}, \tag{27}$$

where we used $D = 10$ and $R = 2$. The function in two dimensional is shown in the right subplot of Fig. 4. This function is more difficult for emulators to predict than some functions like the circular parabola function (left subplot of Fig. 4) because it is less smooth.

The GPAIS algorithm performed better on the circular square root of diameter problem when it was allowed to choose between the exponential and squared exponential correlation functions vs. being forced to use only one correlation function for the GP. Row 7 of Table 1 and the left subplot of Fig. 5 show the GPAIS algorithm's choice of correlation function for this problem. The x axis displays the total number (training plus adaptive) samples. The blue line indicates the number of times the exponential correlation function is chosen. The green line shows the number of times the squared exponential correlation function is chosen. The cyan line is the total number of adaptive samples. The red line is the number of points chosen adaptively that hit the failure region. The left subplot of Fig. 5 shows that the GPAIS algorithm tended to prefer the exponential correlation function in the first 50 adaptive points. After that it tended to prefer the squared exponential correlation function. By the 150th adaptive point (the 250th simulation sample

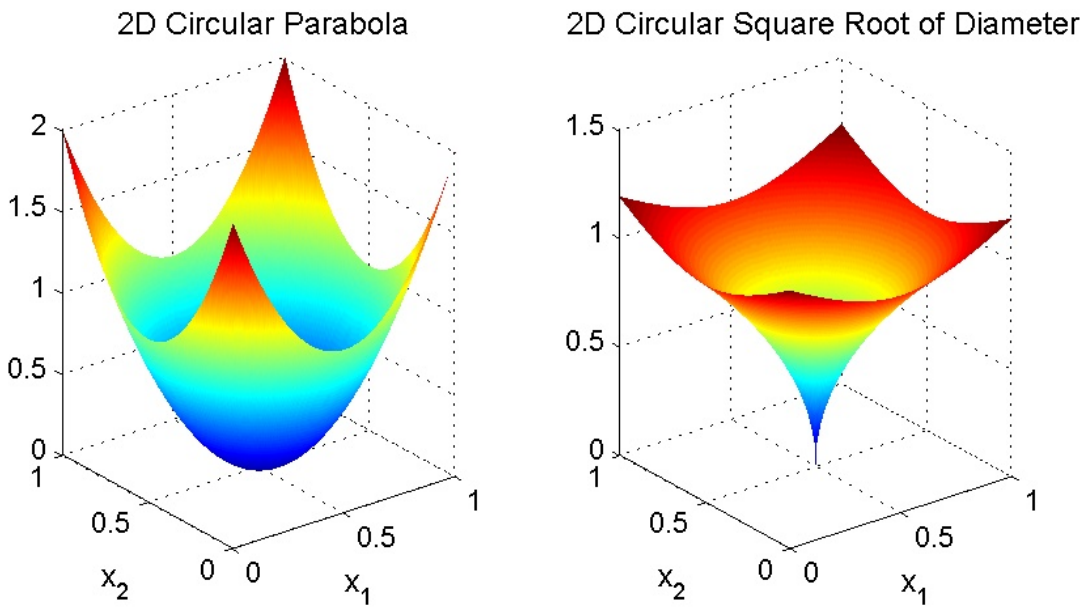

FIG. 4: Left: The circular parabola test function in two-dimensional. Right: The circular square root of diameter test function in two-dimensional.

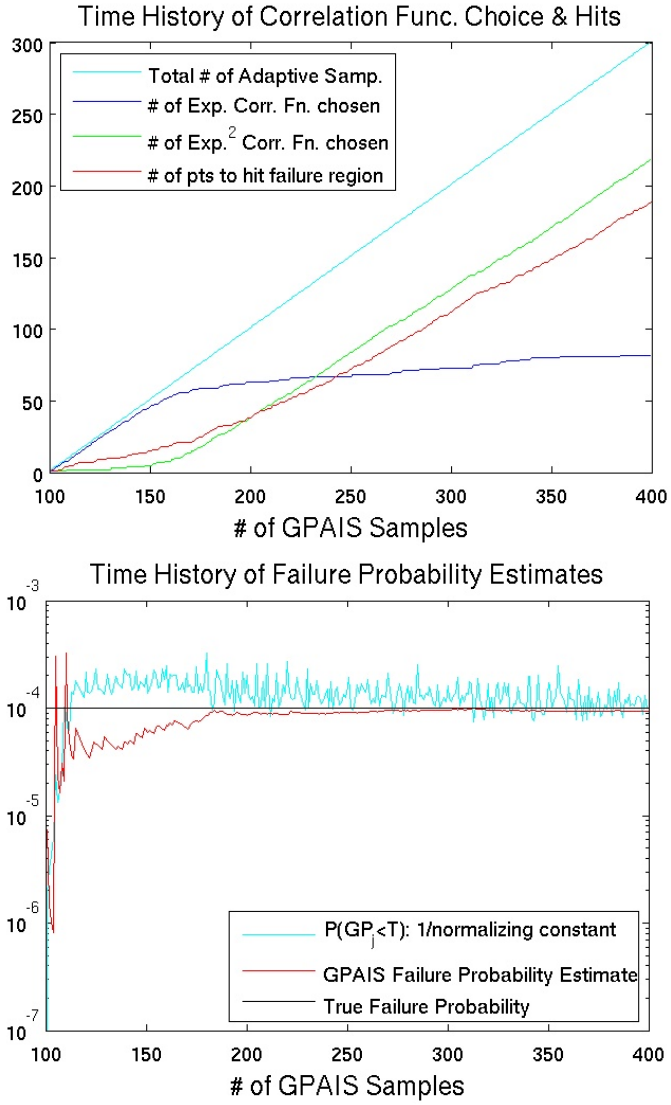

FIG. 5: GPAIS time history for the circular square root of diameter test problem. Top: Choice of correlation function. Bottom: Convergence of probability of failure estimate.

counting the 100 initial samples), the lines cross and the squared exponential becomes the predominant correlation function.

Our GPAIS algorithm is successful at adaptively selecting both the points and correlation function. Although none of the original 100 LHS samples fell in the failure region, GPAIS hit the failure region with 189 points identified as part of the construction of the importance density. That GPAIS produces many samples that "hit" the failure region is an independently useful benefit that should not be overlooked. One can take the samples that hit the failure region and analyze them for particular characteristics (e.g. do the failure region points tend to fall in certain parts of the input parameter space, is there correlation between the input parameters for points in the failure region, etc.). Thus, in addition to providing estimates of failure probability, GPAIS provides a mechanism for quickly generating points in the failure region for problems where MC sampling would take excessively long to generate a comparable number of points.

The right subplot of Fig. 5 shows how the GPAIS estimate converges for the circular square root of diameter problem. The red line shows the GPAIS estimate for failure probability. The cyan line shows the individual $\hat{P}_j$ estimates for each iteration, which is the reciprocal of the normalizing constant given in Eq. (21). The black line is the true probability of failure. The GPAIS failure estimate converges to the true estimate around adaptive sample 150, or by the time 250 samples in total have been evaluated.

We plot relative error based on all of these test problems (e.g., all seven rows of Table 1) as a function of number of hits, N_{F}, in Fig. 6. In the limit of an infinite number of samples, GPAIS should acquire perfect knowledge of the black-box function's response over the input domain, and all further samples would then hit the failure region. Thus plotting the relative error vs. the number of hits instead of the number of samples is a crude way of estimating asymptotic convergence properties for AIS methods from early time data. Note that this is simply an empirical study with no proof of convergence of the behavior of GPAIS. However, the slope of the line is around –0.74 in log scale, which is significantly better than the MC convergence rate of –0.5 as indicated in Section 1. Also note that this does not take into account the improved rate at which samples hit the failure region. For the pathological planar cross problem, row 5 of Table 1, which was the test problem for which our GPAIS algorithm performed the worst, GPAIS hit the failure region about 93 times more often than MC and this rate was accelerating.

5. CONCLUSIONS

We present a gaussian process adaptive importance sampling (GPAIS) algorithm that assumes the system is a black-box simulator and the user has no *a priori* knowledge about the location of important regions within the input space. It achieves the same accuracy as Monte Carlo or Latin hypercube sampling with orders of magnitude fewer function evaluations. It works well for small failure probabilities, small numbers of samples, disjoint failure regions, and a wide array of test problems. It performs surprisingly well on a pathological, six-dimensional, C^0 continuous test problem with infinite magnitude discontinuity in the first derivative and a small failure probability. The robustness of our GPAIS algorithm elevates importance sampling from an art form limited to experts to a practical tool that is beneficial to simulation users who wish to compute failure probabilities with their codes.

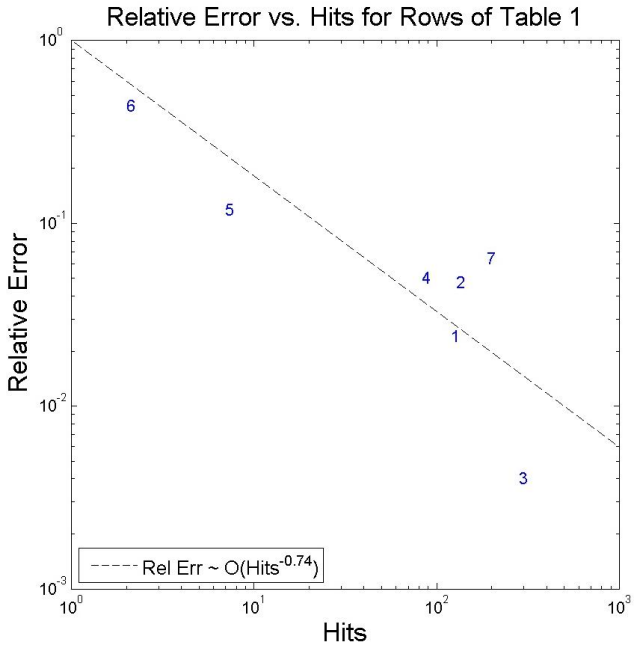

FIG. 6: Relative error estimates for the seven test function examples used in this paper.

ACKNOWLEDGMENTS

This work was sponsored by the Nuclear Energy Advanced Modeling and Simulation (NEAMS) program in the Advanced Modeling and Simulation Office in the Nuclear Energy Division in the US Department of Energy. The authors are grateful to Dr. Brian Williams and Dr. Rick Picard at Los Alamos National Laboratory for useful technical discussions. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

1. Stein, M., Large sample properties of simulations using latin hypercube sampling, *Technometrics*, 29(2):143–151, 1987.
2. Owen, A., A central limit theorem for latin hypercube sampling, *J. Stat. Soc. Ser. B (Methodological)*, 54(2):541–551, 1992.
3. Srinivasan, R., *Importance Sampling*, Springer-Verlag, Berlin, 2002.
4. Denny, M., Introduction to importance sampling in rare-event simulations, *Eur. J. Phys.*, 22(4):401–411, 2001.
5. Richard, J.-F. and Zhang, W., Efficient high-dimensional importance sampling, *J. Econometrics*, 141(2):1385–1411, 2007.
6. Oh, M. S. and Berger, J. O., Integration of multimodal functions by monte carlo importance sampling, *J. Am. Stat. Assoc.*, 88(422):450–456, 1993.
7. Givens, G. H. and Raftery, A. E., Local adaptive importance sampling for multi-variate densities with strong nonlinear relationships, *J. Am. Stat. Assoc.*, 91(433):132–141, 1996.
8. Zhang, P., Nonparametric importance sampling, *J. Am. Stat. Assoc.*, 91(435):1245–1253, 1996.
9. Owen, A. and Zhou, Y., Safe and effective importance sampling, *J. Am. Stat. Assoc.*, 95(449):135–143, 2000.
10. Swiler, L. and West, N., Importance sampling: Promises and limitations, in *Proc. of the 12th AIAA Non-Deterministic Approaches Conf.*, pp. 135–143, Sept. 10–12, Victoria, Canada, 2010.
11. Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., Design and analysis of computer experiments, *Stat. Sci.*, 4(4):409–435, 1989.
12. Cioppa, T. and Lucas, T., Efficient nearly orthogonal and space-filling latin hypercubes, *Technometrics*, 49(1):45–55, 2007.
13. Dalbey, K. and Karystinos, G., Fast generation of space-filling latin hypercube sample designs, in *Proc. of the 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2010–9085, September 2010.
14. Dalbey, K. and Karystinos, G., Generating a maximally spaced set of bins to fill for high dimensional space-filling latin hypercube sampling, *Int. J. Uncertainty Quantification*, 1(3):241–255, 2011.
15. Lee, H., Gramacy, R., Linkletter, C., and Gray, G., Optimization subject to hidden constraints via statistical emulation, *Pac. J. Opt.*, 7:467–478, 2011.