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 \textit{a posteriori} biasing density with optimal parameters, and (2) reusing the \textit{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 from scratch in each iteration.

**Keywords:** Information reuse, importance sampling, biasing density, probability of failure, reliability analysis, optimization under uncertainty, 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), see Ref. [35] for a review, 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.

*Postdoctoral Associate, Department of Aeronautics and Astronautics, anirbanc@mit.edu.
†Postdoctoral Associate, Department of Aeronautics and Astronautics, bokramer@mit.edu.
‡Director, Institute for Computational Engineering and Sciences, kwillcox@ices.utexas.edu
For mildly nonlinear systems, this can be done using first-order and second-order reliability methods \[27, 17, 34\], while nonlinear systems typically require Monte Carlo sampling. 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.

![Diagram](image)

**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. [9] proposed refining kriging surrogates using a population-based adaptive sampling technique through subset simulation for RBDO. Bichon et al. [3, 4, 2] combined adaptive Gaussian-process-based global reliability analysis with efficient global optimization (a.k.a. Bayesian optimization) for RBDO. However, those methods do not reuse information from previous design iterations, which is a source of computational savings that we explore in this work.

Another approach is to use importance sampling, where many fewer samples are needed to estimate the reliability (or failure probability) using a biasing density. Importance sampling is, in general, an efficient method for reliable systems, e.g., for small failure probabilities \[19, 11, 25, 26\] or other measures of risk \[29, 13, 7, 15\]. 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. As is well known, finding a good biasing density in importance sampling is challenging.

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 **R**BDO) 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, we reuse the current batch of samples from the reliability estimate to form an *a posteriori* biasing...
density. The \textit{a posteriori} biasing densities are built in an optimal way by minimizing the Kullback-Leibler (KL) divergence measure to the optimal (zero-variance) biasing density.

2. We reuse the available \textit{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 \textit{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 that is typically left unused. Reusing information from past optimization iterations in optimization under uncertainty has been previously done in the context of robust optimization using control variates \cite{22, 23}. Cook et al. \cite{5} extended the control variates method for information reuse to a larger class of estimators for robust optimization.

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. Second, the method can overcome bad initial biasing densities by reusing samples to build (at every design iteration) a \textit{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. Fourth, there is no bias in the IRIS-RBDO reliability analysis. In contrast, sampling directly from surrogate models—while being computationally cheaper—introduces such a bias.

The rest of the paper is structured as follows. Section \textcolor{blue}{2} provides the RBDO formulation and background on methods used to estimate the probability of failure. Section \textcolor{blue}{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 \textcolor{blue}{4} and a combustion engine model in Section \textcolor{blue}{5}. Section \textcolor{blue}{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 \(d \in D \subseteq \mathbb{R}^{n_d}\) and an \(n_r\)-dimensional random variable \(Z : \Xi \rightarrow \Omega \subseteq \mathbb{R}^{n_r}\) defined on the sample space \(\Xi\) and with the probability density function \(p\), henceforth called nominal density. Here, \(D\) denotes the design space and \(\Omega\) denotes the random sample space. A realization of \(Z\) is denoted as \(z \in \Omega\). We use \(z \sim p\) 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

\[
\min_{d \in D} J(d) = \mathbb{E}_p[f(d, Z)]
\]

subject to \(\mathbb{P}(g(d, Z) > 0) \leq P_{\text{thresh}}\), \hspace{1cm} (1)
The Monte Carlo estimate of the probability of failure $P$ defined as $m$ sampling would be computationally infeasible for expensive-to-evaluate limit state functions. Due to the low probability of failure for reliable designs, standard Monte Carlo to achieve a fixed level of accuracy in the probability estimate scales inversely with the probability itself. Let $d_t$ be the design in optimization iteration $t$ and define the corresponding failure set as

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

We emphasize that evaluating the limit state function, $g$, and hence checking if $z \in \mathcal{G}_t$, 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

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

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

$$\hat{P}^{MC}_P(d_t) = \frac{1}{m_t} \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(d_t, z_i), \quad z_i \sim p, \tag{4}$$

where $z_i$, $i = 1, \ldots, m_t$ are the $m_t$ samples from probability density $p$ used in iteration $t$. The subscript for $\hat{P}$ 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 $q_{\theta_t}$ for optimization iteration $t$, where $\theta_t$ denotes the parameters of the distribution. The biasing density must satisfy $\text{supp}(\mathbb{I}_{\mathcal{G}_t}(d_t, Z)p(Z)) \subset \text{supp}(q_{\theta_t}(Z))$. The importance sampling estimate for $P(d_t)$ is given by

$$\hat{P}^{IS}_{q_{\theta_t}}(d_t) = \frac{1}{m_t} \sum_{i=1}^{m_t} \mathbb{I}_{\mathcal{G}_t}(d_t, z_i' \mid q_{\theta_t}(z_i'), z_i' \sim q_{\theta_t}, \tag{5}$$

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

The unbiased sample variance $\hat{\sigma}^2_{m_t}$ for the importance sampling estimate is computed as

$$\hat{\sigma}^2_{m_t} = \frac{1}{m_t - 1} \sum_{i=1}^{m_t} \left( \mathbb{I}_{\mathcal{G}_t}(d_t, z_i') \frac{p(z_i')}{q_{\theta_t}(z_i')} - \hat{P}^{IS}_{q_{\theta_t}} \right)^2, \quad z_i' \sim q_{\theta_t}.$$
The relative error, or coefficient-of-variation, in the probability of failure estimate is given by
\[
e(\hat{P}_{IS}) = \frac{1}{\hat{P}_{IS}} \sqrt{\frac{\hat{\sigma}^2_{m_t}}{m_t}}.
\]
(6)

The importance sampling estimate of the failure probability is unbiased, i.e.,
\[
E_p[I_{G_t}(d_t, \cdot)] = E_{q_{\theta_t}}[I_{G_t}(d_t, \cdot) \frac{p}{q_{\theta_t}}].
\]

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.

After evaluating the importance-sampling failure probability estimate \( \hat{P}_{IS}^{q_{\theta_t}}(d_t) \) with density \( q_{\theta_t} \), we can compute an a posteriori biasing density that is close to the optimal (also known as zero-variance) biasing density. It is known [24, Chapter 9] that the theoretical optimal biasing density results in the estimate \( \hat{P}_{IS}^{h^*_t}(d_t) \) having zero variance, and is given by
\[
h^*_t(z) = \frac{I_{G_t}(d_t, z)p(z)}{P(d_t)},
\]
(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 \( P(d_t) \) in Eq. (7), this density is not practical to compute.

Consequently, we want to find an a posteriori biasing density that is close to \( h^*_t \). In this work, the KL divergence [18] is used as the distance measure between two distributions. We thus define a density \( q_{\theta} \) parameterized by \( \theta \in P \) and want to minimize the KL divergence to \( h^*_t \), which is defined as
\[
KL(h^*_t \parallel q_{\theta}) = E_{h_t^*} \left[ \ln \left( \frac{h_t^*}{q_{\theta}} \right) \right] = \int_{-\infty}^{\infty} \ln \left( \frac{h^*_t(z)}{q_{\theta}(z)} \right) h_t^*(z) \, dz.
\]
(8)

The optimal parameters for \( q_{\theta} \) for RBDO iteration \( t \) are given by \( \theta^*_t \), where the superscript denotes that it is the optimal solution. Then \( \theta^*_t \) can be found by solving an optimization problem given by
\[
\theta^*_t = \arg \min_{\theta \in P} KL(h^*_t \parallel q_{\theta}) = \arg \min_{\theta \in P} E_{h_t^*} \left[ \ln \left( \frac{h_t^*}{q_{\theta}} \right) \right]\]
\[
= \arg \min_{\theta \in P} \int_{z \in \Omega} \ln(h^*_t(z))h_t^*(z) \, dz - \int_{z \in \Omega} \ln(q_{\theta}(z))h_t^*(z) \, dz
\]
\[
= \arg \max_{\theta \in P} \int_{z \in \Omega} \ln(q_{\theta}(z))h_t^*(z) \, dz = \arg \max_{\theta \in P} E_{h_t^*} [\ln(q_{\theta})]
\]
\[
= \arg \max_{\theta \in P} E_p \left[ I_{G_t}(d_t, \cdot) \ln(q_{\theta}) \right],
\]
(9)
where in the last step we used the definition of the optimal biasing density from Eq. (7), and dropped the term $P(d_t)$ as it does not affect the optimization. Since the integral requires evaluating the failure region, we use again importance sampling with density $q_{th}$ to obtain an efficient estimate, i.e.,

$$
\mathbb{E}_P [\mathbb{I}_{G_t}(d_t, \cdot) \ln(q_{th})] = \mathbb{E}_{q_{th}} \left[ \mathbb{I}_{G_t}(d_t, \cdot) \frac{P}{q_{th}} \ln(q_{th}) \right].
$$

(10)

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

$$
\theta^*_t = \arg \max_{\theta \in \mathcal{P}} \mathbb{E}_{q_{th}} \left[ \mathbb{I}_{G_t}(d_t, \cdot) \frac{P}{q_{th}} \ln(q_{th}) \right] \approx \arg \max_{\theta \in \mathcal{P}} \sum_{i=1}^{m_t} \mathbb{I}_{G_t}(d_t, z'_i) \frac{p(z'_i)}{q_{th}(z'_i)} \ln(q_{th}(z'_i)).
$$

(11)

where we used numerical integration to replace the expectation and $z'_i$ is sampled from the biasing density $q_{th}$.

We choose a multivariate normal distribution as the parametric distribution for $q_{th}$. 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., $q_{th} \sim \mathcal{N}(\mu_t, \Sigma_t)$, (and for several other parametric distributions, specifically the exponential family of distributions), an analytic solution of the optimal parameters $\theta^*_t$ can be derived and shown to be the global optimum for Eq. (11) as described below.

**Theorem 1.** Let $q_{th} \sim \mathcal{N}(\mu_t, \Sigma_t)$ be a multivariate normal distribution, with the mean vector $\mu_t = [\mu_{t,1}, \ldots, \mu_{t,n_r}]^T$ and $\Sigma_t = [\Sigma_{t,j,k}]_{j,k=1,\ldots,n_r}$ being the symmetric positive definite covariance matrix. Let $z'_i = [z'_{i,1}, \ldots, z'_{i,n_r}]^T \sim q_{th}$ represent the $i$th sample vector and $z'_{i,j}$ represent the $j$th entry of the $i$th sample vector for $j \in \{1, \ldots, n_r\}$. Then the parameters $\theta^*_t = \{\mu_t, \Sigma_t\}$ are given by

$$
\mu^*_t = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{G_t}(d_t, z'_i) \frac{p(z'_i)}{q_{th}(z'_i)} z'_{i,j}}{\sum_{i=1}^{m_t} \mathbb{I}_{G_t}(d_t, z'_i) \frac{p(z'_i)}{q_{th}(z'_i)}} = \sum_{i=1}^{m_t} \mathbb{I}_{G_t}(d_t, z'_i) \frac{p(z'_i)}{q_{th}(z'_i)} z'_{i,j},
$$

$$
\Sigma^*_{t,j,k} = \frac{\sum_{i=1}^{m_t} \mathbb{I}_{G_t}(d_t, z'_i) \frac{p(z'_i)}{q_{th}(z'_i)} (z'_{i,j} - \mu^*_t)(z'_{i,k} - \mu^*_t)}{\sum_{i=1}^{m_t} \mathbb{I}_{G_t}(d_t, z'_i) \frac{p(z'_i)}{q_{th}(z'_i)}} = \sum_{i=1}^{m_t} \mathbb{I}_{G_t}(d_t, z'_i) \frac{p(z'_i)}{q_{th}(z'_i)} (z'_{i,j} - \mu^*_t)(z'_{i,k} - \mu^*_t) .
$$

(12)

(13)

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

**Proof.** See [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 [20], where the biasing density...
is chosen to be the normal distribution with mean shifted to the most-probable-point (MPP, see 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 \( g(z) = z_1 + z_2 - 18 \). Failure is defined as \( g(z) > 0 \). 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 \( \theta^* \) 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 \( 9.7 \times 10^{-3} \). The coefficient of variation of probability of failure estimate using the MPP-based biasing density is 0.0262 and KL-MVN-based biasing density is 0.0064. In both cases, \( 10^4 \) samples are used from the biasing densities. We observe that the KL-MVN biasing density is a better biasing density since it leads to a reduction in the coefficient of variation by around a factor of four compared to using the MPP-based biasing density.

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 \textit{a posteriori} biasing densities corresponding to existing designs $d_0, \ldots, d_{t-1}$ from past optimization iterations that are close in design space.

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

$$\beta_j := \begin{cases} \frac{\|d_j - d_t\|_2^{-1}}{\sum_{d_i \in \mathcal{M}_t} \|d_i - d_t\|_2^{-1}}, & 0 \leq j \leq t-1, \|d_j - d_t\|_2 \leq r \\ 0, & \text{else} \end{cases}$$

(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).

The information reuse biasing density for current design iteration $t$ is defined by the mixture of existing \textit{a posteriori} biasing densities as given by

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

(15)

where $q_{\theta_j^*}$ are all the \textit{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 feasible 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 [20] method (see B on how to compute) for building a biasing density for importance sampling from scratch. However, any other method depending on the user’s preference can be chosen to build the biasing density from scratch. 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 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 $\epsilon_{\text{tol}}$ at step $t$. That is, we require

$$e(\widehat{\beta}_{q_{\theta_t}}) \leq \epsilon_{\text{tol}}.$$  

(16)

The value for $\epsilon_{\text{tol}}$ 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 $m_{\text{max}}$.
that shall not be exceeded at every design iteration. Note that $m_{\text{max}}$ can typically be set by taking into account the values of $P_{\text{thresh}}$ and $\epsilon_{\text{tol}}$.

After estimating $P_{\text{IS}}^{q_0}(d_i)$ within the specified relative error tolerance $\epsilon_{\text{tol}}$, we reuse the existing samples to construct the a posteriori biasing density $q_{\theta_t}^*$ with optimal parameters $\theta_t^*$ using the method described in Section 3.1. The algorithm then proceeds to the next optimization iteration.

**Remark 3.** 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 from scratch, one can use defensive importance sampling (described in [C]) in combination with their method of choice (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.

### 4. Benchmark problem: Speed reducer

The speed reducer problem used in Ref. [32, 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. [32]. 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 in Table 2. We fix the number of gear teeth to 17.

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

<table>
<thead>
<tr>
<th>Lower bound (mm)</th>
<th>Upper bound (mm)</th>
<th>Initial design (mm)</th>
<th>Best design (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>2.6</td>
<td>3.6</td>
<td>3.5</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0.7</td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>$d_3$</td>
<td>7.3</td>
<td>8.3</td>
<td>7.3</td>
</tr>
<tr>
<td>$d_4$</td>
<td>7.3</td>
<td>8.3</td>
<td>7.72</td>
</tr>
<tr>
<td>$d_5$</td>
<td>2.9</td>
<td>3.9</td>
<td>3.35</td>
</tr>
<tr>
<td>$d_6$</td>
<td>5.0</td>
<td>5.5</td>
<td>5.29</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Random variable</th>
<th>Distribution</th>
<th>Mean</th>
<th>Standard deviation ((\mu m))</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z_1$</td>
<td>Normal</td>
<td>$d_2$</td>
<td>1</td>
</tr>
<tr>
<td>$z_2$</td>
<td>Normal</td>
<td>$d_4$</td>
<td>30</td>
</tr>
<tr>
<td>$z_3$</td>
<td>Normal</td>
<td>$d_6$</td>
<td>21</td>
</tr>
</tbody>
</table>
The RBDO problem formulation used in this work is given by

$$
\min_{d \in D} \mathbb{E}_p[f(d, Z)]
$$

subject to \( \mathbb{P}(g_i(d, Z) > 0) \leq P_{\text{thresh}} = 10^{-3}, \ i = 1, 2, 3, \)

where

$$
f(d, z) = 0.7854d_1z_1^2(3.3333 \times 17^2 + 14.9334 \times 17 - 43.0934) - 1.5079d_1(z_3^2 + d_6^2) + 7.477(z_3^2 + d_6^2) + 0.7854(z_2z_3^2 + d_4d_7^2),
$$
Algorithm 2 Probability of failure estimate

Input: Limit state function $g(\cdot)$, design $d$, nominal density $p$, biasing density $q_0$, coefficient of variation tolerance $\epsilon_{tol}$, maximum number of samples $m_{max}$

Output: Probability of failure estimate $\hat{P}^{\text{IS}}_{q_0}(d)$, failure set $G$

1: procedure PROBABILITYOFFAILURE($p, q_0, d, \epsilon_{tol}, m_{max}$)
   2: $m_t = 0$ $\triangleright$ Number of samples
   3: $m_{add} = 100$ $\triangleright$ 100 samples are added at a time
   4: $\epsilon(\hat{P}^{\text{IS}}_{q_0}) = 100\epsilon_{tol}$
   5: $Z'_t = \emptyset$
   6: while ($m_t \leq m_{max}$) or ($\epsilon(\hat{P}^{\text{IS}}_{q_0}) > \epsilon_{tol}$) do
   7: Get $m_{add}$ samples $\{z'_1, \ldots, z'_{m_{add}}\}$ from $q_0$
   8: $m_t \leftarrow m_t + m_{add}$
   9: $Z'_t \leftarrow Z'_t \cup \{z'_1, \ldots, z'_{m_{add}}\}$
   10: Compute probability of failure

   $\hat{P}^{\text{IS}}_{q_0}(d) = \frac{1}{m_t} \sum_{t=1}^{m_t} \mathbb{1}_{G_t}(d, z'_t) \frac{p(z'_t)}{q_0(z'_t)}$

11: Calculate coefficient of variation in probability of failure estimate $\epsilon(\hat{P}^{\text{IS}}_{q_0})$ using Eq. (6)
12: end while
13: $G_t \leftarrow \{z \mid z \in Z'_t, g(d, z) > 0\}$ $\triangleright$ Failure set
14: return $\hat{P}^{\text{IS}}_{q_0}(d), G_t$
15: end procedure

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

\[
g_1(d, z) = \frac{1.93 z_2^3}{17 z_1 z_3^4} - 1, \]
\[
g_2(d, z) = \frac{A_1}{B_1} - 1120, \quad A_1 = \left[\left(\frac{745 z_2}{17 z_1}\right)^2 + 16.9 \times 10^6\right]^{0.5}, \quad B_1 = 0.1 z_3^3, \quad (19)
\]
\[
g_3(d, z) = \frac{A_2}{B_2} - 870, \quad A_2 = \left[\left(\frac{745 d_4}{17 z_1}\right)^2 + 157.5 \times 10^6\right]^{0.5}, \quad B_2 = 0.1 d_3^3.
\]

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 $m_{max} = 5 \times 10^5$.

Figure 4 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 $2 \times 10^5$ samples before it finds the first feasible design. The best design obtained in this case is given in Table 1 which had an associated cost of 3029.2 mm$^3$. The probability of failure history seen in Figure 4(a) shows the progress of designs from infeasible to feasible regions during the optimization. Figure 4(b) shows that the coefficient of variation in probability of failure is below the set tolerance ($\epsilon_{tol} = 0.01$) for every optimization iteration, as required.

The total number of samples used in each optimization iteration in IRIS-RBDO versus RBDO with biasing density built from scratch is shown in Figure 5(a). Note that we are showing the plots for the same designs in each RBDO iteration for both cases, which makes it a one-to-one comparison. When no designs are nearby, our method also builds a biasing density from scratch, hence the two markers overlap in those iterations. Otherwise, IRIS-RBDO always
Figure 3: Objective function history for IRIS-RBDO showing (a) the objective function value for designs from all optimization iterations, and (b) magnified convergence plot of feasible designs against the cumulative computational cost in terms of number of samples for the speed reducer problem.

Figure 4: (a) Probability of failure history, and (b) error in probability of failure estimate (quantified by the coefficient of variation) in each optimization iteration using IRIS-RBDO for the speed reducer problem.
outperforms the other method. IRIS-RBDO leads to overall computational savings of around 51% compared to building biasing densities from scratch throughout the optimization. All optimization iterations required less than the maximum number of samples, $m_{\text{max}}$, which is set to $5 \times 10^5$. Consequently, the estimate always meets the error tolerance as shown in Figure 4(b).

Figure 5: (a) Comparison of number of samples required, and (b) number of designs reused in IRIS in each optimization iteration for the speed reducer problem.

The number of designs reused in each optimization iteration of IRIS-RBDO is shown in Figure 5(b). 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 5(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 density was built from scratch (here, MPP-based) without any information reuse.

Figure 6 compares performance of IRIS-RBDO vs RBDO with biasing density built from scratch by showing the number of samples required for the corresponding probability of failure estimates. For the case when biasing density is built from scratch, 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 from scratch during IRIS-RBDO.

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 [1]. 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
Figure 6: Number of samples required by IRIS to calculate the corresponding probabilities of failure compared to using biasing density $q$ built from scratch for the same designs in each optimization iteration for the speed reducer problem.

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 7. The CVRC is an experiment at Purdue University which has been extensively studied both experimentally [21, 36, 11] and computationally [14, 30, 31, 12].

Figure 7: CVRC experimental configuration from [36]. The computational domain for the reactive flow computations is given in Figure 8.

5.1.1. Governing equations and geometry

The experiment is modeled with a quasi-1D partial differential equation model. Figure 8 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

---

1The 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.
segments, separated by the dashed lines: the injector, back-step, combustion chamber, back-step and nozzle. The length of both backsteps are fixed at 3.81cm and the nozzle length is 0.635cm. The injector radius is given by $R_i$ and the combustion chamber radius is given by $R_c$. The nozzle radii are $R_t = 1.0401$cm at the throat and 1.0922cm at the exit. The quasi-1D

![Figure 8: Computational domain (dashed area in Fig. 7) 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.](image)

Euler partial differential equation model for the CVRC [12, 33] is given as

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

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 $Y_{ox}(x)$ is the oxidizer mass fraction. The state variables are density $\rho(x)$, specific momentum $\rho(x)u(x)$, total energy $E(x)$, and $\rho(x)Y_{ox}(x)$. The equation of state

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

relates energy and pressure $p(x)$ via the heat capacity ratio $\gamma$. In the source terms of Eq. (20), which model the chemical reaction and the cross-sectional area variation, $\Delta h_0$ 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. 8. The forcing function $\dot{\omega}_f$ in Eq. (20) is then modeled as

$$\dot{\omega}_f(x, \dot{m}_f) = \frac{\dot{m}_f}{A(x) \int (1 + \sin(\xi(x))) \, dx} (1 + \sin(\xi(x))),$$

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

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 \( Y_{\text{ox}} \). The inlet stagnation temperature \( T_0 \) is determined as follows: we prescribe a reference temperature \( T_\infty \) and reference pressure \( p_\infty \), which are typically given from upstream components of an engine. We then use the relation

\[
T_0 = T_\infty + \frac{\dot{m}_{\text{ox}} R_\text{gas} T_\infty^2}{p_\infty C_p} \times \frac{1}{2}
\]

with universal gas constant \( R_\text{gas} = 8.314 \times 10^3 \frac{\text{J}}{\text{mol} \cdot \text{K}} \) and specific heat of the fuel \( C_p \).

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 \( d \in \mathcal{D} \subseteq \mathbb{R}^4 \):

- 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 8), and the mass-flow rate \( \dot{m}_f \) that enters into the forcing model in Eq. (22). The design variables \( d = [R_i, R_c, L_f, \dot{m}_f] \) and the respective bounds are given in Table 3.

<table>
<thead>
<tr>
<th>Description</th>
<th>Range</th>
<th>Initial design</th>
<th>Best design</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_i ) (Injector radius)</td>
<td>[0.889, 1.143] cm</td>
<td>1.02</td>
<td>1.14</td>
</tr>
<tr>
<td>( R_c ) (Combustion chamber radius)</td>
<td>[1.778, 2.54] cm</td>
<td>2.16</td>
<td>2.41</td>
</tr>
<tr>
<td>( L_f ) (Location of fuel injection)</td>
<td>[3.5, 4] cm</td>
<td>3.75</td>
<td>3.5</td>
</tr>
<tr>
<td>( \dot{m}_f ) (Mass flow rate for fuel injection)</td>
<td>[0.026, 0.028] kg/s</td>
<td>0.027</td>
<td>0.026</td>
</tr>
</tbody>
</table>

5.1.4. Uncertain variables

The reference pressure \( p_\infty \) and the reference temperature \( T_\infty \) are typically measured from upstream components of the combustion engine and are therefore subject to uncertainty. They enter 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 \( Z \in \Omega \subseteq \mathbb{R}^3 \) with a uniform probability distribution. A realization of the random variable is \( z = [p_\infty, T_\infty, C_{f/o}] \). We list the three uncertain variables and their respective probability distributions in Table 4.

<table>
<thead>
<tr>
<th>Uncertain variable</th>
<th>Description</th>
<th>Distribution</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_\infty )</td>
<td>Upstream pressure</td>
<td>Uniform</td>
<td>[1.3, 1.6] MPa</td>
</tr>
<tr>
<td>( T_\infty )</td>
<td>Upstream oxidizer temperature</td>
<td>Uniform</td>
<td>[1000, 1060] K</td>
</tr>
<tr>
<td>( C_{f/o} )</td>
<td>Fuel-to-oxidizer ratio</td>
<td>Uniform</td>
<td>[0.10, 0.11]</td>
</tr>
</tbody>
</table>
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 \( d = [R_i, R_c, L_f, \dot{m}_f] \) and a realization \( z = [p_\infty, T_\infty, C_{f/o}] \) of the uncertain parameters, i.e., the pressure \( p(x) = p(x; d, z) \). 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 \( C^* \) (“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 \( C^* \), we need the total mass flow rate at the exhaust, \( \dot{m}_{\text{out}} = \dot{m}_{\text{ox}} + \dot{m}_f \). The oxidizer mass flow rate \( \dot{m}_{\text{ox}} \) is given via \( \dot{m}_{\text{ox}} = \dot{m}_f C_{f/o} \phi \) with the equivalence ratio \( \phi \) computed from reference mass-flow and oxidizer-flow rates as \( \phi = \frac{0.0844}{C_{f/o}} \). We then obtain \( \dot{m}_{\text{ox}} = 11.852 \dot{m}_f \). The outlet mass flow rate follows as \( \dot{m}_{\text{out}}(d) = 12.852 \dot{m}_f \). Recall, that \( \dot{m}_f \) is a design variable.

The \( C^* \) efficiency measure is defined as

\[
C^*(d, z) = \frac{\bar{p}(d, z) A_t}{\dot{m}_{\text{out}}(d)},
\]

with units of m/s. Here, \( A_t = \pi R_t^2 \) denotes the area of the nozzle throat (see Figure 8 for the nozzle radius \( R_t \)) and

\[
\bar{p}(d, z) := \int p(x; d, z) dx
\]

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

\[
f(d, z) = -C^*(d, z),
\]

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

\[
\max_x \left[ \frac{p(x; d, z) - p_\infty}{p_\infty} \right] < 0.0135.
\]

The limit state function \( g : \mathcal{D} \times \Omega \mapsto \mathbb{R} \) for this example is

\[
g(d, z) = \max_x \left[ \frac{p(x; d, z) - p_\infty}{p_\infty} \right] - 0.0135,
\]

where the pressure \( p(x; d, z) \) is computed by solving Eqs. (20)–(21) for design \( d \) and with a realization \( z \) of the random variable. Note that \( p_\infty \) is an uncertain variable, defined in Section 5.1.4. Failure of the system is defined by \( g(d, z) > 0 \). Recall from Eq. (1) that the reliability constraint is \( \mathbb{P}(g(d, Z) > 0) \leq P_{\text{thresh}} \). 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 \textit{fmincon} optimizer in MATLAB to run the optimization. The maximum number of samples allowed in each optimization iteration for estimating the probability of failure set to \( m_{\text{max}} = 10^4 \). Note that in this case, the \( m_{\text{max}} \) value is governed by cost of evaluation of the computational model. IRIS-RBDO convergence history in Figure 9 shows that it requires
more than $2.5 \times 10^4$ samples before the optimizer finds the first feasible design. The best design obtained through RBDO is given in Table 3 and the optimal mean $C^*$ efficiency obtained is 1426.2 m/s.

The probability of failure history in Figure 10(a) shows the progress of designs from infeasible to feasible regions during the optimization of the combustion engine. Figure 10(b) shows that the coefficient of variation (error) in probability of failure estimate for IRIS-RBDO is below the set tolerance ($\epsilon_{tol} = 0.05$) 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 11(a)). These were also the same cases where there were no nearby designs (as seen from Figure 11(a)) which mean that even IRIS builds the biasing density from scratch and no information is reused.

Figure 11(a) shows the number of samples used in each optimization iteration using IRIS-RBDO compared to using biasing density built from scratch. 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 from scratch, 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 RBDO with biasing density built from scratch.

The number of designs reused in each optimization iteration by IRIS-RBDO is shown in Figure 11(b). We can see that all six cases of IRIS-RBDO that required $10^4$ samples were cases where no nearby designs were available and the biasing densities were built from scratch, 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 from scratch (see Figure 11(a)). The required number of samples are the same when there are zero reused designs.

The efficiency of IRIS-RBDO can also be seen from Figure 12 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 from scratch.
Figure 10: (a) Probability of failure history, and (b) error in probability of failure estimate (quantified by the coefficient of variation) using IRIS-RBDO for the combustion engine problem.

Figure 11: (a) Comparison of number of samples required, and (b) number of designs reused in IRIS-RBDO 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 from scratch.

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.

A. Proof for Theorem 1

Consider the multivariate normal density with parameters $\theta = \{\mu, \Sigma\}$:

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

(24)
Taking the logarithm of $q_θ(z)$, we get
\[
\ln(q_θ(z)) = -\frac{n_r}{2} \ln(2\pi) - \frac{1}{2} \ln|Σ| - \frac{1}{2} (z - μ)^T Σ^{-1} (z - μ).
\] (25)

Then the objective function of the optimization problem given by Eq. (11) at iteration $t$ can be rewritten as
\[
\mathcal{L}(μ, Σ) = \sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)} \left[ -\frac{n_r}{2} \ln(2\pi) - \frac{1}{2} \ln|Σ| - \frac{1}{2} (z'_i - μ)^T Σ^{-1} (z'_i - μ) \right]
\]
\[
= -\frac{n_r}{2} \ln(2\pi) \sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)} - \frac{1}{2} \ln|Σ| \sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)}
\]  
\[
- \frac{1}{2} \sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)} (z'_i - μ)^T Σ^{-1} (z'_i - μ),
\] (26)

where $z'_i \sim q_θ$.

The local optimum of Eq. (11) given by parameters $θ^*_t = \{μ_t, Σ_t\}$ for RBDO iteration $t$ can be found by equating the gradients of Eq. (26) to zero (Karush-Kuhn-Tucker (KKT) conditions).

The local optimum $μ_t$ is found by setting the gradient of $\mathcal{L}(μ, Σ)$ w.r.t. $μ$ to zero as given by
\[
∇_μ \mathcal{L} = -\sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)} (2Σ^{-1}(z'_i - μ)) = 0,
\] (27)

which then leads to the solution for the parameter $μ_t$ as given by
\[
\sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)} (z'_i - μ_t) = 0 \Rightarrow μ_t = \frac{\sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)} z'_i}{\sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)}}.
\] (28)

We used the fact that $Σ$ is symmetric positive definite to get the derivative in Eq. (27). The expression given by Eq. (12) can then be derived by writing out each entry of the vector in Eq. (28) as given by
\[
μ^j_t = \frac{\sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)} z'_{ij}}{\sum_{i=1}^{m_t} I_g_i(d_t, z'_i) \frac{p(z'_i)}{q_θ(z'_i)}} = \frac{∑[G_i] p(z'_i) z'_{ij}}{∑[G_i] p(z'_i)}.
\]

Since in Eq. (12), the indicator function $I_g_i(d_t, z'_i) = 1$ only for the failed samples $z'_i \in G_i$, the indicator function can be removed by taking the sum over the failed samples.

In order to derive the local optimum $Σ_t$, we rewrite $\mathcal{L}(μ, Σ)$ using traces due to its usefulness in calculating derivatives of quadratic form. Note that $(z'_i - μ)^T Σ^{-1}(z'_i - μ)$ is a scalar and thus is equal to its trace, $tr((z'_i - μ)^T Σ^{-1}(z'_i - μ))$. Since the trace is invariant under cyclic permutations, we have
\[
tr((z'_i - μ)^T Σ^{-1}(z'_i - μ)) = tr((z'_i - μ)(z'_i - μ)^T Σ^{-1}).
\] (30)

We can take the derivative of the above expression w.r.t. the matrix $Σ^{-1}$ to get
\[
∇_Σ^{-1}(tr((z'_i - μ)(z'_i - μ)^T Σ^{-1})) = (z'_i - μ)^T (z'_i - μ).
\] (31)

Also note that since $Σ^{-1}$ is a symmetric positive definite matrix, we have
\[
∇_Σ^{-1} \ln|Σ^{-1}| = \frac{1}{|Σ^{-1}|} Σ^{-1} |Σ^T = Σ.
\] (32)
Using Eq. (30) and the fact that the determinant of the inverse of a matrix is the inverse of the determinant, $\mathcal{L}(\mu, \Sigma)$ can be rewritten as

$$
\mathcal{L}(\mu, \Sigma) = \ln|\Sigma^{-1}| \sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)} - \sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)} \text{tr}((z'_i - \mu)(z'_i - \mu)^\top \Sigma^{-1}).
$$  (33)

The local optimum $\Sigma_t$ is found by taking the gradient of $\mathcal{L}(\mu, \Sigma)$ w.r.t. the matrix $\Sigma^{-1}$ using the properties described in Eqs. (31) and (32), and equating it to zero, as given by

$$
\nabla_{\Sigma^{-1}} \mathcal{L} = \Sigma \sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)} - \sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)} (z'_i - \mu)(z'_i - \mu)^\top = 0,
$$  (34)

which then yields

$$
\Sigma_t = \frac{\sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)} (z'_i - \mu)(z'_i - \mu)^\top}{\sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)}},
$$  (35)

where $\mu_t$ is given by Eq. (28). The expression given by Eq. (13) can be derived by writing out each entry of the matrix in Eq. (35) to get

$$
\Sigma_{j,k} = \frac{\sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)} (z'_i - \mu)^{j,k} (z'_i - \mu)^{k,j}}{\sum_{i=1}^{m_f} \mathbb{I}_{g_i}(d_t, z'_i) \frac{p(z'_i)}{q_{0_t}(z'_i)}},
$$

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

The KL divergence, $\text{KL}(h_t \parallel q_{0_t})$, is a convex function (as shown using log-sum inequality in Ref. [6, Theorem 2.7.2, pp. 30]). We know that the local minimum in convex optimization must also be the global minimum and the first-order conditions are sufficient for optimality [28]. Consequently, the parameters $\theta_t^*$ given by Eqs. (12) and (13) are the local minimum as shown above and hence, also the global minimum of the optimization problem given by Eq. (11).

### B. Most Probable Point

The point with the maximum likelihood of failure is called the most probable point (MPP), see [20] for further reading. Typically, this is found by mapping $Z \sim p$ to the standard normal space $U \sim N(0, \text{diag}(1)) \in \mathbb{R}^n$. Let the mapping be done by using some transformation $u = T[z]$. Then the MPP can be found by minimizing the distance from the mean to the limit state failure boundary $g(z) = 0$ in the standard normal space. The optimization problem used to find the MPP is given by

$$
\min_{u \in \mathbb{R}^n} \sqrt{\|u\|_2}
$$

subject to $g(T^{-1}[u]) = 0$.

### 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 [16] proposes to sample from the mixed biasing density

$$
q^\alpha := (1 - \alpha)q + \alpha p.
$$
In [16] 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 $\alpha$ 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 $\alpha$ 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

$$q^\alpha_th := (1 - \alpha) \left( \sum_{i=0}^{l-1} \beta_i q^{*}_i \right) + \alpha p.$$ 

**Algorithm 3** Adaptive defensive importance sampling

**Input:** Nominal density $p$, biasing density (can be mixture) $q$, design $d_t$.

**Output:** Adaptive mixture density $q^\alpha_t$.

1: **procedure** AdaptiveISdensity($p$, $q$, $P_{thresh}$)
2: $\alpha_0 = 1$, $k = 1$;
3: **while** $\hat{P}_{IS}(d_t)$ not converged **do**
4: $q^\alpha_t := (1 - \alpha)q + \alpha p$
5: $m_k = kP_{thresh}^{-1}$ (start with a batch that would get $1/P_{thresh}$ samples
6: Compute $\hat{P}_{IS}(d_t)$ with samples from $q^\alpha_t$.
7: Assign $\alpha = \lfloor \frac{|\# \text{ of failed samples}}{m_k} \rfloor$ (i.e., if all samples fail high with nominal density (=high-FP), no need to use IS.)
8: $k = k + 1$.
9: **end while**
10: **return** $\hat{P}_{IS}(d_t)$
11: **end procedure**

**References**


