Information Reuse for Importance Sampling in Reliability-Based Design Optimization

Anirban Chaudhuri[1], Boris Kramer[2]
Massachusetts Institute of Technology, Cambridge, MA, 02139, USA
Karen E. Willcox[3]
University of Texas at Austin, Austin, TX, 78712, USA

Abstract

This paper introduces a new approach for importance-sampling-based reliability-based design optimization (RBDO) that reuses information from past optimization iterations to reduce computational effort. RBDO is a two-loop process—an uncertainty quantification loop embedded within an optimization loop—that can be computationally prohibitive due to the numerous evaluations of expensive high-fidelity models to estimate the probability of failure in each optimization iteration. In this work, we use the existing information from past optimization iterations to create efficient biasing densities for importance sampling estimates of probability of failure. The method involves two levels of information reuse: (1) reusing the current batch of samples to construct an *a posteriori* biasing density with optimal parameters, and (2) reusing the *a posteriori* biasing densities of the designs visited in past optimization iterations to construct the biasing density for the current design. We demonstrate for the RBDO of a benchmark speed reducer problem and a combustion engine problem that the proposed method leads to computational savings in the range of 51% to 76%, compared to building biasing densities with no reuse in each iteration.

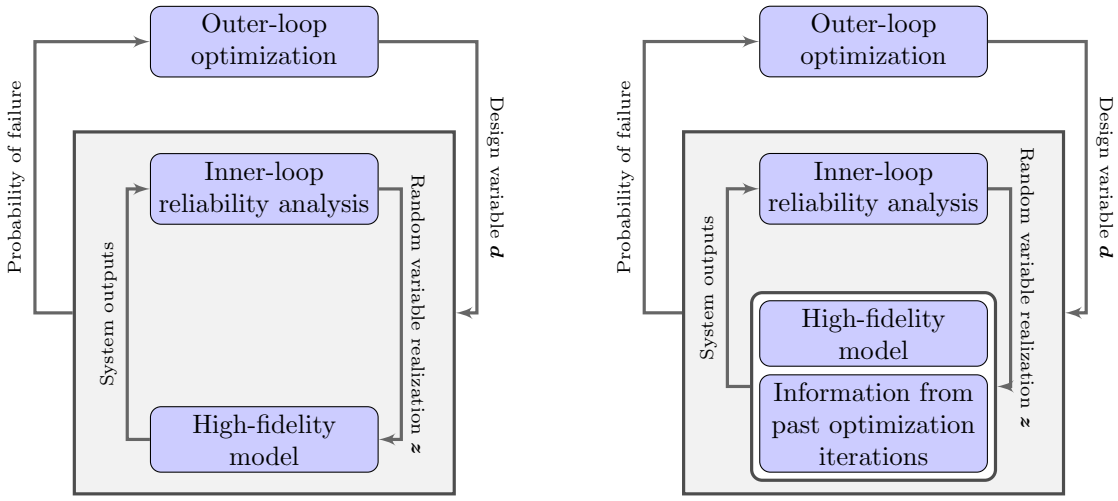
Keywords: Information reuse, importance sampling, biasing density, probability of failure, reliability analysis, optimization under uncertainty, reliability-based optimization, RBDO.

1. Introduction

Designing efficient and robust engineering systems requires dealing with expensive computational models while taking into account uncertainties in parameters and surrounding conditions. Reliability-based design optimization (RBDO) is a framework to minimize a prescribed cost function while simultaneously ensuring that the design is reliable (i.e., has a small probability of failure). RBDO is a two-loop process involving an outer-loop optimization with an inner-loop reliability analysis for each optimization iteration as shown in Figure 1(a). The reliability analysis requires estimating a probability of failure. RBDO approaches include: fully-coupled two-loop methods that evaluate the reliability at each optimization iteration, single-loop methods that introduce optimality criteria for an approximation of the reliability estimate [1, 2, 3, 4], and decoupled approaches

[1]Postdoctoral Associate, Department of Aeronautics and Astronautics, anirbanc@mit.edu.
[2]Postdoctoral Associate, Department of Aeronautics and Astronautics, bokramer@mit.edu.
[3]Director, Oden Institute for Computational Engineering and Sciences, kwillcox@oden.utexas.edu

that transform RBDO into a series of deterministic optimization problems with corrections [5, 6]. Surveys of existing RBDO methods can be found in Refs. [7, 8, 9] In this work, we concentrate on a two-loop RBDO method. For mildly nonlinear systems, reliability can be estimated using first-order and second-order reliability methods [10, 11]. However, strongly nonlinear systems typically require Monte Carlo sampling. The use of Monte Carlo methods is also more appropriate for systems with multiple failure regions, which first-order and second-order reliability methods cannot handle. The high cost of Monte Carlo sampling renders the RBDO problem computationally prohibitive in the presence of expensive-to-evaluate models. Thus, efficient methods are needed for evaluating the reliability constraint in each RBDO iteration for nonlinear systems.



(a) Using high-fidelity model

(b) Using IRIS

Figure 1: Two-loop process for RBDO using (a) the high-fidelity model, and (b) the proposed information reuse method.

One way to reduce the computational cost for RBDO is by using cheap-to-evaluate surrogate evaluations to replace the expensive high-fidelity evaluations in the Monte Carlo estimation of the probability of failure. Several methods use surrogates that have been adaptively refined around the failure boundary. Dubourg et al. [12] proposed refining kriging surrogates using a population-based adaptive sampling technique through subset simulation for RBDO. Bichon et al. [13, 14] combined adaptive Gaussian-process-based global reliability analysis with efficient global optimization (a.k.a. Bayesian optimization) for RBDO. Qu and Haftka [15] presented a method to solve the RBDO problem by building surrogates for probability sufficiency factor, which defines a probability of failure on the safety factor instead of the limit state function directly. Recent work has developed a quantile-based RBDO method using adaptive kriging surrogates [16], which uses the quantile (a.k.a. value-at-risk) instead of the probability of failure constraint. A comprehensive review on the use of surrogates in RBDO can be found in [17]. We note that sampling directly from surrogate models—while being computationally cheaper—introduces a bias, while we propose a method that is unbiased. However, using surrogate models offer substantial computational savings [17], and it would be possible to extend the proposed method to incorporate surrogates. Also, these methods

do not reuse information from previous design iterations, which is a source of computational savings that we explore in this work.

Another approach to reduce computational cost is to use importance sampling for the reliability analysis, which can allow for a drastic reduction in samples needed to estimate the reliability (or failure probability). Importance sampling is, in general, an efficient method for reliable systems and can be used to estimate small failure probabilities [18, 19] or other measures of risk [20, 21, 22, 23]. Although importance-sampling-based approaches can increase the efficiency of probability of failure estimation, they could still require many samples for the estimate if the biasing density is not constructed appropriately. Various efficient adaptive importance sampling methods [19] that iteratively get close to the optimal biasing density have been devised to meet this challenge. The cross-entropy method is a popular adaptive importance sampling method that uses the Kullback-Leibler (KL) divergence to iteratively get closer to the optimal biasing density [24, 25, 26]. Adaptive importance sampling has also been implemented using mixture models to account for multiple failure regions [27, 28]. Subset simulation is another method for efficiently estimating failure probabilities by converting a small failure probability into a series of larger conditional failure probabilities [29, 30, 31]. Surrogate-based importance sampling methods have also been proposed to further improve computational efficiency [32, 33, 34, 35, 36]. In this work, we develop an importance sampling method that builds good biasing densities using the optimization setup in RBDO.

We propose a new importance-sampling-based RBDO method that reuses information from past optimization iterations for computationally efficient evaluation of the reliability constraint as illustrated in Figure 1(b). At the core of the **IRIS-RBDO** (**I**nformation **R**euse for **I**mportance **S**ampling in **RBDO**) method, we propose to build a good importance sampling biasing density by reusing data from previous optimization iterations. The proposed method reduces the computational time for probability of failure estimates in each RBDO iteration through two levels of information reuse:

1. At the current design iteration, once the reliability estimate is computed via importance-sampling based Monte Carlo sampling, we reuse the current batch of samples from the reliability estimate to form an *a posteriori* biasing density. The *a posteriori* biasing density is built in an optimal way by minimizing the KL divergence measure to the optimal (zero-variance) biasing density.
2. At the next design iteration, we reuse the available *a posteriori* biasing densities from nearby designs explored in the past iterations to construct a mixture density at the current iteration. The motivation for this is that nearby designs are likely to have similar failure regions, and hence reusing the *a posteriori* biasing densities from the existing nearby designs can lead to efficient biasing densities.

In our IRIS-RBDO framework, the information from past optimization iterations acts as a surrogate for building biasing densities in each RBDO iteration. The optimization history is a rich source of information. Reusing information from past optimization iterations in optimization under uncertainty has been previously done in the context of robust optimization using control variates [37, 38, 39]. Cook et al. [40] extended the control variates method for information reuse to a larger class of estimators for robust optimization. The probabilistic re-analysis method, which has been used for RBDO [41] and probability of failure sensitivity analysis under mixed uncertainty [42], can be seen as a reuse method. The probabilistic re-analysis method creates an offline library with a large number of samples using a density encompassing the entire design and random variable space, and then uses importance sampling to re-weight the samples according to a specific density in each

RBDO iteration [43]. However, RBDO using probabilistic re-analysis is dependent on restrictive assumptions on the structure of the design and random variables, and does not guarantee the accuracy of probability of failure estimates in each RBDO iteration since it is sensitive to the quality of existing offline samples. Beaurepaire et al. [44] proposed a method for reusing existing information in the context of RBDO through bridge importance sampling, which is an adaptive sampling scheme that uses Markov Chain Monte Carlo to sample from intermediate densities. The initial density for the bridge importance sampling at the current design is constructed using the existing density from previous optimization iterations whose mode leads to the limit state function at the current design to be closest to the failure boundary while being within the failure threshold. This requires multiple new evaluations of the limit state function for implementing the reuse. While similar in purpose, our method is fundamentally different from the existing work of Beaurepaire et al. [44] in the way information from past optimization iterations is reused through a two-step process. We directly connect to variance reduction for probability of failure estimates by building the *a posteriori* biasing densities (leading to optimal biasing densities for the existing designs). Note that no new evaluations of the expensive limit state function are required to implement our information reuse. We next outline several important advantages of our method.

The key contribution of this paper is a new approach for reusing information from past optimization iterations in the context of importance-sampling-based RBDO. There are several advantages of the proposed IRIS-RBDO method. First, the method is computationally efficient as it does not require building a biasing density from scratch at every iteration and can build efficient biasing densities by reusing existing information. The computational efficiency is demonstrated through two numerical experiments. Second, the method can overcome bad initial biasing densities by reusing samples to build (at every design iteration) *a posteriori* biasing densities for future reuse. Third, the method is potentially useful for building biasing densities for disconnected feasible regions or multiple failure regions because it uses a mixture of existing biasing densities. Using mixture densities has been shown to be useful for multiple disconnected failure regions for probability of failure estimation [27, 28]. The novelty of our approach lies in the way we choose the mixture densities from the existing biasing densities, and the mixing weights in the context of RBDO. Fourth, there is no bias in the IRIS-RBDO reliability analysis and it can maintain a desired level of accuracy in the reliability estimates in every optimization iteration.

The rest of the paper is structured as follows. Section 2 provides the RBDO formulation and background on methods used to estimate the probability of failure. Section 3 describes the details of the proposed IRIS-RBDO method along with the complete algorithm. The effectiveness of IRIS-RBDO is shown using a benchmark speed reducer problem in Section 4 and a combustion engine model in Section 5. Section 6 presents the conclusions.

2. Reliability-based design optimization (RBDO)

This section describes the RBDO formulation used in this work (Section 2.1) followed by existing Monte Carlo methods for estimating the probability of failure. Section 2.2 describes the Monte Carlo estimate and Section 2.3 describes the importance sampling estimate for probability of failure.

2.1. RBDO formulation

The inputs to the system are n_d design variables $\boldsymbol{d} \in \mathcal{D} \subseteq \mathbb{R}^{n_d}$ and an n_r-dimensional random variable $Z : \Xi \to \Omega \subseteq \mathbb{R}^{n_r}$ defined on the sample space Ξ and with the probability density function p, henceforth called nominal density. Here, $\mathcal{D}$ denotes the design space and Ω denotes the random

sample space. A realization of Z is denoted as $\boldsymbol{z} \in \Omega$. We use $\boldsymbol{z} \sim p$ to indicate that the realizations are sampled from distribution p. We are interested in the RBDO problem that uses a reliability constraint—herein a failure probability—to drive the optimization. The RBDO problem formulation used in this work is

$$\begin{aligned} \min_{\boldsymbol{d}\in\mathcal{D}} \; & J(\boldsymbol{d}) = \mathbb{E}_p[f(\boldsymbol{d}, Z)] \\ \text{subject to} \quad & \mathbb{P}(g(\boldsymbol{d}, Z) < 0) \leq P_{\text{thresh}}, \end{aligned} \tag{1}$$

where $J : \mathcal{D} \mapsto \mathbb{R}$ is the cost function, $f : \mathcal{D} \times \Omega \mapsto \mathbb{R}$ is the quantity of interest, $g : \mathcal{D} \times \Omega \mapsto \mathbb{R}$ is the limit state function, and P_{thresh} is the acceptable threshold on the probability of failure. Without loss of generality, the failure of the system is defined by $g(\boldsymbol{d}, Z) < 0$. With $\mathbb{E}_p[f(\boldsymbol{d}, Z)] = \int_\Omega f(\boldsymbol{d}, \boldsymbol{z})\ p(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$ we denote the expectation of the random variable $f(\cdot, Z)$ with respect to the density p.

2.2. Monte Carlo estimate for probability of failure

The solution of the RBDO problem given by Eq. (1) is obtained via an iterative procedure where the optimizer evaluates a sequence of design iterates while seeking a minimum-cost solution that satisfies the reliability constraint. Let $\boldsymbol{d}_t$ be the design in optimization iteration t and define the corresponding failure set as

$$\mathcal{G}_t = \{\boldsymbol{z} \mid \boldsymbol{z} \in \Omega,\ g(\boldsymbol{d}_t, \boldsymbol{z}) < 0\}. \tag{2}$$

We emphasize that evaluating the limit state function, g, and hence checking if $\boldsymbol{z} \in \mathcal{G}_t$, requires evaluation of an expensive-to-evaluate model. The indicator function $\mathbb{I}_{\mathcal{G}_t} : \mathcal{D} \times \Omega \to \{0, 1\}$ is defined as

$$\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}) = \begin{cases} 1, & \boldsymbol{z} \in \mathcal{G}_t, \\ 0, & \text{else.} \end{cases} \tag{3}$$

The Monte Carlo estimate of the probability of failure $P(\boldsymbol{d}_t) := \mathbb{P}(g(\boldsymbol{d}_t, Z) < 0)$ is given by

$$\widehat{P}_p^{\text{MC}}(\boldsymbol{d}_t) = \frac{1}{m_t} \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}_i), \quad \boldsymbol{z}_i \sim p, \tag{4}$$

where $\boldsymbol{z}_i$, $i = 1, \ldots, m_t$ are the m_t samples from probability density p used in iteration t. The subscript for $\widehat{P}$ denotes the density from which the random variables are sampled to compute the estimate.

In Monte Carlo simulation for estimating small probabilities, the number of samples required to achieve a fixed level of accuracy in the probability estimate scales inversely with the probability itself. Due to the low probability of failure for reliable designs, standard Monte Carlo sampling would be computationally infeasible for expensive-to-evaluate limit state functions because the number of samples, m_t, required to reach an acceptable level of accuracy would be prohibitively large.

2.3. Importance sampling estimate for probability of failure

Importance sampling is a change of measure—from the nominal density to the *biasing density*—that is corrected via re-weighting of the samples drawn from the new measure. In probability of failure estimation, a biasing density is sought so that many samples lie in the set $\mathcal{G}_t$. In this work, we use a parametric biasing density denoted by q_{θ_t} for optimization iteration t, where θ_t

denotes the parameters of the distribution. The biasing density must satisfy $\text{supp}(\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, Z)p(Z)) \subset \text{supp}(q_{\theta_t}(Z))$. The importance sampling estimate for $P(\boldsymbol{d}_t)$ is given by

$$\widehat{P}^{\text{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t) = \frac{1}{m_t}\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i)\frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}, \quad \boldsymbol{z}'_i \sim q_{\theta_t}, \tag{5}$$

where $\boldsymbol{z}'_i$, $i = 1, \ldots, m_t$ are the m_t samples from probability density q_{θ_t} used in iteration t. The ratio $\frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}$ is called the importance weight, or likelihood ratio.

The unbiased sample variance for the importance sampling estimate is defined by $\frac{\widehat{\sigma}^2_{m_t}}{m_t}$, where $\widehat{\sigma}^2_{m_t}$ is estimated by

$$\widehat{\sigma}^2_{m_t} = \frac{1}{m_t - 1}\sum_{i=1}^{m_t}(\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i)\frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} - \widehat{P}^{\text{IS}}_{q_{\theta_t}})^2, \quad \boldsymbol{z}'_i \sim q_{\theta_t}.$$

The relative error, or coefficient-of-variation, in the probability of failure estimate is given by

$$e(\widehat{P}^{\text{IS}}_{q_{\theta_t}}) = \frac{1}{\widehat{P}^{\text{IS}}_{q_{\theta_t}}}\sqrt{\frac{\widehat{\sigma}^2_{m_t}}{m_t}}. \tag{6}$$

The importance sampling estimate of the failure probability is unbiased, i.e.,

$$\mathbb{E}_p[\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \cdot)] = \mathbb{E}_{q_{\theta_t}}\left[\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \cdot)\frac{p}{q_{\theta_t}}\right].$$

3. IRIS-RBDO: Information reuse in importance sampling for RBDO

We propose an efficient importance-sampling-based RBDO method that reduces computational cost through two levels of reusing existing information:

1. Reusing existing samples from the reliability computation in the current iteration to build an *a posteriori* biasing density with optimal parameters (see Theorem 1) that minimize the Kullback-Leibler divergence measure as described in Section 3.1.
2. Reusing existing biasing densities from nearby designs as described in Section 3.2.

The complete IRIS-RBDO algorithm is summarized in Section 3.3.

3.1. Reusing samples for a posteriori biasing density with optimal parameters

The first level of information reuse consists of building an *a posteriori* biasing density. We propose a method for approximating the optimal biasing density at current iteration t by reusing the current batch of samples that are used in the probability of failure computation. This first level of reuse builds on the ideas developed in the cross-entropy method for probability of failure estimation [24] applied to the context of reusing samples in the RBDO setup. The novelty lies in the way KL divergence is applied to reusing existing data in the optimization loop (after getting the reliability estimate), which is tailored to the two-loop RBDO setup. While we do use KL divergence as a distance measure to formulate the optimization problem to solve for the *a posteriori* density, our approach is different from the cross-entropy method (where KL divergence is used during the

reliability estimate at a particular design in the RBDO iteration). Note that this reuse method can be extended to use with cross-entropy method, where the initial density can be defined by the mixture of these *a posteriori* densities (as described in Section 3.2) to reduce the number of cross-entropy iterations.

After evaluating the importance-sampling failure probability estimate $\widehat{P}^{\mathrm{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t)$ with density q_{θ_t}, we can compute an *a posteriori* biasing density that is close to the optimal (also known as zero-variance) biasing density. It is known [45, Chapter 9] that the theoretical optimal biasing density results in the estimate $\widehat{P}^{\mathrm{IS}}_{h_t^*}(\boldsymbol{d}_t)$ having zero variance, and is given by

$$h_t^*(\boldsymbol{z}) = \frac{\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}) p(\boldsymbol{z})}{P(\boldsymbol{d}_t)}, \tag{7}$$

where the superscript for h_t^* denotes that it is the optimal biasing density for RBDO iteration t. However, due to the occurrence of $P(\boldsymbol{d}_t)$ in Eq. (7), it is not practical to sample from this density.

Consequently, we want to find an *a posteriori* biasing density that is close to h_t^*. In this work, the KL divergence [46] is used as the distance measure between two distributions. We thus define a density q_θ parameterized by $\theta \in \mathcal{P}$ and want to minimize the KL divergence to h_t^*, which is defined as

$$\mathrm{KL}(h_t^* \parallel q_\theta) = \mathbb{E}_{h_t^*}\left[\ln\left(\frac{h_t^*}{q_\theta}\right)\right] = \int_{-\infty}^{\infty} \ln\left(\frac{h_t^*(\boldsymbol{z})}{q_\theta(\boldsymbol{z})}\right) h_t^*(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}. \tag{8}$$

The optimal parameters for q_θ for RBDO iteration t are given by θ_t^*, where the superscript denotes that it is the optimal solution. Then θ_t^* can be found by solving an optimization problem given by

$$\begin{aligned}
\theta_t^* &= \arg\min_{\theta\in\mathcal{P}} \ \mathrm{KL}(h_t^* \parallel q_\theta) \\
&= \arg\min_{\theta\in\mathcal{P}} \ \mathbb{E}_{h_t^*}\left[\ln\left(\frac{h_t^*}{q_\theta}\right)\right] \\
&= \arg\min_{\theta\in\mathcal{P}} \int_{\boldsymbol{z}\in\Omega} \ln(h_t^*(\boldsymbol{z})) h_t^*(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} - \int_{\boldsymbol{z}\in\Omega} \ln(q_\theta(\boldsymbol{z})) h_t^*(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\
&= \arg\min_{\theta\in\mathcal{P}} - \int_{\boldsymbol{z}\in\Omega} \ln(q_\theta(\boldsymbol{z})) h_t^*(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\
&= \arg\min_{\theta\in\mathcal{P}} \ -\mathbb{E}_{h_t^*}\left[\ln(q_\theta)\right] \\
&= \arg\min_{\theta\in\mathcal{P}} \ -\mathbb{E}_{p}\left[\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \cdot)\ln(q_\theta)\right],
\end{aligned} \tag{9}$$

where in the last step we used the definition of the optimal biasing density from Eq. (7), and dropped the term $P(\boldsymbol{d}_t)$ as it does not affect the optimization. Since the integral requires evaluating the failure region, we use again importance sampling with density q_{θ_t} to obtain an efficient estimate, i.e.,

$$\mathbb{E}_p\left[\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \cdot)\ln(q_\theta)\right] = \mathbb{E}_{q_{\theta_t}}\left[\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \cdot)\frac{p}{q_{\theta_t}}\ln(q_\theta)\right]. \tag{10}$$

Overall, we obtain the closest biasing density in KL distance via

$$\theta_t^* = \arg\min_{\theta\in\mathcal{P}} \ -\mathbb{E}_{q_{\theta_t}}\left[\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \cdot)\frac{p}{q_{\theta_t}}\ln(q_\theta)\right] \approx \arg\min_{\theta\in\mathcal{P}} \ -\sum_{i=1}^{m_t}\left[\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}_i')\frac{p(\boldsymbol{z}_i')}{q_{\theta_t}(\boldsymbol{z}_i')}\ln(q_\theta(\boldsymbol{z}_i'))\right], \tag{11}$$

where we replaced the expectation by an importance sampling estimate and $\boldsymbol{z}'$ is sampled from the biasing density q_{θ_t}. Note that $\boldsymbol{z}'$ samples are existing samples that are reused after the probability of failure for $\boldsymbol{d}_t$ is already estimated.

We choose a multivariate normal distribution as the parametric distribution for q_θ. However, the method can be applied to any choice of parametric distribution. In order to find analytic solutions for the parameters, the chosen distribution can be mapped to an exponential family. One can also directly sample from the zero-variance optimal biasing density h_t^* using Markov chain Monte Carlo and this could be a possible extension to the proposed method.

For the case of the multivariate normal distribution, i.e., $q_{\theta_t^*} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, (and for several other parametric distributions, specifically the exponential family of distributions), an analytic solution of the optimal parameters θ_t^* can be derived and shown to be the global optimum for Eq. (11) as described below.

Theorem 1. *Let $q_{\theta_t^*} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ be a multivariate normal distribution, with the mean vector $\boldsymbol{\mu}_t = \left[\mu_t^1, \ldots, \mu_t^{n_r}\right]^\top$ and $\boldsymbol{\Sigma}_t = \left[\Sigma_t^{j,k}\right]_{j,k=1,\ldots,n_r}$ being the symmetric positive definite covariance matrix. Let $\boldsymbol{z}'_i = \left[z'_{i,1}, \ldots, z'_{i,n_r}\right]^\top \sim q_{\theta_t}$ represent the ith sample vector and $z'_{i,j}$ represent the jth entry of the ith sample vector for $j \in \{1, \ldots, n_r\}$. Then the parameters $\theta_t^* = \{\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\}$ are given by*

$$\mu_t^j = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} z'_{i,j}}{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}} = \frac{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} z'_{i,j}}{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}}, \tag{12}$$

$$\Sigma_t^{j,k} = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} (z'_{i,j} - \mu_t^j)(z'_{i,k} - \mu_t^k)}{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}} = \frac{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} (z'_{i,j} - \mu_t^j)(z'_{i,k} - \mu_t^k)}{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}}. \tag{13}$$

and are the global optimum for the optimization problem given by Eq. (11)*.*

Proof. See Appendix A. □

Constructing the *a posteriori* biasing density by reusing the existing samples as proposed here can help overcome a bad initial biasing density. Example 2 presents a two-dimensional example to illustrate the effectiveness of the *a posteriori* biasing density constructed through the first level of information reuse in the proposed IRIS-RBDO method. These *a posteriori* biasing densities are then stored in a database for future optimization iterations to facilitate the second level of reuse in IRIS-RBDO as described in Section 3.2.

Example 2. *We give a simple example to illustrate the reuse of samples to build the a posteriori biasing density, which constitutes the first level of information reuse in IRIS-RBDO. We compare with a common method to built biasing densities following [18], where the biasing density is chosen to be the normal distribution with mean shifted to the most probable failure point (MPP, see Appendix B on how to compute) and the same standard deviation as the nominal density.*

Given is a two-dimensional random variable Z with nominal density $p \sim \mathcal{N}\left(\begin{bmatrix}1\\10\end{bmatrix}, \begin{bmatrix}0.1^2 & 0\\0 & 3^2\end{bmatrix}\right)$. The limit state function is $g(\boldsymbol{z}) = 18 - z_1 - z_2$. Failure is defined as $g(\boldsymbol{z}) < 0$. In this case, the MPP is located at $z_1 = 1.0078, z_2 = 16.9922$. Therefore, the MPP-based biasing density is $q^{MPP} \sim \mathcal{N}\left(\begin{bmatrix}1.0078\\16.9922\end{bmatrix}, \begin{bmatrix}0.1^2 & 0\\0 & 3^2\end{bmatrix}\right)$.

KL-MVN denotes the first level of information reuse in IRIS-RBDO used for building the a posteriori biasing density with optimal parameters θ^* for the multivariate normal distribution using KL divergence as described in Section 3.1. In this case, $q_{\theta^*} \sim \mathcal{N}\left(\begin{bmatrix}1.0107\\18.0124\end{bmatrix}, \begin{bmatrix}0.01 & -0.0112\\-0.0112 & 0.8812\end{bmatrix}\right)$.

We compute the probability of failure using importance sampling to be 9.7×10^{-3}. *The coefficient of variation of probability of failure estimate using the MPP-based biasing density is* 0.0164 *and KL-MVN-based biasing density is* 0.0081. *In both cases,* 10^4 *samples are used from the biasing densities. For this example, we observe that the KL-MVN biasing density (constructed without any additional calls to the limit state function) is a better biasing density since it leads to a reduction in the coefficient of variation by around a factor of two compared to using the MPP-based biasing density. Notice that this example has a linear limit state function which makes it easy to find the MPP, leading to a good biasing density using MPP. This makes the reduction in coefficient of variation of probability of failure estimate by a factor of two using KL-MVN biasing density even more impressive and the gains will be potentially much higher for non-linear limit states, where finding a good biasing density using MPP is much more difficult.*

Figure 2 shows the resulting biasing densities and the failure boundary. Note that the MPP-based biasing density successfully tilts the distribution towards the failure region. However, since the MPP-based method chooses the same variance as the nominal density with mean on the failure boundary, samples drawn from that biasing density are far from the failure boundary and about 50% of the samples are in the safe region. This results in either small importance weights (which are prone to numerical errors and increase variance in the estimate) or uninformative samples (in the safe region with indicator function equal to zero). As can be seen from Figure 2, the information-reuse-based a posteriori biasing density places a large portion of samples in the failure region, and close to the failure boundary, which leads to good importance weights and variance reduction.

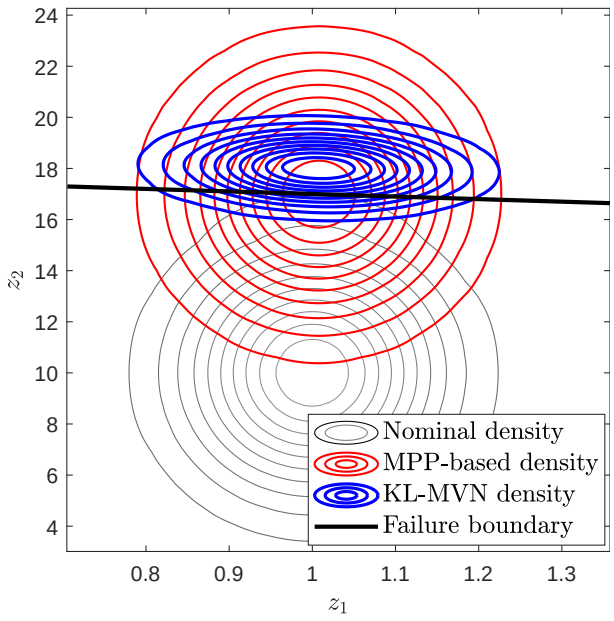


Figure 2: Illustrative example comparing *a posteriori* biasing density with MPP-based biasing density.

3.2. Reusing biasing densities from nearby designs for failure probability computation

The second level of information reuse involves reusing the existing *a posteriori* biasing densities (see Section 3.1) from past RBDO iterations to construct a biasing density in current design iteration t. We propose to reuse the *a posteriori* biasing densities corresponding to existing designs $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{t-1}$ from past optimization iterations that are close in design space.

A neighborhood of designs is defined as: $\mathcal{M}_t := \{\boldsymbol{d}_j \mid 0 \leq j \leq t-1,\ \|\boldsymbol{d}_j - \boldsymbol{d}_t\|_2 \leq r\}$, where r is the radius of the hypersphere defined by the user. The weights $\beta_j,\ j = 0, \ldots, t-1$ for existing *a posteriori* biasing densities are defined according to the relative distance of design point $\boldsymbol{d}_t$ to previously visited design points as

$$\beta_j := \begin{cases} \frac{\|\boldsymbol{d}_j - \boldsymbol{d}_t\|_2^{-1}}{\sum_{\boldsymbol{d}_j \in \mathcal{M}_t} \|\boldsymbol{d}_i - \boldsymbol{d}_t\|_2^{-1}}, & 0 \leq j \leq t-1,\ \|\boldsymbol{d}_j - \boldsymbol{d}_t\|_2 \leq r \\ 0, & \text{else} \end{cases}. \tag{14}$$

Note that $\sum_{j=0}^{t-1} \beta_j = 1$. The weights for each biasing density in our second level of information reuse reflects a correlation between the designs, which exists if designs are close to each other. Nearby designs from past optimization iterations will likely have similar failure regions and thus similar biasing densities. We set the radius of the hypersphere to only include designs that are in close proximity to the current design, and otherwise set the weight to zero as seen in Eq. (14). In this work, we chose the l^2-norm as the distance metric, however, any other norm can also be used in this method.

The information reuse biasing density for current design iteration t is defined by the mixture of existing *a posteriori* biasing densities as given by

$$q_{\theta_t} := \sum_{j=0}^{t-1} \beta_j q_{\theta_j^*}, \tag{15}$$

where $q_{\theta_j^*}$ are all the *a posteriori* biasing densities constructed using the first level of information reuse (see Section 3.1) from the past RBDO iterations $j \in \{0, \ldots, t-1\}$ that are stored in a database. Using a mixture distribution for constructing the biasing density also has the potential to capture multiple disconnected failure regions.

3.3. Algorithm and implementation details

Algorithm 1 describes the implementation of IRIS-RBDO method that constitutes of two levels of information reuse for an RBDO problem as described by Eq. (1). In this work, we set the radius of the specified hypersphere r to 0.5% of the largest diagonal of the hypercube defining the design space. We note that a distance measure of this nature works best if the design space is normalized so that each design variable has the same scale. The second level of information reuse described in Section 3.2 is used only if there are nearby designs. If there are no nearby designs ($\mathcal{M}_t = \emptyset$) for current optimization iteration t, we use an MPP-based [18] method (see Appendix B on how to compute) for building a biasing density for importance sampling with no reuse. However, any other method depending on the user's preference can be chosen to build the biasing density for importance sampling with no reuse. Note that we do not put any restrictions on how many designs to reuse, i.e., we reuse information from all nearby designs within the specified radius r, see Eq. (14). However, the user can choose to limit the number of reused designs.

Algorithm 1 IRIS-RBDO: Information reuse in importance-sampling-based RBDO

Input: Cost function $J(\cdot)$, limit state function $g(\cdot)$, nominal density p, design space $\mathcal{D}$, initial design $\boldsymbol{d}_0$, hypersphere radius r, coefficient of variation tolerance ϵ_{tol}, maximum number of samples $m_{\max}$, threshold on probability of failure P_{thresh}

Output: Optimal design $\boldsymbol{d}^*$, optimal cost J_{opt}

procedure IRIS_RBDO($p, \mathcal{D}, \boldsymbol{d}_0, r, \epsilon_{\text{tol}}, m_{\max}, P_{\text{thresh}}$)
  Build biasing density q_{θ_0} for $\boldsymbol{d}_0$ from scratch ▷ For MPP-based method, see Appendix B
  $(\widehat{P}^{\text{IS}}_{q_{\theta_0}}(\boldsymbol{d}_0), \mathcal{G}_0) = \text{ProbabilityOfFailure}(p, q_{\theta_0}, \boldsymbol{d}_0, \epsilon_{\text{tol}}, m_{\max})$ ▷ See Algorithm 2
  Evaluate cost function $J(\boldsymbol{d}_0)$
  if $\widehat{P}^{\text{IS}}_{q_{\theta_0}}(\boldsymbol{d}_0) \leq P_{\text{thresh}}$ **then** ▷ Check if design is reliable
    $J_{\text{opt}} \leftarrow J(\boldsymbol{d}_0)$ ▷ Assign optimal cost
    $\boldsymbol{d}^* \leftarrow \boldsymbol{d}_0$ ▷ Assign optimal design
  else
    $J_{\text{opt}} \leftarrow 10^{10}$ ▷ Initialize optimal cost
  end if

  $t = 0$
  while optimization not converged **do**
    Calculate $\boldsymbol{\mu}_t$ using Eq. (12) and samples from q_{θ_t}
    Calculate $\boldsymbol{\Sigma}_t$ using Eq. (13) and samples from q_{θ_t}
    $q_{\theta_t^*} \leftarrow \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ ▷ **Reusing existing samples: *a posteriori* biasing density**

    $t \leftarrow t + 1$
    Get design $\boldsymbol{d}_t$ from optimizer
    $\mathcal{M}_t \leftarrow \{\boldsymbol{d}_j \mid 0 \leq j \leq t-1,\ \|\boldsymbol{d}_j - \boldsymbol{d}_t\|_2 \leq r\}$ ▷ Neighborhood designs
    if $\mathcal{M}_t \neq \emptyset$ **then** ▷ Nearby designs exist
      Compute weights $\beta_j\ \forall\ j = 0, \ldots, t-1$ using Eq. (14)
      $q_{\theta_t} \leftarrow \sum_{j=0}^{t-1} \beta_j q_{\theta_j^*}$ ▷ **Reusing existing biasing densities: mixture of** $q_{\theta_j^*}$
    else
      Build biasing density q_{θ_t} from scratch ▷ No nearby previous designs
    end if
    $(\widehat{P}^{\text{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t), \mathcal{G}_t) = \text{ProbabilityOfFailure}(p, q_{\theta_t}, \boldsymbol{d}_t, \epsilon_{\text{tol}}, m_{\max})$ ▷ See Algorithm 2
    Evaluate cost function $J(\boldsymbol{d}_t)$

    if $J(\boldsymbol{d}_t) \leq J_{\text{opt}}$ **and** $\widehat{P}^{\text{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t) \leq P_{\text{thresh}}$ **then**
      $J_{\text{opt}} \leftarrow J(\boldsymbol{d}_t)$ ▷ Assign optimal cost
      $\boldsymbol{d}^* \leftarrow \boldsymbol{d}_t$ ▷ Assign optimal design
    end if
  end while
  return $\boldsymbol{d}^*, J_{\text{opt}}$
end procedure

Algorithm 2 shows the implementation for the failure probability estimation required in every iteration of Algorithm 1. We require the coefficient of variation defined in Eq. (6), in the probability

of failure estimate to be within acceptable tolerance ϵ_{tol} at iteration t. That is, we require

$$e(\widehat{P}^{\text{IS}}_{q_{\theta_t}}) \leq \epsilon_{\text{tol}}. \tag{16}$$

The value for ϵ_{tol} can be set by the user depending on the level of accuracy required for a specific application. We choose $\epsilon_{\text{tol}} \in [10^{-2}, 10^{-1}]$ in the applications presented in this paper. We follow an iterative process to add samples and check if the coefficient of variation is below a specified error tolerance. In this work, 100 samples are added at a time. However, to ensure termination of the algorithm in case this criterion is not met, we set a maximum number of samples $m_{\max}$ that shall not be exceeded at every design iteration. Note that $m_{\max}$ can typically be set by taking into account the values of P_{thresh} and ϵ_{tol}.

Algorithm 2 Probability of failure estimate

Input: Limit state function $g(\cdot)$, design $\boldsymbol{d}_t$, nominal density p, biasing density q_{θ_t}, coefficient of variation tolerance ϵ_{tol}, maximum number of samples $m_{\max}$
Output: Probability of failure estimate $\widehat{P}^{\text{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t)$, failure set $\mathcal{G}_t$

procedure ProbabilityOfFailure($p, q_{\theta_t}, \boldsymbol{d}_t, \epsilon_{\text{tol}}, m_{\max}$)
  $m_t = 0$ ▷ Number of samples
  $m_{\text{add}} = 100$ ▷ 100 samples are added at a time
  $e(\widehat{P}^{\text{IS}}_{q_{\theta_t}}) = 100\epsilon_{\text{tol}}$
  $\mathcal{Z}'_t = \emptyset$
  while $(m_t \leq m_{\max})$ **or** $(e(\widehat{P}^{\text{IS}}_{q_{\theta_t}}) > \epsilon_{\text{tol}})$ **do**
    Get m_{add} samples $\{\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_{m_{\text{add}}}\}$ from q_{θ_t}
    $m_t \leftarrow m_t + m_{\text{add}}$
    $\mathcal{Z}'_t \leftarrow \mathcal{Z}'_t \cup \{\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_{m_{\text{add}}}\}$
    Compute probability of failure

$$\widehat{P}^{\text{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t) = \frac{1}{m_t} \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}$$

    Calculate coefficient of variation in probability of failure estimate $e(\widehat{P}^{\text{IS}}_{q_{\theta_t}})$ using Eq. (6)
  end while
  $\mathcal{G}_t \leftarrow \{\boldsymbol{z} \mid \boldsymbol{z} \in \mathcal{Z}'_t,\ g(\boldsymbol{d}_t, \boldsymbol{z}) < 0\}$ ▷ Failure set
  return $\widehat{P}^{\text{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t), \mathcal{G}_t$
end procedure

After estimating $\widehat{P}^{\text{IS}}_{q_{\theta_t}}(\boldsymbol{d}_t)$ within the specified relative error tolerance ϵ_{tol}, we reuse the existing samples to construct the *a posteriori* biasing density $q_{\theta^*_t}$ with optimal parameters θ^*_t using the method described in Section 3.1. The algorithm then proceeds to the next optimization iteration.

Remark 3 (Defensive importance sampling)**.** *Importance sampling (with a good biasing density) is efficient for small probabilities and we are commonly interested in low probabilities of failure in reliable engineering design applications. It should be noted that for large probabilities, importance sampling can be inefficient. To ensure a robust method to guard against such cases when building biasing densities for importance sampling with no reuse, one can use defensive importance sampling (described in Appendix C) in combination with a method of choice for importance sampling (in*

our case, MPP-based). However, we found biasing densities built using IRIS to be efficient without the use of defensive importance sampling even for higher probabilities of failure. A reason is that we build the a posteriori biasing density via a weighted sample average as given by Eqs. (12) *and* (13)*, where the weights depend on the distance from the nominal density, naturally and efficiently encoding the defensive part. For high probabilities of failure, IRIS should potentially lead to similar number of required samples as defensive importance sampling or generic Monte Carlo sampling, and we see that to be the case in our numerical experiments.*

Remark 4 (Optimizer and gradients)**.** *We emphasize that this work does not develop a particular optimization algorithm for RBDO but provides a general method of efficiently integrating information from past optimization iterations into the reliability analysis. In this work, we show the efficiency of the proposed method for both a gradient-free optimizer and a gradient-based optimizer. Typically, it is difficult to accurately estimate gradients of probability of failure making it challenging to use a gradient-based optimizer for RBDO. However, for high-dimensional optimization problems gradient-based optimizers often are the only choice. If one wants to use a gradient-based optimizer for black-box optimization, finite difference (although not the most efficient) is a generic choice for calculating derivatives in a number of off-the-shelf optimizers. The IRIS method offers an advantage when finite difference is used to calculate the gradients because it reuses biasing densities from a very close design (since the finite difference step-size is very small) and leads to efficient estimates for the probability of failure gradients. For differentiable $\widehat{P}^{IS}_{q_\theta}(\boldsymbol{d})$, the finite difference estimate for derivative of the probability of failure at any given design $\boldsymbol{d}$ is*

$$\frac{\partial \widehat{P}^{IS}_{q_\theta}(\boldsymbol{d})}{\partial d_i} \approx \frac{\widehat{P}^{IS}_{q_{\theta'}}(\boldsymbol{d}+\delta \boldsymbol{e}_i) - \widehat{P}^{IS}_{q_\theta}(\boldsymbol{d})}{\delta}, \forall i = 1, \ldots, n_d$$

where δ is a small perturbation, and $\boldsymbol{e}_i$ is the i^{th} unit vector. We can ensure that every probability of failure estimate meets a set error tolerance ϵ_{tol} as seen in Algorithm 2. The error in probability of failure estimates directly affect the variance of the finite difference estimator for the derivatives, and thus ensures that the variance is also under a certain tolerance. The variance of the finite difference estimate for the derivatives using IRIS is given by

$$\mathbb{V}\mathrm{ar}\left[\frac{\partial \widehat{P}^{IS}_{q_\theta}(\boldsymbol{d})}{\partial d_i}\right] \approx \frac{1}{\delta^2}\left(\mathbb{V}\mathrm{ar}[\widehat{P}^{IS}_{q_{\theta'}}(\boldsymbol{d}+\delta\boldsymbol{e}_i)] + \mathbb{V}\mathrm{ar}[\widehat{P}^{IS}_{q_\theta}(\boldsymbol{d})]\right) \leq \frac{\epsilon^2_{tol}}{\delta^2}\left(\widehat{P}^{IS}_{q_{\theta'}}(\boldsymbol{d}+\delta\boldsymbol{e}_i)^2 + \widehat{P}^{IS}_{q_\theta}(\boldsymbol{d})^2\right).$$

Notice that since $\delta(\ll r)$ is a small perturbation, we can use the IRIS method for rest of the n_d probability of failure estimates required in the finite difference scheme after estimating $\widehat{P}^{IS}_{q_\theta}(\boldsymbol{d})$. Thus, using IRIS we need substantially fewer samples to estimate $\widehat{P}^{IS}_{q_{\theta'}}(\boldsymbol{d}+\delta\boldsymbol{e}_i)$ while maintaining the same error tolerance ϵ_{tol} in the probability of failure estimates required in the finite difference scheme as compared to regular Monte Carlo estimator or importance sampling without reuse. The variance of the derivative is also controlled through the error tolerance given by Equation (16)*. Further variance reduction at the cost of additional bias can be obtained by using common random numbers [47, 48] for the finite difference scheme. There are also several approximate methods available for estimating the gradient of probability of failure that can also be used to solve the RBDO problem [49, 50, 12, 51].*

4. Benchmark problem: Speed reducer

The speed reducer problem used in Ref. [52, Ch.10] is a benchmark problem in RBDO. Here, we make the problem more challenging by modifying the limit state functions in order to lead

to lower probabilities of failure for the system. We set a lower threshold probability of failure of $P_{\text{thresh}} = 10^{-3}$, as compared to 10^{-2} in Ref. [52]. The tolerance on coefficient of variation in probability of failure estimation within IRIS-RBDO is set to $\epsilon_{\text{tol}} = 0.01$. This makes the estimation of the lower failure probabilities within the specified tolerance more expensive and challenging than the original problem. The inputs to the system are six design variables defined in Table 1 and three uncertain variables defined as uncorrelated random variables in Table 2. Note that the method can handle any distribution for the random variables and they do not need to be uncorrelated. We fix the number of gear teeth to 17.

Table 1: Design variables $\boldsymbol{d} = [d_1, \ldots, d_6] \in \mathcal{D} \subseteq \mathbb{R}^6$ used in the speed reducer application.

Design variable	Lower bound (mm)	Upper bound (mm)	Initial design (mm)	Best design (mm)
d_1	2.6	3.6	3.5	3.5
d_2	0.7	0.8	0.7	0.7
d_3	7.3	8.3	7.3	7.3
d_4	7.3	8.3	7.72	7.88
d_5	2.9	3.9	3.35	3.45
d_6	5.0	5.5	5.29	5.34

Table 2: Uncertain variables modeled as vector-valued random variable $\boldsymbol{z} \in \Omega \subseteq \mathbb{R}^3$ with realization $\boldsymbol{z} = [z_1, z_2, z_3]$ used in the speed reducer application.

Random variable	Distribution	Mean	Standard deviation (μm)
z_1	Normal	d_2	1
z_2	Normal	d_4	30
z_3	Normal	d_6	21

The RBDO problem formulation used in this work is given by

$$\begin{aligned} &\min_{\boldsymbol{d} \in \mathcal{D}} \; J(\boldsymbol{d}) = \mathbb{E}_p[f(\boldsymbol{d}, Z)] \\ &\text{subject to} \quad \mathbb{P}(g_i(\boldsymbol{d}, Z) < 0) \leq P_{\text{thresh}} = 10^{-3}, \; i \in 1, 2, 3, \end{aligned} \tag{17}$$

where

$$\begin{aligned} f(\boldsymbol{d}, \boldsymbol{z}) = {} & 0.7854 d_1 z_1^2 (3.3333 \times 17^2 + 14.9334 \times 17 - 43.0934) \\ & - 1.5079 d_1 (z_3^2 + d_6^2) + 7.477 (z_3^3 + d_6^3) + 0.7854 (z_2 z_3^2 + d_4 d_7^2), \end{aligned} \tag{18}$$

is a cost function that penalizes the material used in the manufacturing process with units of mm^3. The limit state functions are

$$\begin{aligned} g_1(\boldsymbol{d}, \boldsymbol{z}) &= 1 - \frac{1.93 z_2^3}{17 z_1 z_3^4}, \\ g_2(\boldsymbol{d}, \boldsymbol{z}) &= 1120 - \frac{A_1}{B_1}, \; A_1 = \left[\left(\frac{745 z_2}{17 z_1} \right)^2 + 16.9 \times 10^6 \right]^{0.5}, \; B_1 = 0.1 z_3^3, \\ g_3(\boldsymbol{d}, \boldsymbol{z}) &= 870 - \frac{A_2}{B_2}, \; A_2 = \left[\left(\frac{745 d_4}{17 z_1} \right)^2 + 157.5 \times 10^6 \right]^{0.5}, \; B_2 = 0.1 d_6^3. \end{aligned} \tag{19}$$

We used the COBYLA (constrained optimization by linear approximation) optimizer from the NLopt package to run the optimization and also set a cut-off for maximum number of samples to be used in each optimization iteration to $m_{\max} = 5 \times 10^5$. COBYLA is a gradient-free optimizer. We compare the efficiency of probability of failure estimates using the proposed IRIS-RBDO method that reuses information to importance sampling with no information reuse. We also compare those results to subset simulation [29], a state-of-the-art method in reliability analysis and failure probability estimation.[4]

Figure 3 (a) shows the IRIS-RBDO convergence history versus the cumulative computational cost in terms of number of samples used. We see that the optimization requires around 2×10^5 samples before it finds the first feasible design. The probability of failure history seen in Figure 3 (c) shows the progress of designs from infeasible to feasible regions during the optimization. The best design obtained in this case is given in Table 1, which had an associated cost of 3029.2 mm^3.

The total number of samples used in each optimization iteration in IRIS-RBDO versus importance sampling with no reuse and subset sampling is shown in Figure 4 (a). Note that we are showing the plots for the same designs in each RBDO iteration for all the cases, which makes it a one-to-one comparison. When no designs are nearby, our method also builds a biasing density with no reuse, hence the two markers overlap in those iterations. Otherwise, IRIS-RBDO always outperforms the other methods. IRIS-RBDO leads to overall computational savings of around 51% compared to importance sampling with no reuse and subset sampling throughout the optimization. Current implementation of subset simulation performs similar to the importance sampling with no reuse for the speed reducer problem. For this problem, subset simulation using 10^4 samples in each level does not meet the error tolerance (see Figure 4 (c)). However, IRIS and importance sampling with no reuse meet the set tolerance ($\epsilon_{\text{tol}} = 0.01$) on the coefficient of variation in probability of failure estimate for every optimization iteration as seen in Figure 4 (c).

Figure 4 (b) compares performance of IRIS-RBDO vs importance sampling with no reuse and subset simulation by showing the number of samples required for the corresponding probability of failure estimates. For the case when there is no reuse, we see that the required number of samples is approximately inversely proportional to the respective probability of failure. However, for IRIS the required number of samples depend on the quality and amount of information reused. In this case, using IRIS we considerably reduce the number of samples required even for lower probabilities of failure due to the extra information about the failure boundary encoded in the biasing density by reusing information from the past optimization iterations. As noted before, when the markers overlap it means that there was no nearby designs and the biasing density was built with no reuse (here, MPP-based) during IRIS-RBDO.

The number of designs reused in each optimization iteration of IRIS-RBDO is shown in Figure 5. Reusing designs leads to computational savings because of better biasing densities. As the iteration converges, IRIS-RBDO finds many close designs and beneficially reuses the biasing densities; compare this to Figure 4 (a) to see how reuse saves model evaluations. However, note that the computational savings are not directly proportional to the number of reused designs. For iterations where no designs were reused—typically in the early design space exploration stage—the biasing

[4]We use the recent Markov Chain Monte Carlo implementation for subset simulation of [30] (`https://www.mathworks.com/matlabcentral/fileexchange/57947-monte-carlo-and-subset-simulation-example?s_tid=prof_contriblnk`) where we use 10^4 samples in each level. Note that we tried different sample sizes for each level and settled on 10^4 in order to get close to the set error tolerance. For this problem, subset simulation does not meet the set error tolerance as seen later. We use the approximate error estimate for subset simulation from [30].

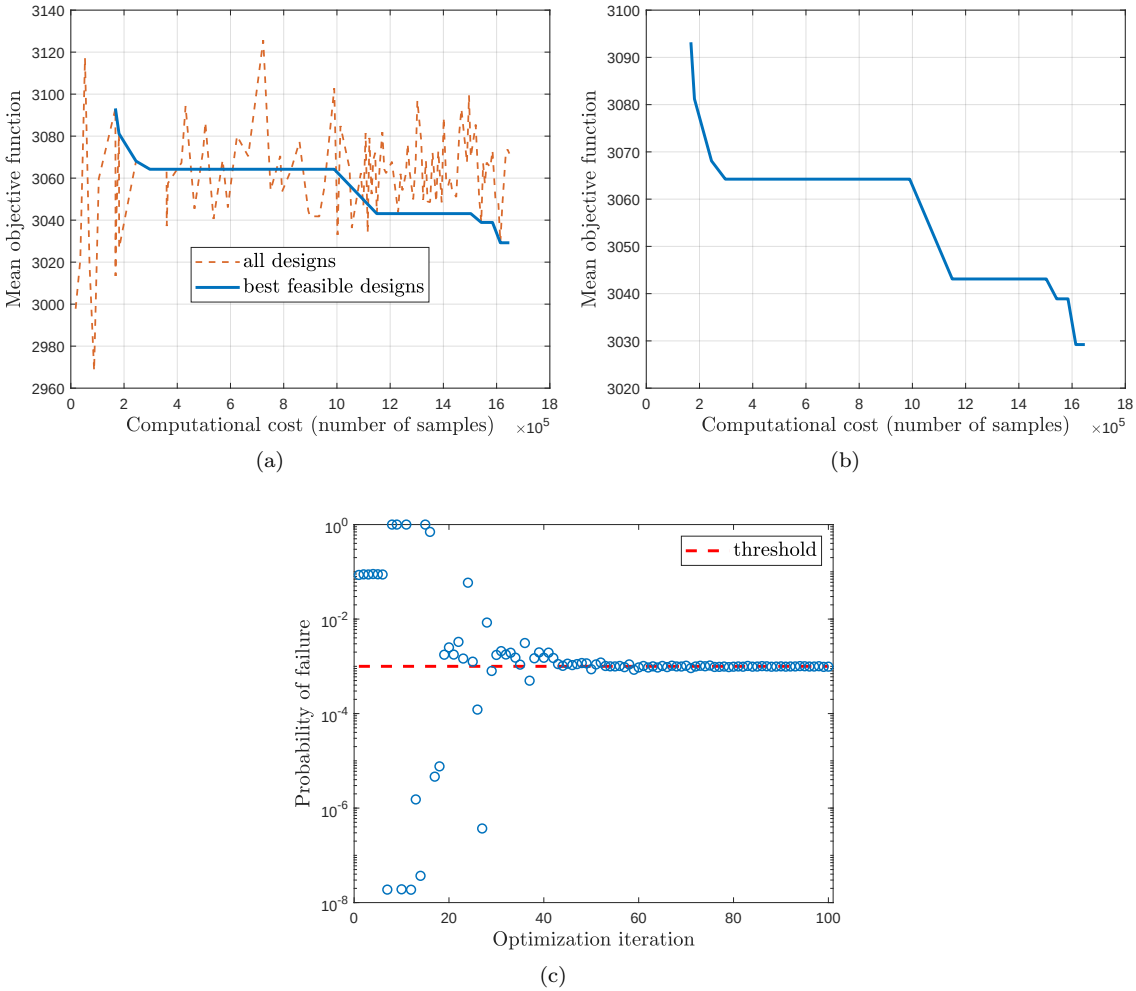


Figure 3: Optimization progress using IRIS-RBDO showing (a) the objective function value for designs from all optimization iterations, (b) magnified convergence plot of feasible designs against the cumulative computational cost in terms of number of samples, and (c) probability of failure history in each optimization iteration for the speed reducer problem.

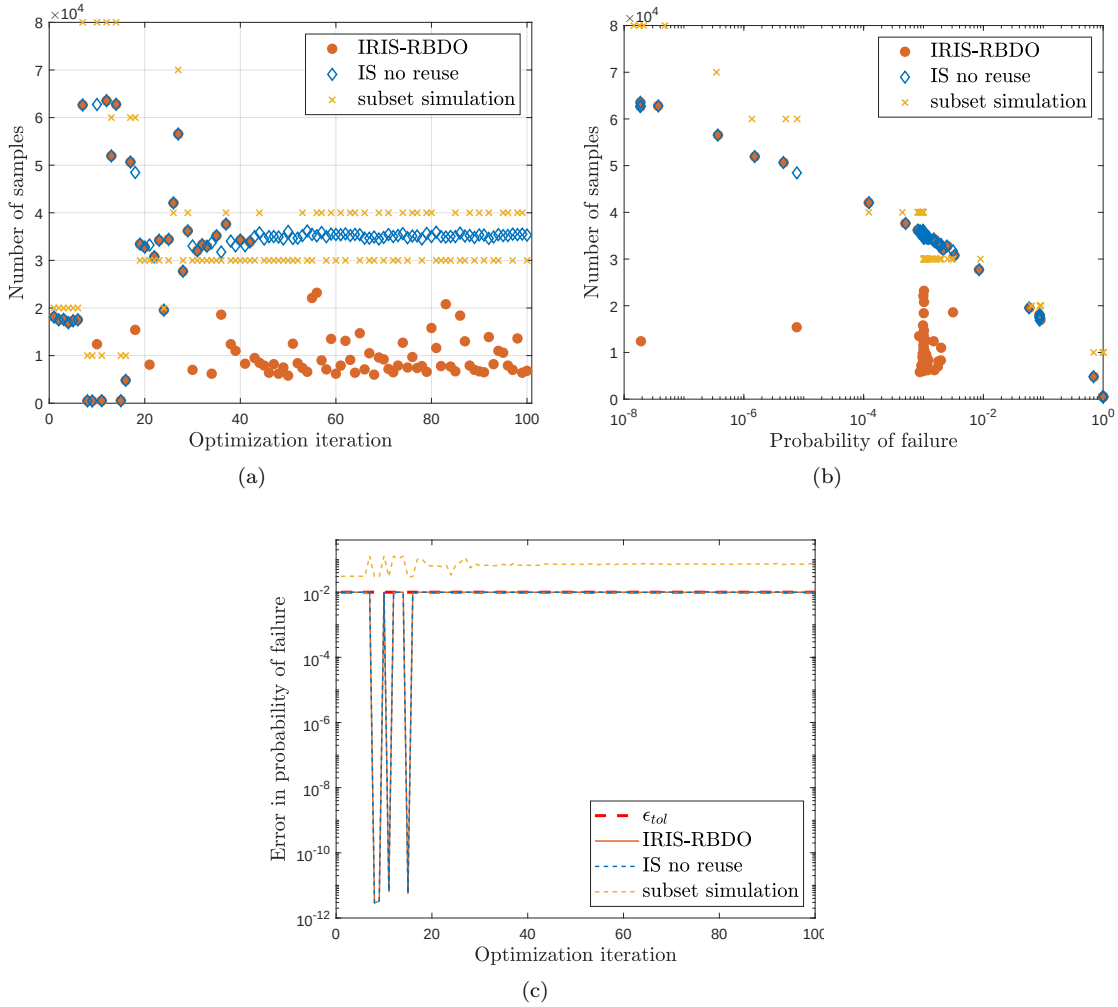


Figure 4: Comparison of IRIS, importance sampling (IS) with no reuse and subset simulation for the same designs showing (a) number of samples required in each optimization iteration, (b) number of samples required to calculate the corresponding probabilities of failure, and (c) error in probability of failure estimate (quantified by the coefficient of variation) in each optimization iteration for the speed reducer problem.

density was built without any information reuse (here, MPP-based).

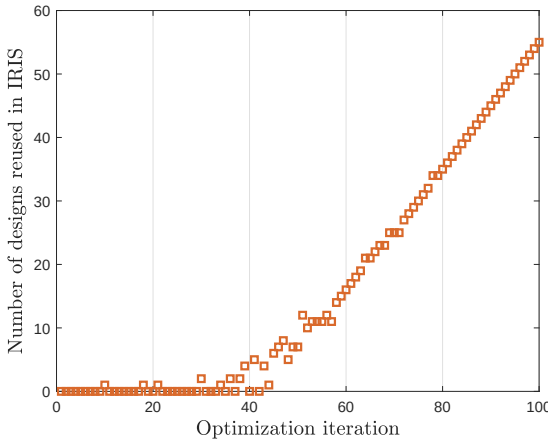


Figure 5: Number of designs reused in IRIS in each optimization iteration for the speed reducer problem.

5. RBDO for a combustion engine model

In this section, we show the effectiveness of the proposed IRIS-RBDO method when applied to the design of a combustion engine. The advent of reusable rockets for space flight requires new, more durable, engine designs [53]. Satisfying reliability constrains is not only important for safety, but also for durability, as failure to meet reliability constraints results in excessive wear of the engine, in turn limiting the rockets' repeated use. The computational model used to analyze the combustion engine is described in Section 5.1. Section 5.2 describes the RBDO problem formulation and the results are discussed in Section 5.3.

5.1. Computational model

We consider a continuously variable resonance combustor (CVRC), which is a single element model rocket combustor as illustrated in Figure 6. The CVRC is an experiment at Purdue University which has been extensively studied both experimentally [54, 55, 56] and computationally [57, 58, 59].

5.1.1. Governing equations and geometry

The experiment is modeled with a quasi-1D[5] partial differential equation model. Figure 7 shows the computational domain of the combustor, plotting the one-dimensional spatial variable x versus the combustor radius $R(x)$. From left to right, the figure shows the five important combustor segments, separated by the dashed lines: the injector, back-step, combustion chamber, back-step and nozzle. The injector length is $L_i = 3.49$cm, the chamber length is $L_c = 38.1$cm, the length

[5]The three dimensional state variables are averaged across the combustor, resulting in a stream-wise dependence of the state variables, i.e., states are x-dependent.

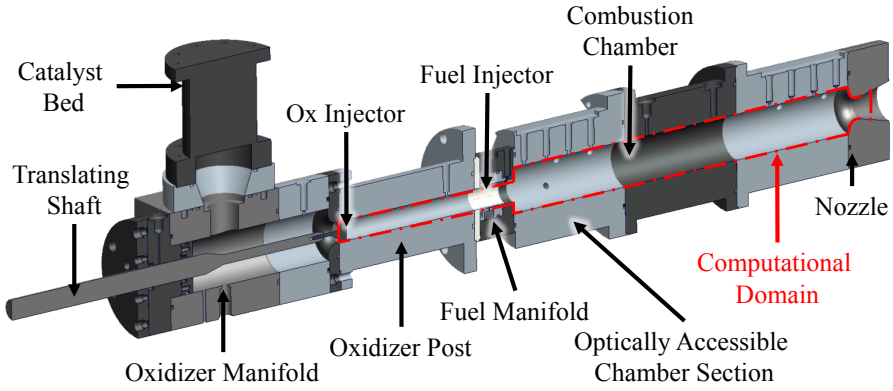


Figure 6: CVRC experimental configuration from [55]. The computational domain for the reactive flow computations is given in Figure 7.

of both backsteps is fixed at $L_{\text{bs}} = 3.81\text{cm}$ and the nozzle length is $L_{\text{n}} = 0.635\text{cm}$. The spatial variable is thus considered as $x \in -L_i \leq x \leq 2L_{\text{bs}} + L_{\text{c}} + L_{\text{n}}$. The injector radius is given by R_i and the combustion chamber radius is given by R_c. The nozzle radii are $R_{\text{t}} = 1.0401\text{cm}$ at the throat and 1.0922cm at the exit. The quasi-1D Euler partial differential equation model for the

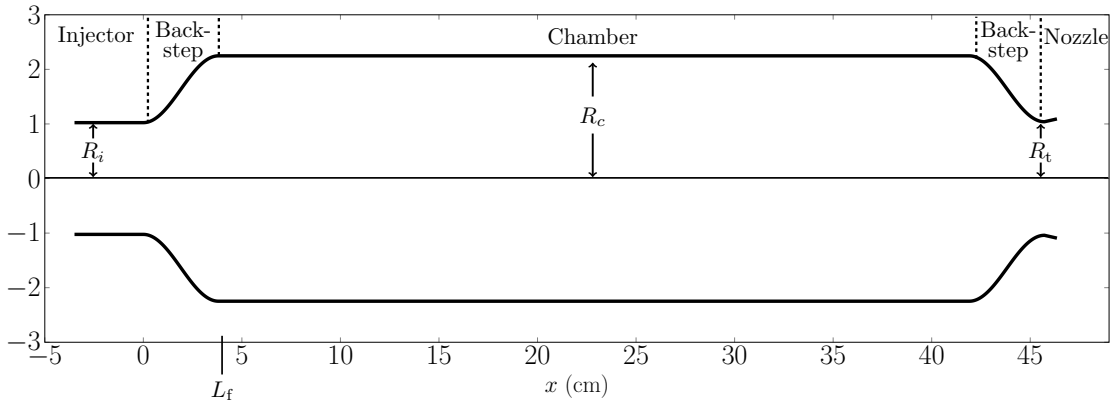


Figure 7: Computational domain (dashed area in Fig. 6) for the CVRC model combustor and its segments. The injector radius R_i and combustion chamber radius R_c are design parameters (for this plot chosen as the mean of the design parameter intervals). The location of the fuel source, L_f, is also a design variable.

CVRC [59, 60] is given as

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ E\\ \rho Y_{\text{ox}}\end{pmatrix} = -\frac{1}{A}\frac{\partial}{\partial x}\begin{pmatrix}A\rho u\\ A(\rho u^2+p)\\ Au(E+p)\\ A\rho u Y_{\text{ox}}\end{pmatrix} + \begin{pmatrix}\dot{\omega}_f\\ \frac{p}{A}\frac{\mathrm{d}A}{\mathrm{d}x} + \dot{\omega}_f u\\ \dot{\omega}_f \Delta h_0\\ -\dot{\omega}_f / C_{f/o}\end{pmatrix}, \tag{20}$$

which we solve for the steady-state solution $\frac{\partial}{\partial t} = 0$ via pseudo-time stepping. In the following we compute steady-state solutions for Eq. (20), i.e., time-independent solutions. The velocity is $u(x)$ and $Y_{\text{ox}}(x)$ is the oxidizer mass fraction. The state variables are density $\rho(x)$, specific momentum $\rho(x)u(x)$, total energy $E(x)$, and $\rho(x)Y_{\text{ox}}(x)$. The equation of state

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2 \tag{21}$$

relates energy and pressure $p(x)$ via the heat capacity ratio γ. In the source terms of Eq. (20), which model the chemical reaction and the cross-sectional area variation, Δh_0 denotes the heat of reaction, which is taken as a constant. The fuel-to-oxidizer ratio is the parameter $C_{f/o}$. Moreover, $A = A(x) = \pi R^2(x)$ encodes the cross-sectional area of the combustor as a function of x. Fuel at a mass-flow rate $\dot{m}_f$ is injected through an annular rig at the backstep after the oxidizer injector, centered at coordinate $x = L_f$, see also Fig. 7. The forcing function $\dot{\omega}_f$ in Eq. (20) is then modeled as

$$\dot{\omega}_f(x, \dot{m}_f) = \frac{\dot{m}_f}{A(x)\int_{-L_{\text{i}}}^{2L_{\text{bs}}+L_{\text{c}}+L_{\text{n}}}(1+\sin(\xi(x)))\text{d}x}(1+\sin(\xi(x))), \tag{22}$$

$$\xi(x) = \begin{cases} -\frac{\pi}{2} + 2\pi\frac{x-l_s}{l_f-l_s}, & l_s < x < l_f \\ 0, & \text{else} \end{cases}. \tag{23}$$

The computational model is a finite-volume discretization with upwinding, where we use 800 non-uniform finite volume elements and a fourth order Runge-Kutta integration scheme. The CPU time required for one evaluation of the computational model is on average around 20 seconds.

5.1.2. Boundary conditions

The inlet boundary condition is modeled via a subsonic inlet. At the inlet, we prescribe the oxidizer mass flow rate $\dot{m}_{\text{ox}}$ and the oxidizer concentration Y_{ox}. The inlet stagnation temperature T_0 is determined as follows: we prescribe a reference temperature T_∞ and reference pressure p_∞, which are typically given from upstream components of an engine. We then use the relation $T_0 = T_\infty + \frac{1}{2}\frac{\dot{m}_{\text{ox}}^2 R_{\text{gas}}^2 T_\infty^2}{A^2 p_\infty^2 C_p}$ with universal gas constant $R_{\text{gas}} = 8.314 \times 10^3 \frac{g}{mol}$ and specific heat of the fuel $C_p = 4.668 \times 10^3 \frac{J}{kg\ K}$. Due to the subsonic nature of the boundary, the pressure is extrapolated from the domain. Having $\dot{m}_{\text{ox}}, Y_{\text{ox}}, T_0, p$ at the inlet allows us to compute the boundary conditions for the state variables. The downstream boundary is modeled as a supersonic outlet, with constant extrapolation of the state variables.

5.1.3. Design variables

We define a four-dimensional design space $\mathcal{D}$ with the following design variables $\boldsymbol{d} \in \mathcal{D} \subseteq \mathbb{R}^4$: the geometric parameters of the injector radius R_i, the combustion chamber radius R_c and the location of the fuel injection L_f (see Figure 7), and the mass-flow rate $\dot{m}_f$ that enters into the forcing model in Eq. (22). The design variables $\boldsymbol{d} = [R_i,\ R_c,\ L_f,\ \dot{m}_f]$ and the respective bounds are given in Table 3.

5.1.4. Uncertain variables

The reference pressure p_∞ and the reference temperature T_∞ are typically measured from upstream components of the combustion engine and are therefore subject to uncertainty. They enter

Table 3: Design variables $\boldsymbol{d} = [R_i,\ R_c,\ L_f,\ \dot{m}_f] \in \mathcal{D} \subseteq \mathbb{R}^4$ used in the combustion engine problem.

Design variable	Description	Range	Initial design	Best design
R_i	Injector radius	[0.889, 1.143]cm	1.02	1.14
R_c	Combustion chamber radius	[1.778, 2.54]cm	2.16	2.41
L_f	Location of fuel injection	[3.5, 4]cm	3.75	3.5
$\dot{m}_f$	Mass flow rate for fuel injection	[0.026, 0.028]kg/s	0.027	0.026

in the inlet boundary conditions, see Section 5.1.2. Another uncertain variable is the fuel-to-oxidizer ratio $C_{f/o}$ which enters into the forcing term in the governing equations, Eq. (20), and in practice is also uncertain. Since all three uncertain variables are known within certain bounds, we model them as a vector-valued random variable Z with a uniform probability distribution. A realization of the random variable is $\boldsymbol{z} = [p_\infty,\ T_\infty,\ C_{f/o}] \in \Omega \subseteq \mathbb{R}^3$. We list the three uncertain variables and their respective probability distributions (in this case, uncorrelated) in Table 4.

Table 4: Random variable $\boldsymbol{z} \in \Omega \subseteq \mathbb{R}^3$ with realization $\boldsymbol{z} = [p_\infty,\ T_\infty,\ C_{f/o}]$ used in the combustion engine problem.

Uncertain variable	Description	Distribution	Range
p_∞	Upstream pressure	Uniform	[1.3, 1.6]MPa
T_∞	Upstream oxidizer temperature	Uniform	[1000, 1060]K
$C_{f/o}$	Fuel-to-oxidizer ratio	Uniform	[0.10, 0.11]

5.2. RBDO formulation: Objective function and reliability constraints

Having defined both the design variables and uncertain variables, we note that solutions to the state Eqs. (20)–(21) depend on the design $\boldsymbol{d} = [R_i,\ R_c,\ L_f,\ \dot{m}_f]$ and a realization $\boldsymbol{z} = [p_\infty,\ T_\infty,\ C_{f/o}]$ of the uncertain parameters, i.e., the pressure $p(x) = p(x; \boldsymbol{d}, \boldsymbol{z})$. We next describe the cost function for RBDO and the reliability constraints,which then completes the RBDO problem formulation from Eq.(1).

5.2.1. Cost function

We are interested in maximizing $C^\star$ ("C-star") efficiency, also known as characteristic exhaust velocity, a common measure of the energy available from the combustion process of the engine. To compute $C^\star$, we need the total mass flow rate at the exhaust, $\dot{m}_\mathrm{out} = \dot{m}_\mathrm{ox} + \dot{m}_\mathrm{f}$. The oxidizer mass flow rate $\dot{m}_\mathrm{ox}$ is given via $\dot{m}_\mathrm{ox} = \frac{\dot{m}_f}{C_{f/o}\phi}$ with the equivalence ratio ϕ computed from reference mass-flow and oxidizer-flow rates as $\phi = \frac{0.0844}{C_{f/o}}$. We then obtain $\dot{m}_\mathrm{ox} = 11.852\dot{m}_\mathrm{f}$. The outlet mass flow rate follows as $\dot{m}_\mathrm{out}(\boldsymbol{d}) = 12.852\dot{m}_\mathrm{f}$. Recall, that $\dot{m}_\mathrm{f}$ is a design variable.

The $C^\star$ efficiency measure is defined as

$$C^\star(\boldsymbol{d}, \boldsymbol{z}) = \frac{\bar{p}(\boldsymbol{d}, \boldsymbol{z})\ A_\mathrm{t}}{\dot{m}_\mathrm{out}(\boldsymbol{d})},$$

with units of m/s. Here, $A_\mathrm{t} = \pi R_\mathrm{t}^2$ denotes the area of the nozzle throat (see Figure 7 for the nozzle radius R_t) and

$$\bar{p}(\boldsymbol{d}, \boldsymbol{z}) := \frac{1}{L_\mathrm{i} + 2L_\mathrm{bs} + L_\mathrm{c} + L_\mathrm{n}} \int_{-L_\mathrm{i}}^{2L_\mathrm{bs}+L_\mathrm{c}+L_\mathrm{n}} p(x; \boldsymbol{d}, \boldsymbol{z})\mathrm{d}x$$

is the spatial mean of the steady-state pressure. We then define the quantity of interest $f : \mathcal{D} \times \Omega \mapsto \mathbb{R}$ as

$$f(\boldsymbol{d}, \boldsymbol{z}) = -C^{\star}(\boldsymbol{d}, \boldsymbol{z}),$$

and recall that the RBDO objective is to minimize the cost function from Eq.(1).

5.2.2. Reliability constraint

The reliability constraint is based on the maximum pressure, as engines are unsafe if the maximum chamber pressure exceeds a certain threshold. Here, we limit the pressure deviation in the engine relative to the inflow pressure to 13.5% to define failure, i.e., the engine is safe if $\max_x \left[\frac{p(x;\boldsymbol{d},\boldsymbol{z})-p_\infty}{p_\infty}\right] < 0.135$. The limit state function $g : \mathcal{D} \times \Omega \mapsto \mathbb{R}$ for this example is

$$g(\boldsymbol{d}, \boldsymbol{z}) = 0.135 - \max_x \left[\frac{p(x; \boldsymbol{d}, \boldsymbol{z}) - p_\infty}{p_\infty}\right],$$

where the pressure $p(x; \boldsymbol{d}, \boldsymbol{z})$ is computed by solving Eqs. (20)–(21) for design $\boldsymbol{d}$ and with a realization $\boldsymbol{z}$ of the random variable. Note that p_∞ is an uncertain variable, defined in Section 5.1.4. Failure of the system is defined by $g(\boldsymbol{d}, \boldsymbol{z}) < 0$. Recall from Eq. (1) that the reliability constraint is $\mathbb{P}(g(\boldsymbol{d}, Z) < 0) \leq P_{\text{thresh}}$. For the CVRC application, the threshold on the reliability constraint is set at $P_{\text{thresh}} = 0.005$ with error tolerance $\epsilon_{\text{tol}} = 0.05$ in Eq. (16).

5.3. Results of RBDO

We use the `fmincon` optimizer in MATLAB to run the optimization. `fmincon` is a gradient-based optimizer that uses finite difference to estimate the gradients. The maximum number of samples allowed in each optimization iteration for estimating the probability of failure set to $m_{\max} = 10^4$. Note that in this case, the $m_{\max}$ value is governed by cost of evaluation of the computational model. IRIS-RBDO convergence history in Figure 8 (a) shows that it requires more than 2.5×10^4 samples before the optimizer finds the first feasible design. The probability of failure history in Figure 8 (c) shows the progress of designs from infeasible to feasible regions during the optimization of the combustion engine. The best design obtained through RBDO is given in Table 3 and the optimal mean C^* efficiency obtained is 1426.2 m/s.

Figure 9 (a) shows the number of samples used in each optimization iteration using IRIS-RBDO compared to importance sampling with no reuse. Note that the comparison is shown for the same designs in each optimization iteration so that we can make a direct comparison of the computational efficiency. We can see that when biasing density was built with no reuse (here, MPP-based), the number of required samples reached the maximum of 10^4 in most of the optimization iterations. There were only six cases for IRIS-RBDO that reached the maximum number of samples. In this case, IRIS-RBDO leads to overall computational saving of around 76% compared to importance sampling with no reuse. The efficiency of IRIS-RBDO can also be seen from Figure 9 (b) that shows the required number of samples for corresponding probability of failure estimates. We can see that specifically for low probabilities of failure, the required number of samples are substantially lower when compared to building biasing densities with no reuse.

Figure 9 (c) shows that the coefficient of variation (error) in probability of failure estimate for IRIS-RBDO is below the set tolerance ($\epsilon_{\text{tol}} = 0.05$) for all but six optimization iterations. All of the cases where the error tolerance was not met for IRIS-RBDO occurred because for these cases the required number of samples reached $m_{\max}$, which is set to 10^4 (as seen from Figure 9 (a)). These

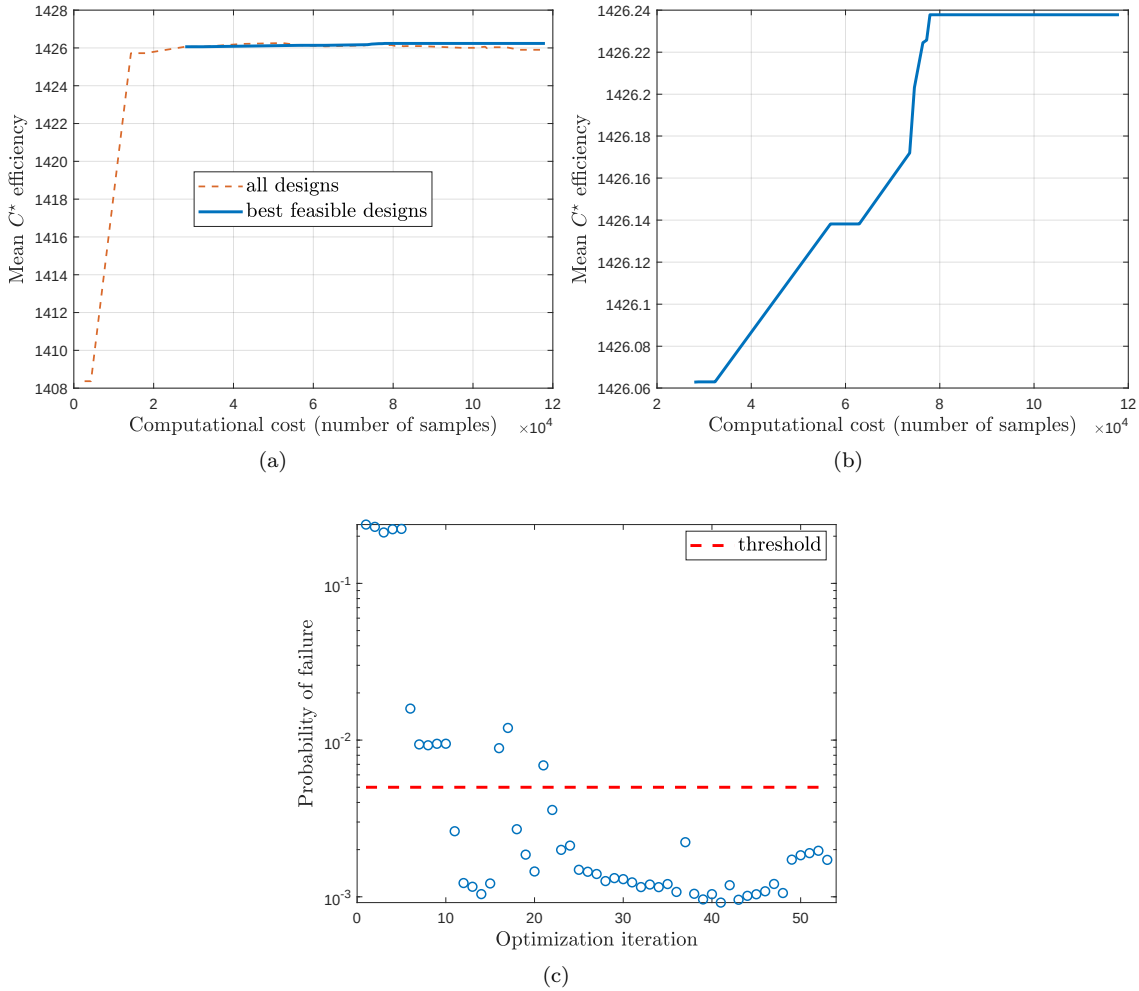


Figure 8: Optimization progress using IRIS-RBDO showing (a) convergence history of mean C^* values for designs from all optimization iterations, (b) magnified convergence plot of feasible designs vs the cumulative computational cost in terms of number of samples, and (c) probability of failure history in each optimization iteration for the combustion engine problem.

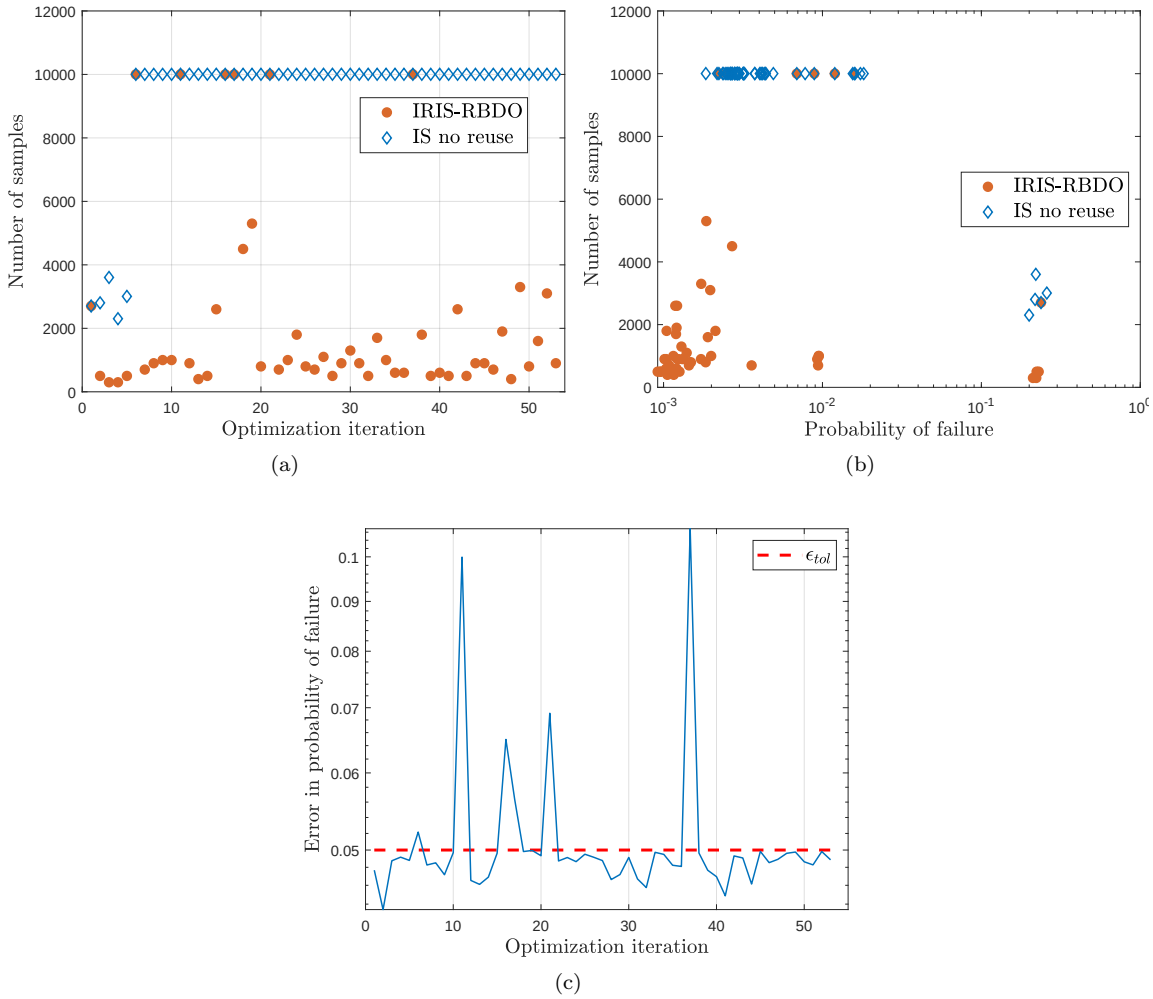


Figure 9: (a) Comparison of IRIS and importance sampling (IS) with no reuse for the same designs showing number of samples required in each optimization iteration, (b) number of samples required to calculate the corresponding probabilities of failure, and (c) the error in probability of failure estimate (quantified by the coefficient of variation) using IRIS in each optimization iteration for the combustion engine problem.

were also the same cases where there were no nearby designs (as seen from Figure 9 (a)) which mean that even IRIS builds the biasing density with no information reuse (here, MPP-based).

The number of designs reused in each optimization iteration by IRIS-RBDO is shown in Figure 10. We can see that all six cases of IRIS-RBDO that required 10^4 samples were cases where no nearby designs were available, i.e., no information was reused. However, we can see that for most of the cases where information was reused, the required number of samples was lower compared to building biasing densities with no reuse (see Figure 9 (a)). The required number of samples are the same when there are zero reused designs. We can also see the additional advantage of the IRIS method for the gradient-based optimizer using finite difference. As pointed out in Remark 4, after estimating the probability of failure at a particular design, the next n_d probability of failure estimates required for the finite difference estimate of the derivative will always use the IRIS method in the implementation of our method with a gradient-based optimizer. This can be clearly seen from the first $n_d + 1$ (here, five) optimization iterations shown in Figure 10, where designs are reused after the first iteration. The efficiency of using the IRIS method is reflected by concurrently looking at the number of samples required by IRIS in the first n_d+1 optimization iterations in Figure 9 (a). Note that in this case, an optimization iteration refers to either a probability of failure estimate at a given design or the probability of failure estimates required for the gradient.

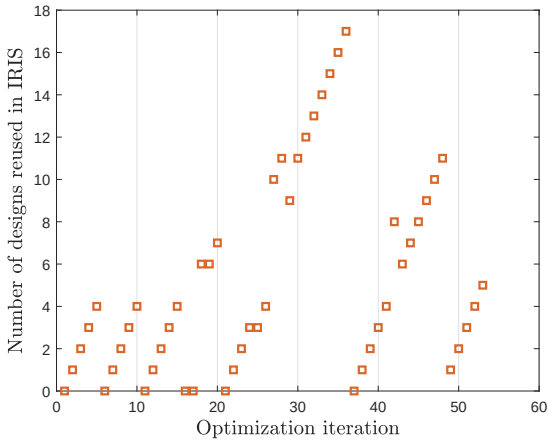


Figure 10: Number of designs reused in IRIS in each optimization iteration for the combustion engine problem.

6. Concluding remarks

This paper introduced IRIS-RBDO (Information Reuse for Importance Sampling in RBDO), a new importance-sampling-based RBDO method. IRIS-RBDO reuses information from past optimization iterations for computationally efficient reliability estimates. The method achieves this by building efficient biasing distributions through two levels of information reuse: (1) reusing the current batch of samples to build an *a posteriori* biasing density with optimal parameters for all designs, and (2) reusing a mixture of the *a posteriori* biasing densities from nearby past designs to build biasing density for the current design. The rich source of existing information from past RBDO iterations helps in constructing very efficient biasing densities. The method can also overcome bad initial biasing densities and there is no bias in the reliability estimates. We show the

efficiency of IRIS-RBDO through a benchmark speed reducer problem and a combustion engine problem. IRIS-RBDO leads to computational savings of around 51% for the speed reducer problem and around 76% for the combustion engine problem as compared to building biasing densities with no reuse (using MPP in this case). In this work, we develop the information reuse idea for importance sampling in RBDO but the method can be easily extended to building initial biasing densities for adaptive importance sampling schemes used in the RBDO setup, and we will explore this in a future work.

Acknowledgements

This work has been supported in part by the Air Force Office of Scientific Research (AFOSR) MURI on managing multiple information sources of multi-physics systems, Award Numbers FA9550-15-1-0038 and FA9550-18-1-0023 and by the Air Force Center of Excellence on Multi-Fidelity Modeling of Rocket Combustor Dynamics under award FA9550-17-1-0195. The authors thank Dr. Cheng Huang for valuable discussions relating to the CVRC design problem, and Dr. Alexandre Marques and Jiayang Xu for making this solver for the CVRC available.

Appendix A. Proof for Theorem 1

The proof follows from similar work in the cross-entropy method [24], where KL divergence is applied in a different context than in this RBDO work. In this section, we derive the analytic solution for the parameters of multivariate normal density, which is chosen to be the distribution in this work. Consider the multivariate normal density with parameters $\theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$:

$$q_\theta(\boldsymbol{z}) = \frac{1}{\sqrt{(2\pi)^{n_r}|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right), \tag{A.1}$$

Taking the logarithm of $q_\theta(\boldsymbol{z})$, we get

$$\ln(q_\theta(\boldsymbol{z})) = -\frac{n_r}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}). \tag{A.2}$$

Then the objective function of the optimization problem given by Eq. (11) at iteration t can be rewritten as

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) =& -\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}_i') \frac{p(\boldsymbol{z}_i')}{q_{\theta_t}(\boldsymbol{z}_i')} \left[-\frac{n_r}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{z}_i'-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i'-\boldsymbol{\mu})\right] \\
=& \frac{n_r}{2}\ln(2\pi)\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}_i') \frac{p(\boldsymbol{z}_i')}{q_{\theta_t}(\boldsymbol{z}_i')} + \frac{1}{2}\ln|\boldsymbol{\Sigma}| \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}_i') \frac{p(\boldsymbol{z}_i')}{q_{\theta_t}(\boldsymbol{z}_i')} \\
&+ \frac{1}{2}\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}_i') \frac{p(\boldsymbol{z}_i')}{q_{\theta_t}(\boldsymbol{z}_i')} (\boldsymbol{z}_i'-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i'-\boldsymbol{\mu}),
\end{aligned} \tag{A.3}$$

where $\boldsymbol{z}_i' \sim q_{\theta_t}$.

The local optimum of Eq. (11) given by parameters $\theta_t^* = \{\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\}$ for RBDO iteration t can be found by equating the gradients of Eq. (A.3) to zero (Karush-Kuhn-Tucker (KKT) conditions).

The local optimum $\boldsymbol{\mu}_t$ is found by setting the gradient of $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ w.r.t. $\boldsymbol{\mu}$ to zero as given by

$$\nabla_{\boldsymbol{\mu}} \mathcal{L} = -\frac{1}{2} \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} \left(2\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}'_i - \boldsymbol{\mu})\right) = 0, \tag{A.4}$$

which then leads to the solution for the parameter $\boldsymbol{\mu}_t$ as given by

$$\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} (\boldsymbol{z}'_i - \boldsymbol{\mu}_t) = 0 \quad \Rightarrow \quad \boldsymbol{\mu}_t = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} \boldsymbol{z}'_i}{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}}. \tag{A.5}$$

We used the fact that $\boldsymbol{\Sigma}$ is symmetric positive definite to get the derivative in Eq. (A.4). The expression given by Eq. (12) can then be derived by writing out each entry of the vector in Eq. (A.5) as given by

$$\mu_t^j = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} z'_{i,j}}{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}} = \frac{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} z'_{i,j}}{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}}. \tag{A.6}$$

Since in Eq. (12), the indicator function $\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) = 1$ only for the failed samples $\boldsymbol{z}'_i \in \mathcal{G}_t$, the indicator function can be removed by taking the sum over the failed samples.

In order to show that Eq. (12) is the global minimum, we take the second-order partial derivative of $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ w.r.t. $\boldsymbol{\mu}$ as given by

$$\nabla^2_{\boldsymbol{\mu}} \mathcal{L} = \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} \boldsymbol{\Sigma}^{-1}. \tag{A.7}$$

We get convexity in $\boldsymbol{\mu}$ because $\nabla^2_{\boldsymbol{\mu}} \mathcal{L}$ is positive definite. We know that the local minimum in convex optimization must also be the global minimum [61]. Thus, Eq. (12) is the global optimum for $\boldsymbol{\mu}$.

In order to derive the local optimum $\boldsymbol{\Sigma}_t$, we rewrite $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ using traces due to its usefulness in calculating derivatives of quadratic form. Note that $(\boldsymbol{z}'_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{z}'_i - \boldsymbol{\mu})$ is a scalar and thus is equal to its trace, $\mathrm{tr}((\boldsymbol{z}'_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{z}'_i - \boldsymbol{\mu}))$. Since the trace is invariant under cyclic permutations, we have

$$\mathrm{tr}((\boldsymbol{z}'_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{z}'_i - \boldsymbol{\mu})) = \mathrm{tr}((\boldsymbol{z}'_i - \boldsymbol{\mu})(\boldsymbol{z}'_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}). \tag{A.8}$$

We can take the derivative of the above expression w.r.t. the matrix $\boldsymbol{\Sigma}^{-1}$ to get

$$\nabla_{\boldsymbol{\Sigma}^{-1}} (\mathrm{tr}((\boldsymbol{z}'_i - \boldsymbol{\mu})(\boldsymbol{z}'_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1})) = (\boldsymbol{z}'_i - \boldsymbol{\mu})^\top (\boldsymbol{z}'_i - \boldsymbol{\mu}). \tag{A.9}$$

Also note that since $\boldsymbol{\Sigma}^{-1}$ is a symmetric positive definite matrix, we have

$$\nabla_{\boldsymbol{\Sigma}^{-1}} \ln|\boldsymbol{\Sigma}^{-1}| = \frac{1}{|\boldsymbol{\Sigma}^{-1}|} |\boldsymbol{\Sigma}^{-1}| \boldsymbol{\Sigma}^\top = \boldsymbol{\Sigma}. \tag{A.10}$$

Using Eq. (A.8) and the fact that the determinant of the inverse of a matrix is the inverse of the determinant, $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be rewritten as

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}) = -\ln|\boldsymbol{\Sigma}^{-1}| \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} + \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} \mathrm{tr}((\boldsymbol{z}'_i - \boldsymbol{\mu})(\boldsymbol{z}'_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}). \tag{A.11}$$

We can substitute the optimum value $\boldsymbol{\mu}_t$ given by Eq. (A.5) in Eq. (A.11) to get

$$\mathcal{L}(\boldsymbol{\Sigma}^{-1}) = -\ln|\boldsymbol{\Sigma}^{-1}| \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} + \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} \text{tr}((\boldsymbol{z}'_i - \boldsymbol{\mu}_t)(\boldsymbol{z}'_i - \boldsymbol{\mu}_t)^\top \boldsymbol{\Sigma}^{-1}). \tag{A.12}$$

The local optimum $\boldsymbol{\Sigma}_t$ is found by taking the gradient of $\mathcal{L}(\boldsymbol{\Sigma})$ w.r.t. the matrix $\boldsymbol{\Sigma}^{-1}$ using the properties described in Eqs. (A.9) and (A.10), and equating it to zero, as given by

$$\nabla_{\boldsymbol{\Sigma}^{-1}} \mathcal{L} = -\boldsymbol{\Sigma} \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} + \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} (\boldsymbol{z}'_i - \boldsymbol{\mu}_t)^\top (\boldsymbol{z}'_i - \boldsymbol{\mu}_t) = 0, \tag{A.13}$$

which then yields

$$\boldsymbol{\Sigma}_t = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} (\boldsymbol{z}'_i - \boldsymbol{\mu}_t)^\top (\boldsymbol{z}'_i - \boldsymbol{\mu}_t)}{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}}. \tag{A.14}$$

The expression given by Eq. (13) can be derived by writing out each entry of the matrix in Eq. (A.14) to get

$$\Sigma_t^{j,k} = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} (z'_{i,j} - \mu_t^j)(z'_{i,k} - \mu_t^k)}{\sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}} = \frac{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)} (z'_{i,j} - \mu_t^j)(z'_{i,k} - \mu_t^k)}{\sum_{i=1}^{|\mathcal{G}_t|} \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}}.$$

As noted before, the indicator function $\mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) = 1$ only for the failed samples $\boldsymbol{z}'_i \in \mathcal{G}_t$ and can be removed by taking the sum over the failed samples in Eq. (13).

In order to show that Eq. (13) is the global minimum, we take the second-order derivative of $\mathcal{L}(\boldsymbol{\Sigma}^{-1})$ w.r.t. $\boldsymbol{\Sigma}^{-1}$ as given by

$$\nabla^2_{\boldsymbol{\Sigma}^{-1}} \mathcal{L} = \boldsymbol{\Sigma}^2 \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(\boldsymbol{d}_t, \boldsymbol{z}'_i) \frac{p(\boldsymbol{z}'_i)}{q_{\theta_t}(\boldsymbol{z}'_i)}. \tag{A.15}$$

We get convexity in $\boldsymbol{\Sigma}^{-1}$ because $\nabla^2_{\boldsymbol{\Sigma}^{-1}} \mathcal{L}$ is positive definite. Thus, Eq. (13) is the global optimum for $\boldsymbol{\Sigma}$ [61].

Appendix B. Most Probable Failure Point

The point with the maximum likelihood of failure is called the most probable failure point (MPP), see [18, 62] for further reading. Typically, this is found by mapping $Z \sim p$ to the standard normal space $U \sim \mathcal{N}(\mathbf{0}, \text{diag}(\mathbf{1})) \in \mathbb{R}^{n_r}$. Let the mapping be done by using some transformation $\boldsymbol{u} = T[\boldsymbol{z}]$. Then the MPP can be found by minimizing the distance from the mean to the limit state failure boundary $g(\boldsymbol{z}) = 0$ in the standard normal space. The optimization problem used to find the MPP is given by

$$\begin{aligned} &\min_{\boldsymbol{u} \in \mathbb{R}^{n_r}} \|\boldsymbol{u}\|_2 \\ &\text{subject to} \quad g(T^{-1}[\boldsymbol{u}]) = 0. \end{aligned} \tag{B.1}$$

Appendix C. Defensive importance sampling

While exploring the design space, the system can have small and large failure probabilities. For small failure probabilities, importance sampling with q is an efficient sampling scheme. For large failure probabilities, standard sampling from the nominal density p leads to good convergence of the estimate. *Defensive importance sampling* [63] proposes to sample from the mixed biasing density

$$q^\alpha := (1-\alpha)q + \alpha p.$$

In [63] it is suggested to use $0.1 \leq \alpha < 0.5$. However, this is for computing small failure probabilities only. An adaptive approach to choose α can be used to account for both rare and common events. Algorithm 3 describes one such adaptive method where we start with $\alpha = 1$ and sample the mean. Then decrease α if the mean has not converged (which is often the case in small failure probabilities), effectively sampling more from the biasing density.

Combining defensive importance sampling with IRIS, the information reuse biasing density with defensive importance sampling is given by

$$q^\alpha_{\theta_t} := (1-\alpha)\left(\sum_{i=0}^{t-1} \beta_i q_{\theta^*_i}\right) + \alpha p.$$

Algorithm 3 Adaptive defensive importance sampling

Input: Nominal density p, biasing density (can be mixture) q, design $\boldsymbol{d}_t$.
Output: Adaptive mixture density q^α.

procedure AdaptiveISDensity(p, q, P_{thresh})
  $\alpha_0 = 1,\ k = 1$;
  while $\widehat{P}^{\text{IS}}(\boldsymbol{d}_t)$ not converged **do**
    $q^\alpha_{\theta_t} := (1-\alpha)q + \alpha p$
    $m_k = kP^{-1}_{\text{thresh}}$ (start with a batch that would get $1/P^{-1}_{\text{thresh}}$) samples
    Compute $\widehat{P}^{\text{IS}}(\boldsymbol{d}_t)$ with samples from $q^\alpha_{\theta_t}$.
    Assign $\alpha = \frac{|\text{\# of failed samples }|}{m_k}$ (i.e., if all samples fail high with nominal density (=high-FP), no need to use IS.))
    $k = k + 1$.
  end while
  return $\widehat{P}^{\text{IS}}_{q^\alpha_{\theta_t}}(\boldsymbol{d}_t)$
end procedure

References

[1] N. Kuschel, R. Rackwitz, Two basic problems in reliability-based structural optimization, Mathematical Methods of Operations Research 46 (3) (1997) 309–333.

[2] R. Yang, L. Gu, Experience with approximate reliability-based optimization methods, Structural and Multidisciplinary Optimization 26 (1-2) (2004) 152–159.

[3] J. Liang, Z. P. Mourelatos, E. Nikolaidis, A single-loop approach for system reliability-based design optimization, Journal of Mechanical Design 129 (12) (2007) 1215–1224.

[4] J. Liang, Z. P. Mourelatos, J. Tu, A single-loop method for reliability-based design optimisation, International Journal of Product Development 5 (1-2) (2008) 76–92.

[5] X. Du, W. Chen, Sequential optimization and reliability assessment method for efficient probabilistic design, in: ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2002, pp. 871–880.

[6] A. Sopory, S. Mahadevan, Z. P. Mourelatos, J. Tu, Decoupled and single loop methods for reliability-based optimization and robust design, in: ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2004, pp. 719–727.

[7] W. Yao, X. Chen, W. Luo, M. van Tooren, J. Guo, Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles, Progress in Aerospace Sciences 47 (6) (2011) 450–479.

[8] M. A. Valdebenito, G. I. Schuëller, A survey on approaches for reliability-based optimization, Structural and Multidisciplinary Optimization 42 (5) (2010) 645–663.

[9] Y. Aoues, A. Chateauneuf, Benchmark study of numerical methods for reliability-based design optimization, Structural and multidisciplinary optimization 41 (2) (2010) 277–294.

[10] R. Rackwitz, Reliability analysis–a review and some perspectives, Structural Safety 23 (4) (2001) 365–395.

[11] M. Hohenbichler, S. Gollwitzer, W. Kruse, R. Rackwitz, New light on first-and second-order reliability methods, Structural Safety 4 (4) (1987) 267–284.

[12] V. Dubourg, B. Sudret, J.-M. Bourinet, Reliability-based design optimization using kriging surrogates and subset simulation, Structural and Multidisciplinary Optimization 44 (5) (2011) 673–690.

[13] B. J. Bichon, J. M. McFarland, S. Mahadevan, Efficient surrogate models for reliability analysis of systems with multiple failure modes, Reliability Engineering & System Safety 96 (10) (2011) 1386–1395.

[14] B. J. Bichon, M. S. Eldred, S. Mahadevan, J. M. McFarland, Efficient global surrogate modeling for reliability-based design optimization, Journal of Mechanical Design 135 (1) (2013) 011009.

[15] X. Qu, R. Haftka, Reliability-based design optimization using probabilistic sufficiency factor, Structural and Multidisciplinary Optimization 27 (5) (2004) 314–325.

[16] M. Moustapha, B. Sudret, J.-M. Bourinet, B. Guillaume, Quantile-based optimization under uncertainties using adaptive kriging surrogate models, Structural and Multidisciplinary Optimization 54 (6) (2016) 1403–1421.

[17] M. Moustapha, B. Sudret, Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework, Structural and Multidisciplinary Optimization (2019) 1–20.

[18] R. Melchers, Importance sampling in structural systems, Structural Safety 6 (1) (1989) 3–10.

[19] J. S. Liu, Monte Carlo strategies in scientific computing, Springer Science & Business Media, 2008.

[20] R. T. Rockafellar, J. O. Royset, On buffered failure probability in design and optimization of structures, Reliability Engineering & System Safety 95 (5) (2010) 499–510.

[21] M. M. Harajli, R. T. Rockafellar, J. O. Royset, Importance sampling in the evaluation and optimization of buffered failure probability, in: Proceedings of 12th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP12), Vancouver, Canada, July 12-15, 2015, 2015. doi:10.14288/1.0076214.

[22] I. Depina, I. Papaioannou, D. Straub, G. Eiksund, Coupling the cross-entropy with the line sampling method for risk-based design optimization, Structural and Multidisciplinary Optimization 55 (5) (2017) 1589–1612.

[23] M. Heinkenschloss, B. Kramer, T. Takhtaganov, K. Willcox, Conditional-value-at-risk estimation via reduced-order models, SIAM/ASA Journal on Uncertainty Quantification 6 (4) (2018) 1395–1423.

[24] R. Y. Rubinstein, D. P. Kroese, The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning, Springer Science & Business Media, 2013.

[25] D. P. Kroese, R. Y. Rubinstein, P. W. Glynn, The cross-entropy method for estimation, in: Handbook of Statistics, Vol. 31, Elsevier, 2013, pp. 19–34.

[26] Z. Wang, J. Song, Cross-entropy-based adaptive importance sampling using von Mises-Fisher mixture for high dimensional reliability analysis, Structural Safety 59 (2016) 42–52.

[27] J.-M. Cornuet, J.-M. Marin, A. Mira, C. P. Robert, Adaptive multiple importance sampling, Scandinavian Journal of Statistics 39 (4) (2012) 798–812.

[28] N. Kurtz, J. Song, Cross-entropy-based adaptive importance sampling using Gaussian mixture, Structural Safety 42 (2013) 35–44.

[29] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic Engineering Mechanics 16 (4) (2001) 263–277.

[30] I. Papaioannou, W. Betz, K. Zwirglmaier, D. Straub, MCMC algorithms for subset simulation, Probabilistic Engineering Mechanics 41 (2015) 89–103.

[31] S.-K. Au, On MCMC algorithm for subset simulation, Probabilistic Engineering Mechanics 43 (2016) 117–120.

[32] J. Li, J. Li, D. Xiu, An efficient surrogate-based method for computing rare failure probability, Journal of Computational Physics 230 (24) (2011) 8683–8697.

[33] V. Dubourg, B. Sudret, F. Deheeger, Metamodel-based importance sampling for structural reliability analysis, Probabilistic Engineering Mechanics 33 (2013) 47–57.

[34] B. Peherstorfer, B. Kramer, K. Willcox, Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models, Journal of Computational Physics 341 (2017) 61–75. doi:https://doi.org/10.1016/j.jcp.2017.04.012.

[35] B. Peherstorfer, B. Kramer, K. Willcox, Multifidelity preconditioning of the cross-entropy method for rare event simulation and failure probability estimation, SIAM/ASA Journal on Uncertainty Quantification 6 (2) (2018) 737–761.

[36] B. Kramer, A. Marques, B. Peherstorfer, U. Villa, K. Willcox, Multifidelity probability estimation via fusion of estimators, Journal of Computational Physics 392 (2019) 385–402. URL https://doi.org/10.1016/j.jcp.2019.04.071

[37] L. W.-T. Ng, Multidelity approaches for design under uncertainty, Ph.D. thesis, Massachusetts Institute of Technology (2013).

[38] L. W.-T. Ng, K. E. Willcox, Multifidelity approaches for optimization under uncertainty, International Journal for Numerical Methods in Engineering 100 (10) (2014) 746–772.

[39] L. W.-T. Ng, K. E. Willcox, Montecarlo information-reuse approach to aircraft conceptual design optimization under uncertainty, Journal of Aircraft 53 (2) (2016) 427–438. doi:10.2514/1.C033352.

[40] L. W. Cook, J. P. Jarrett, K. E. Willcox, Generalized information reuse for optimization under uncertainty with non-sample average estimators, International Journal for Numerical Methods in Engineering 115 (12) (2018) 1457–1476.

[41] G. Zhang, E. Nikolaidis, Z. P. Mourelatos, An efficient re-analysis methodology for probabilistic vibration of large-scale structures, Journal of Mechanical Design 131 (5) (2009) 051007.

[42] A. Chaudhuri, G. Waycaster, N. Price, T. Matsumura, R. T. Haftka, NASA uncertainty quantification challenge: an optimization-based methodology and validation, Journal of Aerospace Information Systems 12 (1) (2015) 10–34.

[43] R. C. Kuczera, Z. P. Mourelatos, E. Nikolaidis, J. Li, A simulation-based RBDO method using probabilistic re-analysis and a trust region approach, in: ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 2009, pp. 1149–1159.

[44] P. Beaurepaire, H. Jensen, G. Schuëller, M. Valdebenito, Reliability-based optimization using bridge importance sampling, Probabilistic Engineering Mechanics 34 (2013) 48–57.

[45] A. B. Owen, Monte Carlo theory, methods and examples, 2013.

[46] S. Kullback, R. A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86.

[47] R. Rubinstein, G. Samorodnitsky, Variance reduction by the use of common and antithetic random variables, Journal of Statistical Computation and Simulation 22 (2) (1985) 161–180.

[48] P. Glasserman, D. D. Yao, Some guidelines and guarantees for common random numbers, Management Science 38 (6) (1992) 884–908.

[49] K. Marti, Differentiation of probability functions: The transformation method, Computers & Mathematics with Applications 30 (3-6) (1995) 361–382.

[50] H. Jensen, M. Valdebenito, G. Schuëller, D. Kusanovic, Reliability-based optimization of stochastic systems using line search, Computer Methods in Applied Mechanics and Engineering 198 (49-52) (2009) 3915–3924.

[51] S. Lacaze, L. Brevault, S. Missoum, M. Balesdent, Probability of failure sensitivity with respect to decision variables, Structural and Multidisciplinary Optimization 52 (2) (2015) 375–381.

[52] J. Sobieszczanski-Sobieski, A. Morris, M. van Tooren, Multidisciplinary design optimization supported by knowledge based engineering, John Wiley & Sons, 2015.

[53] SpaceX to begin testing on reusable Falcon 9 technology this year, https://www.nasaspaceflight.com/2012/01/spacex-testing-reusable-falcon-9-technology-this-year/, accessed: 2018-11-14 (2012).

[54] K. Miller, J. Sisco, N. Nugent, W. Anderson, Combustion instability with a single-element swirl injector, Journal of Propulsion and Power 23 (5) (2007) 1102–1112.

[55] Y. Yu, J. C. Sisco, S. Rosen, A. Madhav, W. E. Anderson, Spontaneous longitudinal combustion instability in a continuously-variable resonance combustor, Journal of Propulsion and Power 28 (5) (2012) 876–887.

[56] T. Feldman, M. Harvazinski, C. Merkle, W. Anderson, Comparison between simulation and measurement of self-excited combustion instability, in: 48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, 2012, p. 4088.

[57] R. Smith, M. Ellis, G. Xia, V. Sankaran, W. Anderson, C. Merkle, Computational investigation of acoustics and instabilities in a longitudinal-mode rocket combustor, AIAA Journal 46 (11) (2008) 2659–2673.

[58] R. Smith, G. Xia, W. Anderson, C. Merkle, Computational studies of the effects of oxidiser injector length on combustion instability, Combustion Theory and Modelling 16 (2) (2012) 341–368.

[59] M. L. Frezzotti, F. Nasuti, C. Huang, C. Merkle, W. E. Anderson, Response function modeling in the study of longitudinal combustion instability by a quasi-1d Eulerian solver, in: 51st AIAA/ASME/SAE/ASEE Joint Propulsion Conference, 2015, pp. 27–29.

[60] J. Xu, K. Duraisamy, Reduced-order reconstruction of model rocket combustor flows, in: 53rd AIAA/SAE/ASEE Joint Propulsion Conference, 2017, p. 4918. doi:doi:10.2514/6.2017-4918.

[61] R. T. Rockafellar, Lagrange multipliers and optimality, SIAM Review 35 (2) (1993) 183–238.

[62] X. Du, W. Chen, A most probable point-based method for efficient uncertainty analysis, Journal of Design and Manufacturing Automation 4 (1) (2001) 47–66.

[63] T. Hesterberg, Weighted average importance sampling and defensive mixture distributions, Technometrics 37 (2) (1995) 185–194.