EFFICIENT SURROGATE MODELING FOR RELIABILITY ANALYSIS AND DESIGN

By

Barron James Bichon

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Civil Engineering

May, 2010

Nashville, Tennessee

Approved:

Professor Sankaran Mahadevan

Professor Prodyot K. Basu

Professor Mark Ellingham

Professor Caglar Oskay

Michael S. Eldred

To my wife, Angela,
who has worked as hard as I have to make this happen.

ACKNOWLEDGMENTS

This work was supported by funds from the National Science Foundation, through the Integrative Graduate Education and Research Traineeship multidisciplinary doctoral program in Risk and Reliability Engineering. My advisor, Dr. Sankaran Mahadevan, has made this program successful beyond any hopes of its originators and has led many of his students to fruitful careers in this burgeoning field. Without his determined efforts, this work and the work of many others would never have been possible.

I would also like to express my gratitude to the Computer Science Research Institute at Sandia National Laboratories for supporting me through several extended internships, which allowed me to work closely with the DAKOTA development team. Many of the analyses appearing throughout this dissertation were performed using this powerful software package. In particular, I am grateful for the mentorship and guidance provided so selflessly by Dr. Michael Eldred and Dr. Laura Swiler.

A special thanks goes to my classmate and colleague, Dr. John McFarland who taught me a lot and helped me debug many lines of code. I look forward to many years of continued collaboration and am eager to see what it brings.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Computer simulations are becoming increasingly common across all fields of engineering. These tools allow engineers to explore the response of a system at input values (demands, system properties, initial and boundary conditions) that could be difficult or even impossible to precisely control in the laboratory, and these "virtual tests" can be performed at far less cost than the laboratory experiments they simulate. These characteristics have made computer models especially useful in the field of reliability analysis where repeated tests at precise input values are required to investigate how variations in the inputs could lead to failures in the engineered system. This type of analysis is vital to ensure the safety and reliability of the system while in service.

One popular and powerful method for estimating reliability is Monte Carlo sampling. This method is favored for its accuracy and ease of implementation, but it can require a large number of computer simulations, especially for the high-reliability systems in which engineers are typically interested. As computer models have gained in fidelity, they have also gained in expense; it is not uncommon for a computer model to take several hours (or even days) to complete just one simulation. When this simulation must be repeated many thousands of times, basic Monte Carlo sampling becomes practically impossible. Other less computationally expensive reliability analysis methods are available, but they typically rely on approximations to simplify the analysis and can therefore be far less accurate. Then engineers are often left with a choice between an inexpensive method that could produce an inaccurate reliability estimate, or a more accurate method that they likely cannot afford.

The main contribution of this dissertation is the development of a new method called Efficient Global Reliability Analysis (EGRA) that bridges this gap. EGRA can provide the accuracy of Monte Carlo sampling at an expense typically less than that

required by even the least expensive approximate methods. One way EGRA reduces cost is through the use of an inexpensive surrogate model to capture the relationship between the inputs and outputs of the expensive computer simulation. This concept of using a so-called "response surface" is not new, but there are two aspects to EGRA that make it very different from other response surface methods. First, most response surfaces are made up of polynomial approximations; these can clearly only be accurate if the relationship between the computer simulation inputs and outputs truly follows this rigid structure. EGRA avoids this limitation by using a considerably more flexible model known as a Gaussian process model. Second, EGRA uses an optimization method to greatly reduce the number of expensive simulations necessary to construct the surrogate model while simultaneously ensuring that the surrogate provides an accurate representation of the underlying computer model. Using this optimization, a quality surrogate model can be constructed at a small fraction of the cost typically required by other response surface methods. Once this surrogate model has been constructed, it can be sampled to calculate the reliability. Because all of the simulations now come from this inexpensive surrogate, a large number of samples can be drawn at little cost, leading to highly accurate reliability estimates.

This chapter provides an introduction to the concepts of reliability analysis and reliability-based design and the previously available methods for solving these problems.

I.1 Reliability Analysis

The goal of reliability analysis is to determine the probability that an engineered device, component, system, etc. will fail in service given that its behavior is dependent on random inputs. This behavior is defined by a response function $g(\mathbf{x})$, where $\mathbf{x}$ represents the vector of random variables defined by known probability distributions. Failure is then defined by that response function exceeding (or failing to exceed) some

threshold value $\bar{z}$. The probability of failure, p_f, is then defined by

$$p_f = \int_{g>\bar{z}} \cdots \int f_{\mathbf{x}}(\mathbf{x})\, d\mathbf{x} \tag{I.1}$$

where $f_{\mathbf{x}}$ is the joint probability density function of the random variables $\mathbf{x}$, and the integration is performed over the failure region where $g > \bar{z}$. In general, $f_{\mathbf{x}}$ is impossible to obtain, and even when it is available, evaluating the multiple integral is difficult.[39] Because of these complications, methods of approximating this integral are used in practice.

I.1.1 Mean Value Method

The Mean Value First-Order Second-Moment method (MVFOSM)[39] is the simplest and least expensive reliability method. This method approximates the mean and variance of the response function based only on the response and its derivatives at the mean values of the random input variables. These statistics of the response can then be used to calculate the reliability at any response level of interest. The response mean μ_g, response variance σ_g^2, reliability index β, and probability of failure are approximated by

$$\mu_g = g\left(\mu_{\mathbf{x}}\right) \tag{I.2}$$

$$\sigma_g^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{Cov}\left(x_i, x_j\right) \frac{\partial g}{\partial x_i}\left(\mu_{\mathbf{x}}\right) \frac{\partial g}{\partial x_j}\left(\mu_{\mathbf{x}}\right) \tag{I.3}$$

$$\beta = \frac{\bar{z} - \mu_g}{\sigma_g} \tag{I.4}$$

$$p_f = \Phi\left(-\beta\right) \tag{I.5}$$

where n and $\mu_{\mathbf{x}}$ represent the number and mean values of the input random variables, $\mathrm{Cov}\left(x_i, x_j\right)$ is the covariance of x_i and x_j, and Φ is the standard normal cumulative distribution function. These approximations are reasonably accurate when the response function is nearly linear and the input random variables are approximately Gaussian,

but are not accurate if these conditions are not met.

I.1.2 MPP-Based Methods

These methods involve solving a nonlinear optimization problem to locate the point on the limit state (the contour on the response function where $g = \bar{z}$) that has the greatest probability of occurring. This point is known as the most probable point or MPP. An approximation to the limit state is then constructed at this point to facilitate the integration required to compute the probability of failure.

The MPP search is performed in the space of uncorrelated reduced normal variables because it simplifies the probability integration; in this space, the distance from the origin to the MPP is equivalent to the reliability index. The transformation from correlated non-normal distributions (x-space) to uncorrelated reduced normal distributions (u-space) is denoted as $\mathbf{u} = T(\mathbf{x})$ with the reverse transformation denoted as $\mathbf{x} = T^{-1}(\mathbf{u})$. These transformations are nonlinear in general, and possible approaches include the Rosenblatt,[67] Nataf,[19] and Box-Cox[10] transformations. The nonlinear transformations may also be linearized, and common approaches for this include the Rackwitz-Fiessler[65] two-parameter equivalent normal and the Chen-Lind[14] and Wu-Wirsching[91] three-parameter equivalent normals. This study employs the Nataf nonlinear transformation, which occurs in the following two steps. To transform between the original correlated x-space variables and correlated reduced normals (z-space), the CDF matching condition is used:

$$\Phi(z_i) = F(x_i) \tag{I.6}$$

where $F()$ is the cumulative distribution function of the original probability distribution. Then, to transform from correlated z-space variables to uncorrelated u-space variables, the Cholesky factor $\mathbf{L}$ of a modified correlation matrix is used:

$$\mathbf{z} = \mathbf{L}\mathbf{u} \tag{I.7}$$

where the original correlation matrix for non-normals in x-space has been modified for z-space.[19]

The forward reliability analysis algorithm for computing the probability/reliability level that corresponds to a specified response level is often called the z-level or reliability index approach,[78] and the inverse reliability analysis algorithm for computing the response level that corresponds to a specified probability/reliability level is often called the p-level or performance measure approach.[78] The differences between the z-level and p-level formulations appear in the objective function and equality constraint formulations in the MPP searches. For the forward z-level analysis, the MPP search for achieving the specified response level $\bar{z}$ is formulated as

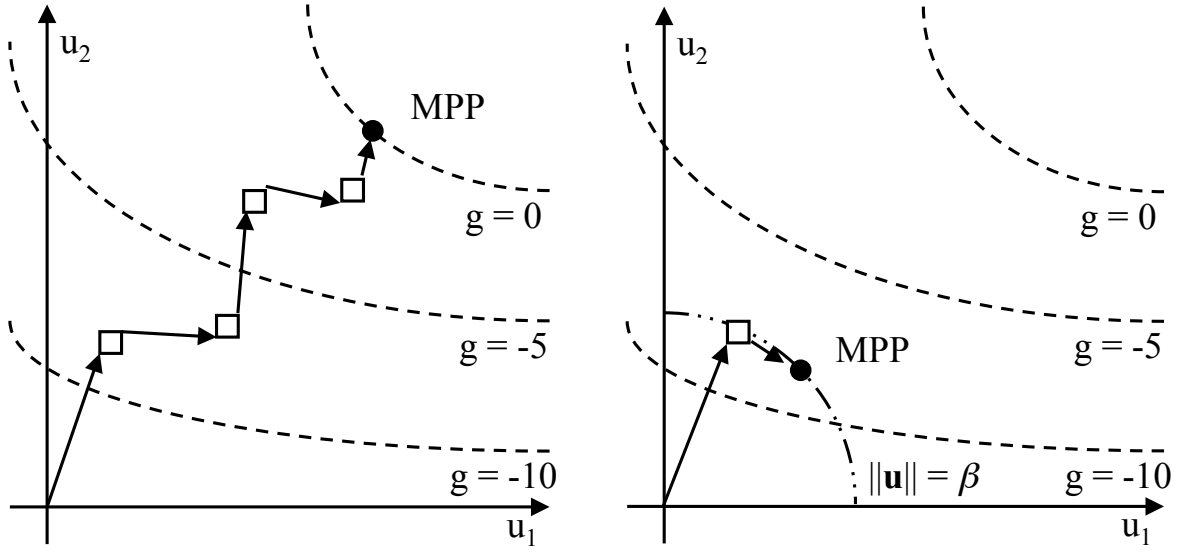
$$\begin{aligned} &\text{minimize} && \mathbf{u}^T\mathbf{u} \\ &\text{subject to} && G(\mathbf{u}) = \bar{z} \end{aligned} \tag{I.8}$$

and for the inverse p-level analysis, the MPP search for achieving the specified probability/reliability level $\overline{p_f}, \bar{\beta}$ is formulated as

$$\begin{aligned} &\text{minimize} && \pm G(\mathbf{u}) \\ &\text{subject to} && \mathbf{u}^T\mathbf{u} = \bar{\beta}^2 \end{aligned} \tag{I.9}$$

where $\mathbf{u}$ is a vector centered at the origin in u-space and $G(\mathbf{u}) \equiv g(\mathbf{x})$ by definition. Graphical representations of possible iteration histories for forward and inverse reliability analysis problems are shown in Figure I.1 (adapted from Ref. 52).

In the forward reliability case, the optimal MPP solution $\mathbf{u}^*$ defines the reliability index from $\beta = \pm\|\mathbf{u}^*\|_2$, which in turn defines the probability of failure through the



(a) Forward reliability analysis (z-level). (b) Inverse reliability analysis (p-level).

Figure I.1: Possible reliability analysis iteration histories.

probability integration, e.g. $p_f = \Phi(-\beta)$. The sign of β is defined by

$$G(\mathbf{u}^*) > G(\mathbf{0}) : \beta < 0 \tag{I.10}$$

$$G(\mathbf{u}^*) < G(\mathbf{0}) : \beta > 0 \tag{I.11}$$

where $G(\mathbf{0})$ represents the response at the median values of the input random variables, i.e. the origin in u-space. In the inverse reliability case, the sign applied to $G(\mathbf{u})$ in the objective function (equivalent to choosing whether to minimize or maximize the response) is determined by the sign of the target reliability level $\bar{\beta}$:

$$\bar{\beta} < 0 : \text{maximize } G(\mathbf{u}) \tag{I.12}$$

$$\bar{\beta} > 0 : \text{minimize } G(\mathbf{u}) \tag{I.13}$$

The value of the response function at the optimal MPP solution $G(\mathbf{u}^*)$ defines the desired response level result.

Response Function Approximations

The search for the MPP requires the solution of an optimization problem, which necessitates multiple evaluations of the response function. If evaluating this function is computationally expensive, the cost of the MPP search can be prohibitive. To reduce this cost, the response function can be replaced with an approximation. Several approaches based on the Advanced Mean Value (AMV) method were explored in detail in Refs. 26, 87:

1. AMV in x-space: A single linearization of the response function performed in x-space at the mean values of the input random variables.

$$g(\mathbf{x}) \approx g(\mu_{\mathbf{x}}) + \nabla_x g(\mu_{\mathbf{x}})^T (\mathbf{x} - \mu_{\mathbf{x}}) \tag{I.14}$$

2. AMV in u-space: A single linearization of the response function performed in u-space at the mean values of the input random variables.

$$G(\mathbf{u}) \approx G(\mu_{\mathbf{u}}) + \nabla_u G(\mu_{\mathbf{u}})^T (\mathbf{u} - \mu_{\mathbf{u}}) \tag{I.15}$$

   where $\mu_{\mathbf{u}} = T(\mu_{\mathbf{x}})$.

3. AMV+ in x-space: An initial linearization at the random variable means, with the linearization iteratively updated at each candidate MPP ($\mathbf{x}^*$) until the MPP converges.

$$g(\mathbf{x}) \approx g(\mathbf{x}^*) + \nabla_x g(\mathbf{x}^*)^T (\mathbf{x} - \mathbf{x}^*) \tag{I.16}$$

4. AMV+ in u-space: An initial linearization at the random variable means, with the linearization iteratively updated at each candidate MPP ($\mathbf{u}^*$) until the MPP converges.

$$G(\mathbf{u}) \approx G(\mathbf{u}^*) + \nabla_u G(\mathbf{u}^*)^T (\mathbf{u} - \mathbf{u}^*) \tag{I.17}$$

Depending on the form of the response function and the variable transformation, either $g(\mathbf{x})$ or $G(\mathbf{u})$ may be more linear than the other, meaning the approximation in the appropriate space (x- or u-space) will be more accurate over a larger range. This will result in more accurate MPP estimates for AMV, and faster convergence to the MPP for AMV+. Second-order information can be included to create the AMV^2, AMV^2+, and TANA response function approximations, which have been recently explored in Ref. 24.

Probability Integration

In the forward reliability case, once the MPP is located, the probability integration is performed based upon the location of the MPP (more specifically, its distance from the median response, i.e. the reliability index β) and an approximation to the shape of the limit state at this point. The simplest method assumes the limit state is linear, simplifying the probability integration to

$$p_f = \Phi(-\beta) \tag{I.18}$$

Second-order integration methods that provide a curvature correction to the first-order method are also available, the simplest of which is[11]

$$p_f = \Phi(-\beta) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1+\beta\kappa_i}} \tag{I.19}$$

where κ_i are the principal curvatures of the response function at the MPP and $\beta > 0$. An alternate correction in Ref. 43 is consistent in the asymptotic regime ($\beta \to \infty$) but does not collapse to first-order integration for $\beta = 0$:

$$p = \Phi(-\beta) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1+\psi(-\beta)\kappa_i}} \tag{I.20}$$

where $\psi() = \frac{\phi()}{\Phi()}$ and $\phi()$ is the standard normal density function. Ref. 44 applies further corrections to Eq. I.20 based on point concentration methods. Ref. 79 develops full second-order approximations that can generally be more accurate than these parabolic approximations.

For highly nonlinear limit states, these low-order approximations cannot provide accurate representations. Additionally, such problems often possess multiple local solutions to the MPP search problem and are thus termed "multimodal" problems. Ref. 55 proposed a method where these multiple MPPs are first located, then a separate linear approximation is formed at each MPP. These multiple point linearizations are then combined using methods from system-level reliability analysis (discussed further in Chapter III) to calculate the probability of failure. Using multiple approximations makes this method more accurate, but locating the multiple MPPs can be prohibitively expensive. Moreover, the method does not appear to work well for multidimensional problems with irregular-shaped limit states, due to the difculty in the identication of union and intersection domains.[55]

Combining an MPP search that does not use a response function approximation with first- or second-order integrations results in the common First- and Second-Order Reliability Methods (FORM and SORM, respectively).

I.1.3 Sampling Methods

Sampling methods are used to numerically integrate Equation I.1. Because they do not rely on any simplifying assumptions about the shape of the limit state like first- and second-order integration, they are generally more accurate than these methods, but they also typically require a large number of function evaluations, making them impractical if the response function is expensive to evaluate.

Monte Carlo Sampling

The simplest sampling method is basic Monte Carlo. First, some number of samples of the input variables are randomly generated based on their random distributions and the response function is then evaluated at these input values. The value of the response function at these sample points is then compared to the response level that defines the limit state to determine if the observed response is a success or failure. The probability of failure is then calculated as simply the ratio of observed failures to the total number of observations:

$$p_f = \frac{N_f}{N} \tag{I.21}$$

where N_f is the number of observed failures from among all N samples.

One major drawback to this method is that the majority of the samples it generates lie in the high-probability region of the random variable space. Because engineers are typically concerned with high-reliability problems, this region is of little interest as the limit state most likely lies in a much less probable region of the design space. To ensure that enough samples are generated in these low-probability regions to give an accurate estimate of the probability of failure, a very large number of samples can be required. The percent error $\epsilon^{\%}$ in the probability estimate can be approximately related to the number of samples used by:

$$\epsilon^{\%} = \sqrt{\frac{1 - p_f^T}{N p_f^T}} \times 200\% \tag{I.22}$$

where p_f^T is the true probability of failure. This relationship can be inverted to show that for $p_f^T = 0.0057$ and a desired error of 1%, approximately seven million samples would be required. These numbers were not chosen arbitrarily. A two-dimensional example problem presented in Section II.4.2 will show that the Efficient Global Reliability Analysis method developed later in this dissertation is capable of resolving this probability level with an error below 1% with only 18 function evaluations.

Because of its simplicity, basic Monte Carlo sampling is widely used, but more efficient sampling methods are available.

Latin Hypercube Sampling

Latin hypercube sampling (LHS) is an alternative to basic Monte Carlo sampling that can provide adequate coverage of the random variable space with far fewer samples. First, each dimension of the space is broken into an equal number of bins of equal probability. The number of bins defines the number of samples required. The samples are then randomly placed within the bins so that each one-dimensional projection of the space will show that each bin contains exactly one sample.

Figure I.2 shows a comparison between sampling two uniformly distributed variables using basic Monte Carlo and Latin Hypercube sampling. Note that each dimension is broken into 10 intervals and 10 samples have been generated. It is easy to see that for the LHS example, each of these intervals contains exactly one sample, but the Monte Carlo samples obviously do not possess this structure.

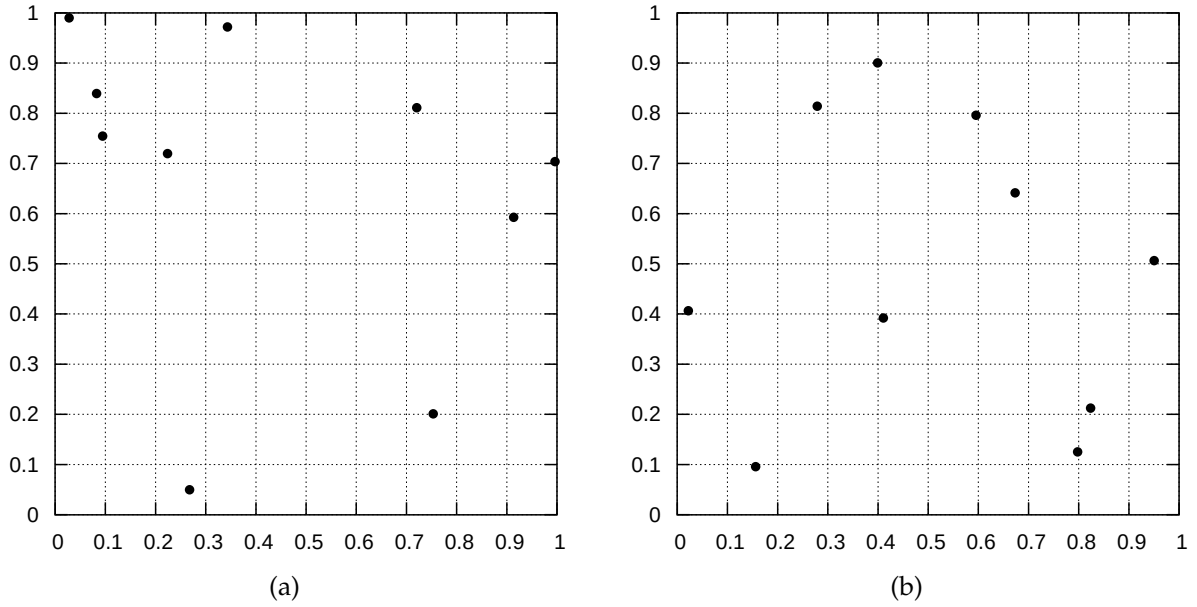


Figure I.2: Comparison of (a) basic Monte Carlo and (b) Latin Hypercube sampling.

By forcing the samples into these bins, coverage throughout the random variable space is guaranteed. However, because the bins are divided such that each has an

equal probability, the low-probabilty regions will have larger bins and therefore fewer samples. To counteract this, more bins are required, which in turn means that more samples are required. In general, LHS requires considerably fewer samples than basic Monte Carlo, but still requires too many function evaluations to be a feasible option for expensive response functions.

I.1.4 Hybrid Methods

Hybrid methods are thus called because they combine MPP-based methods with sampling. By first locating the failure region via an MPP search, and then focusing the sampling in this region, hybrid methods are both more accurate and more efficient than either MPP-based or sampling methods alone.

Importance Sampling

There are two main methods of importance sampling. The first simply centers a sampling density at the MPP rather than at the mean of the input variables. This ensures that more samples will lie within the failure region, i.e. the important region, thereby increasing the efficiency of the sampling. The second, introduced in Ref. 40, restricts all samples to lie outside of a hypersphere of radius β because it is already known that no failures exist within this region. Using these methods, the probability of failure is calculated by:

$$p_f = \frac{1}{N} \sum_{i=1}^{N} I\left(\mathbf{x}_i\right) \frac{f(\mathbf{x}_i)}{\varphi(\mathbf{x}_i)} \tag{I.23}$$

where I is an indicator function ($I = 1$ if $\mathbf{x}_i$ is a failure point, otherwise $I = 0$), φ is the sampling density used to generate the samples $\mathbf{x}_i$, and f is the original density of $\mathbf{x}$. The choice of sampling density depends on the importance sampling method being used.

Adaptive Importance Sampling

These methods are termed adaptive because they iteratively update the sampling density being used based on the sample data from previous iterations. One such method, introduced in Refs. 89, 90, uses the sample data to form and update an approximation to the curvature of the limit state. Because this curvature is approximated from sample data rather than the Hessian information at the MPP, higher-order curves can be represented. This method can therefore be more accurate than SORM for limit states that are more nonlinear than a second-order approximation can capture, but it still only allows for unimodal limit states.

Multimodal Adaptive Importance Sampling

Multimodal adaptive importance sampling[20,94] (MAIS) is a variation of importance sampling that allows for the use of multiple sampling densities making it better suited for cases where multiple sections of the limit state are highly probable. MAIS is performed through the following steps:

1. Generate m_0 initial samples using a sampling density centered at the MPP.
2. Find k representative points from these samples:
   (a) Select the sample with the largest true probability of occurrence.
   (b) Eliminate all samples within a specified distance (e.g. β) of this point.
   (c) Repeat steps (a) and (b) until all samples are exhausted.
3. Calculate the coefficient of variation (COV) of p_f from the m_0 samples.

$$p_f = \frac{1}{m_0} \sum_{i=1}^{m_0} \frac{f(\mathbf{x}_i)}{\varphi(\mathbf{x}_i)} I(\mathbf{x}_i) \tag{I.24}$$

$$\sigma^2 = \frac{1}{m_0(m_0-1)} \sum_{i=1}^{m_0} \left[\frac{f(\mathbf{x}_i)}{\varphi(\mathbf{x}_i)} I(\mathbf{x}_i) - p_f \right]^2 \tag{I.25}$$

$$COV = \frac{\sigma}{p_f} \tag{I.26}$$

where φ is the multimodal sampling density function defined in Eq. I.28 (initially, it is the sampling density function centered at the MPP).

4. Use the k representative points to construct a multimodal sampling density φ.

   (a) Calculate the weight for each representative point, which is based on its probability density relative to that of the other representative points:

   $$w_i = \frac{f(\mathbf{x}_i)}{\sum_{j=1}^{k} f(\mathbf{x}_j)} \tag{I.27}$$

   (b) The multimodal sampling density is then the weighted sum of the probability densities centered at the representative points:

   $$\varphi(\mathbf{x}) = \sum_{i=1}^{k} w_i f_i(\mathbf{x}) \tag{I.28}$$

   where f_i is the true probability density with the mean shifted to the i^{th} representative point.

   (c) Generate m_1 samples using φ. If desired, the variance of f_i in Eq. I.28 can be increased to force greater exploration of the failure region.

   (d) Repeat Steps 2-4 until the COV converges, replacing m_0 with m_1 where needed.

5. Generate m_2 samples using the sampling density φ from the final set of representative points.

6. Calculate p_f and its change δ_p from these samples.

7. Repeat Steps 5-6 until p_f converges and δ_p is less than some threshold value.

Note that all of these hybrid methods require that the location of the MPP be known because this point is used as the center of the initial sampling density. However, current gradient-based, local search methods used in the MPP search may fail to

converge or may converge to poor solutions, possibly making these methods inapplicable or inaccurate.

I.1.5 Surrogate Models

The basic idea of a surrogate model (also known as a response surface or metamodel) is to use a relatively small number of evaluations from the (presumably expensive) response function of interest to construct an approximation to that function that is cheaper to evaluate. Ref. 95 employs a notably different tactic by forming a surrogate model of the indicator function I rather than the response function. The surrogate model can then serve as a "stand-in" for the real model in a sampling method to perform a reliability analysis. The first use of a surrogate model was proposed in Ref. 9 where a second-order polynomial approximation was used. Polynomial models have remained popular ever since due to their ease of construction and simple evaluation, but they are fairly rigid in their form and may produce inaccurate models if the form of the underlying model is not a polynomial, leading to inaccurate reliability estimates.

Several more flexible types of surrogate models are available including radial basis functions,[13] multivariate adaptive regression splines,[33] polynomial chaos expansions,[30,92] and kriging (or, more generally termed, Gaussian Process) models.[18] The main advantage of these types of models is their ability to adapt to the training data. Consider the comparison of polynomial models to a kriging model fit to the same data shown in Figure I.3.[73] The quadratic polynomial provides a poor fit to the training data (the red triangles), and the quartic polynomial provides only a slightly more accurate fit. However, the kriging model passes directly through all of the training points and creates a smooth interpolation.

Ref. 83 has shown that Gaussian Process models can be used to provide accurate reliability estimates for nonlinear response functions using far fewer function evaluations than Monte Carlo sampling alone. EGRA takes additional advantage of some

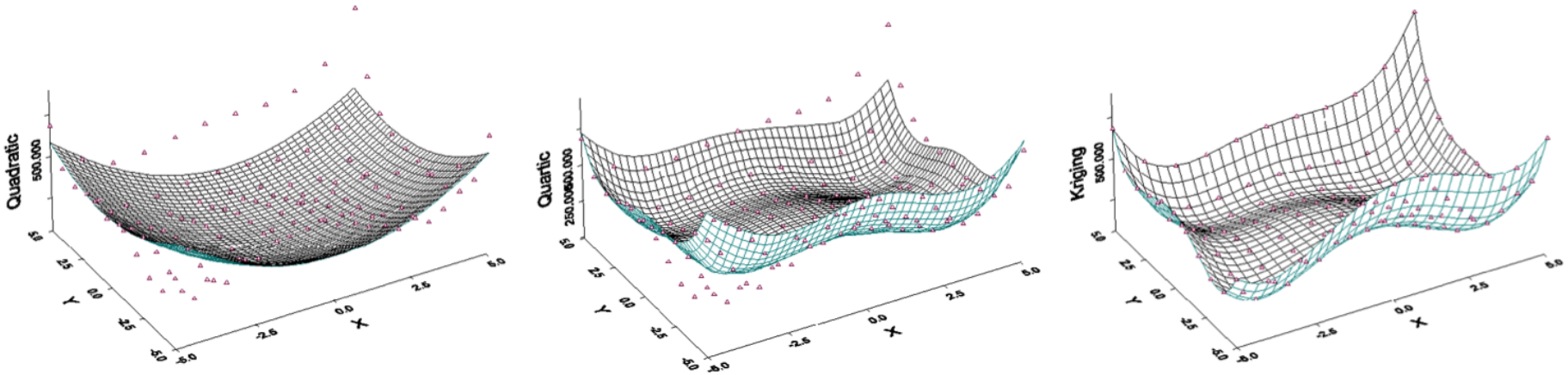


Figure I.3: Comparison of polynomial response surfaces to a kriging model. On the left is a quadratic polynomial; in the center is a quartic polynomial; on the right is the kriging model. All models are fit to the same set of data.

special features of this type of model to reduce this cost even further. Additional details on Gaussian Process models are provided in Section II.1.1.

I.2 Reliability-Based Design Optimization

Reliability-Based Design Optimization (RBDO) is used to perform design optimization (such as minimizing the weight of a component) while accounting for reliability constraints. A general RBDO problem is typically of the form:

$$\begin{aligned} &\text{minimize} && f(\mathbf{d}) \\ &\text{subject to} && P\left[g(\mathbf{d},\mathbf{x}) \geq \bar{z}\right] \leq \bar{p}_f \end{aligned} \tag{I.29}$$

where the objective function f is a function of only the deterministic design variables $\mathbf{d}$, but the response function in the reliability constraint g is a function of $\mathbf{d}$ and $\mathbf{x}$, a vector of random variables defined by known probability distributions.

I.2.1 Nested RBDO

The simplest and most direct RBDO approach is the nested approach in which a full reliability analysis is performed for every evaluation of the constraint function in Eq. V.1. This involves a nesting of two distinct levels of optimization within each other, one at the design level and one at the reliability analysis level.

An important performance enhancement for nested methods when MPP-based reliability analysis methods are used is the use of sensitivity analysis to analytically compute the design gradients of the probability constraint. When design variables are separate from the random variables (i.e., they are not distribution parameters), then the following first-order expression may be used:[4,41,48]

$$\nabla_{\mathbf{d}} p_f = -\phi(-\beta)\frac{1}{\|\nabla_{\mathbf{u}} G\|}\nabla_{\mathbf{d}} g \tag{I.30}$$

where β is the reliability index (the distance from the median response to the MPP),

ϕ is the standard normal probability density function, $\mathbf{u}$ are the random variables transformed into standard normal space (where the MPP search is performed), and $G(\mathbf{u}) \equiv g(\mathbf{x})$ by definition. Utilizing the higher-order sensitivities that result from second-order reliability analyses are discussed in Ref. 24.

When the design variables are distribution parameters of the random variables, $\nabla_{\mathbf{d}} g$ is expanded with the chain rule and Eq. I.30 becomes:

$$\nabla_{\mathbf{d}} p_f = -\phi(-\beta) \frac{1}{\|\nabla_{\mathbf{u}} G\|} \nabla_{\mathbf{d}} x \nabla_{\mathbf{x}} g \tag{I.31}$$

where the design Jacobian of the $\mathbf{x} \rightarrow \mathbf{u}$ transformation $\nabla_{\mathbf{d}} x$ can only be obtained analytically for uncorrelated $\mathbf{x}$.

The ability to provide this kind of sensitivity information is an advantage of MPP-based methods, but because of the possible inaccuracy in this type of reliability analysis, they can lead to optimal solutions that do not satisfy the constraint on the probability of failure. Approximate sensitivities can be derived for Monte Carlo Sampling, but in a nested formulation, the probability of failure may need to be calculated a large number of times (depending on the number of iterations for convergence of the optimization), making Monte Carlo sampling rarely feasible for RBDO due to the computational expense involved.

I.2.2 Single-Loop RBDO

This formulation of the RBDO problem simultaneously optimizes the objective function and searches for the MPP, satisfying the probabilistic constraint only at the optimal solution. The Karush-Kuhn-Tucker (KKT) conditions are used to reformulate the first-order reliability constraint into an equivalent deterministic constraint. In this way, the need to locate the MPP for the constraints is completely eliminated, but at convergence, the MPPs of the active constraints will be found.

Recall that FORM uses a linear approximation to the shape of the limit state in $\mathbf{u}$ space. This is equivalent to using a first-order Taylor series expansion about the MPP.

$G(\mathbf{u})$ is then a normal variable with parameters defined by:

$$\mu_G = G(\mathbf{u}) - \nabla_{\mathbf{u}} G^T \mathbf{u} \tag{I.32}$$

$$\sigma_G = \|\nabla_{\mathbf{u}} G\| \tag{I.33}$$

The general definition of the reliability index is the ratio of the mean output response and its standard deviation.[39] For this first-order approximation, at the MPP $\mathbf{u}^*$ (assuming that the limit state value $\bar{z}=0$, which can always be satisfied), β is therefore:

$$\beta = \frac{-\nabla_{\mathbf{u}} G^T \mathbf{u}^*}{\|\nabla_{\mathbf{u}} G\|} \tag{I.34}$$

The KKT conditions can be used to show that this is equivalent to defining the reliability index as $\beta = \|\mathbf{u}^*\|$.[15,51] Substituting this definition of β into Eq. I.34 and rearranging gives the KKT optimality condition:

$$\nabla_{\mathbf{u}} G^T \mathbf{u} + \|\mathbf{u}\| \|\nabla_{\mathbf{u}} G\| = 0 \tag{I.35}$$

Because this condition is only satisfied at the MPP, it can be used as an equality constraint in a now deterministic formulation of the RBDO problem:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{d}) \\ \text{subject to} \quad & \beta \geq \bar{\beta} \\ & G(\mathbf{d}, \mathbf{u}) = 0 \\ & \nabla_{\mathbf{u}} G^T \mathbf{u} + \beta \|\nabla_{\mathbf{u}} G\| = 0 \end{aligned} \tag{I.36}$$

The first constraint states that the final reliability index must be greater than a minimum reliability index $\bar{\beta}$, which is derived from the original probability constraint assuming a first order limit state, $\bar{\beta} = -\Phi^{-1}(\bar{p}_f)$, where Φ is the standard normal cumulative distribution function. The second constraint ensures that the MPP lies on

the limit state. The third constraint is the KKT optimality condition, the derivation of which (as described above) also relies on the first-order limit state assumption.

Investigations into this formulation of the RBDO problem in Ref. 3, 15, 51, 54 have shown that it is far more efficient than the nested formulation. Ref. 57 developed extensions to this method making it applicable to system-level RBDO (discussed in more detail in Chapters III and V). However, the accuracy of the final solution relies on the accuracy of the underlying first-order assumption.

I.2.3 Sequential RBDO

An alternative RBDO approach is the sequential approach, in which additional efficiency is sought by breaking the nested relationship of the MPP and design searches. The general concept is to iterate between optimization and uncertainty quantification, updating the optimization goals based on the most recent probabilistic assessment results. This update may be based on safety factors[88] or other approximations.[22]

A particularly effective approach for updating the optimization goals is to use the p_f sensitivity analysis described in Eq. I.30 in combination with local surrogate models.[96–98] In Ref. 26, first-order Taylor series approximations were explored, and a trust-region model management framework[36] was used to adaptively manage the extent of the approximations and ensure convergence of the RBDO process. Surrogate models can be used for both the objective function and the constraints, although the use of constraint surrogates alone is sufficient to remove the nesting.

In particular, trust-region surrogate-based RBDO employs surrogate models of f and p_f within a trust region Δ^k centered at $\mathbf{d}_c$:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{d}_c) + \nabla_{\mathbf{d}} f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \\ \text{subject to} \quad & p_f(\mathbf{d}_c) + \nabla_{\mathbf{d}} p_f(\mathbf{d}_c)^T (\mathbf{d} - \mathbf{d}_c) \\ & \|\mathbf{d} - \mathbf{d}_c\|_\infty \leq \Delta^k \end{aligned} \tag{I.37}$$

Sequential RBDO is more efficient than the nested formulation, but generally more expensive than the single-loop formulation. It can be more accurate than single-loop results due to the possible inclusion of higher order approximations to the limit state, but will still be inaccurate if those approximations are poor.

I.3 Summary

This chapter has provided an introduction to prominent methods for reliability analysis and reliability-based design optimization. It has discussed the challenges involved in both types of problems and the shortcomings of currently available methods in solving them. Subsequent chapters will describe proposed new approaches that provide vast improvements in terms of both accuracy and efficiency.

Chapter II outlines a new approach to reliability analysis termed Efficient Global Reliability Analysis. Chapters III, IV, and V outline how this new method can be applied to several challenging problems in reliability analysis and the advantages that it brings.

CHAPTER II

EFFICIENT GLOBAL RELIABILITY ANALYSIS

As engineering applications become increasingly complex, they are often characterized by implicit response functions that are expensive to evaluate and perhaps nonlinear in their behavior. Reliability analysis given this type of response is difficult with available methods. As previous chapters have discussed, analytical reliability methods (e.g., FORM, SORM) solve a local optimization problem to locate the MPP, and then quantify the reliability based on its location and an approximation to the shape of the limit state at this point. Typically, gradient-based solvers are used to solve this optimization problem, which may fail to converge for nonsmooth response functions with unreliable gradients or may converge to only one of several solutions for response functions that possess multiple local optima. In addition to these MPP convergence issues, the evaluated probabilites can be adversely affected by limit state approximations that may be inaccurate.

Engineers are then forced to revert to sampling methods, which do not rely on MPP convergence or simplifying approximations to the true shape of the limit state. However, sampling methods typically require a large number of response function evaluations, which can make their application infeasible for computationally expensive problems. Ref. 77 provides a good overview of the errors in current methods when applied to nonlinear problems from structural dynamics and material fatigue, motivating the need for new methods with greater accuracy.

A reliability analysis method that is both efficient when applied to expensive response functions and accurate for a response function of any arbitrary shape is needed. This chapter develops a method based on efficient global optimization[47] (EGO) to adaptively search for multiple points on or near the limit state throughout the random variable space. By locating multiple points near the limit state, more complicated and

nonlinear limit states can be accurately modeled, resulting in an accurate assessment of the reliability.

EGO was developed to facilitate the unconstrained minimization of expensive implicit response functions. The method builds an initial Gaussian process model as a global surrogate for the response function, then adaptively selects additional samples to be added for inclusion in a new Gaussian process model in subsequent iterations. The new samples are selected based on how much they are expected to improve the current best solution to the optimization problem. When this expected improvement is acceptably small, the globally optimal solution has been found. The application of this methodology to equality-constrained reliability analysis is the primary contribution detailed in this chapter. Combining this EGO-based search for the limit state contour with a sampling method to calculate the probability of failure results in what is referred to as efficient global reliability analysis (EGRA).

The use of Gaussian process models in reliability analysis was previously investigated in Refs. 37 and 83. However, there are key differences in the previous work and the EGRA method introduced here. The earlier methods used a number of randomly selected samples to construct the model and global accuracy of that model was sought. This results in either a lack of accuracy if too few samples are used, or wasted expense creating models that are accurate in areas where they need not be. EGRA avoids these problems by not requiring the surrogate model to have high accuracy throughout the random variable domain, but only in the vicinity of the limit state. This is accomplished by focusing the training data around the limit state and greatly reduces the number of samples required. Additionally, the search for the limit state is performed using an iterative process with a rigorous convergence criteria, ensuring that the final model provides an accurate depiction of the limit state.

Section II.1 gives an overview of the EGO algorithm. Sections II.2 and II.3 outline how EGO is adapted for application to reliability analysis to create the EGRA method. Section II.4 describes a collection of example problems and compares the performance

of EGRA to that of other available methods. Finally, Section II.5 provides concluding remarks on this new method.

II.1 Efficient Global Optimization

Efficient global optimization was originally proposed by Jones et al.[47] and has been adapted into similar methods such as sequential kriging optimization (SKO).[45] The main difference between SKO and EGO lies within the specific formulation of what is known as the expected improvement function (EIF), which is the feature that sets all EGO/SKO-type methods apart from other global optimization methods. The EIF is used to select the location at which a new training point should be added to the Gaussian process model by maximizing the amount of improvement in the objective function that can be expected by adding that point. A point could be expected to produce an improvement in the objective function if its predicted value is better than the current best solution, or if the uncertainty in its prediction is such that the probability of it producing a better solution is high. Because the uncertainty is higher in regions of the design space with fewer observations, this provides a balance between exploiting areas of the design space that predict good solutions, and exploring areas where more information is needed. The general procedure of these EGO-type methods is:

1. Build an initial Gaussian process model of the objective function.
2. Find the point that maximizes the EIF. If the EIF value at this point is sufficiently small, stop.
3. Evaluate the objective function at the point where the EIF is maximized. Update the Gaussian process model using this new point. Go to Step 2.

The following sections discuss the construction of the Gaussian process model used, present the form of the EIF, demonstrate EGO through a simple example, and then provide a description of how that EIF is modified for application to reliability analysis.

II.1.1 Gaussian Process Models

Gaussian process (GP) models are set apart from other surrogate models because they provide not just a predicted value at an unsampled point, but also an estimate of the prediction variance. This variance gives an indication of the uncertainty in the GP model, which results from the construction of the covariance function. This function is based on the idea that when input points are near one another, the correlation between their corresponding outputs will be high. As a result, the uncertainty associated with the model's predictions will be small for input points which are near the points used to train the model, and will increase as one moves further from the training points.

It is assumed that the true response function being modeled $g(\mathbf{x})$ can be described by:[18]

$$g(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta} + Z(\mathbf{x}) \tag{II.1}$$

where $\mathbf{h}()$ is the trend of the model, $\boldsymbol{\beta}$ is the vector of trend coefficients, and $Z()$ is a stationary Gaussian process with zero mean (and covariance defined below) that describes the departure of the model from its underlying trend. The trend of the model can be assumed to be any function, but taking it to be a constant value has been reported to be generally sufficient.[68] For the work presented here, the trend is assumed constant and $\boldsymbol{\beta}$ is calculated through a Generalized Least Squares estimate. The covariance between outputs of the Gaussian process $Z()$ at points $\mathbf{a}$ and $\mathbf{b}$ is defined as:

$$Cov\left[Z(\mathbf{a}), Z(\mathbf{b})\right] = \sigma_Z^2 R(\mathbf{a}, \mathbf{b}) \tag{II.2}$$

where σ_Z^2 is the process variance and $R()$ is the correlation function. There are several options for the correlation function, but the squared-exponential function is common,[68] and is used here for $R()$:

$$R(\mathbf{a}, \mathbf{b}) = \exp\left[-\sum_{i=1}^{d} \theta_i (a_i - b_i)^2\right] \tag{II.3}$$

where d represents the dimensionality of the problem (the number of random variables), and θ_i is a scale parameter that indicates the correlation between the points within dimension i. A large θ_i is representative of a short correlation length.

The expected value $\mu_g()$ and variance $\sigma_g^2()$ of the GP model prediction at point $\mathbf{x}$ are:

$$\mu_g(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T\boldsymbol{\beta} + \mathbf{r}(\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{g} - \mathbf{F}\boldsymbol{\beta}) \tag{II.4}$$

$$\sigma_g^2(\mathbf{x}) = \sigma_Z^2 - \begin{bmatrix}\mathbf{h}(\mathbf{x})^T & \mathbf{r}(\mathbf{x})^T\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{F}^T\\ \mathbf{F} & \mathbf{R}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{h}(\mathbf{x})\\ \mathbf{r}(\mathbf{x})\end{bmatrix} \tag{II.5}$$

where $\mathbf{r}(\mathbf{x})$ is a vector containing the covariance between $\mathbf{x}$ and each of the n training points (defined by Eq. II.2), $\mathbf{R}$ is an $n \times n$ matrix containing the correlation between each pair of training points, $\mathbf{g}$ is the vector of response outputs at each of the training points, and $\mathbf{F}$ is an $n \times q$ matrix with rows $\mathbf{h}(\mathbf{x}_i)^T$ (the trend function for training point i containing q terms; for a constant trend $q = 1$). This form of the variance accounts for the uncertainty in the trend coefficients $\boldsymbol{\beta}$, but assumes that the parameters governing the covariance function (σ_Z^2 and $\boldsymbol{\theta}$) have known values.

The parameters σ_Z^2 and $\boldsymbol{\theta}$ are determined through maximum likelihood estimation. This involves taking the log of the probability of observing the response values $\mathbf{g}$ given the covariance matrix $\mathbf{R}$, which can be written as:[68]

$$\log\left[p(\mathbf{g}|\mathbf{R})\right] = -\frac{1}{n}\log|\mathbf{R}| - \log(\hat{\sigma}_Z^2) \tag{II.6}$$

where $|\mathbf{R}|$ indicates the determinant of $\mathbf{R}$, and $\hat{\sigma}_Z^2$ is the optimal value of the variance given an estimate of $\boldsymbol{\theta}$ and is defined by:

$$\hat{\sigma}_Z^2 = \frac{1}{n}(\mathbf{g} - \mathbf{F}\boldsymbol{\beta})^T\mathbf{R}^{-1}(\mathbf{g} - \mathbf{F}\boldsymbol{\beta}) \tag{II.7}$$

Maximizing Eq. II.6 gives the maximum likelihood estimate of $\boldsymbol{\theta}$, which in turn defines

σ_Z^2.

An advantage to Gaussian process models as compared to other methods such as polynomial regression is that they directly interpolate the training data rather than attempting to fit a curve between them. This means that the error in the model at known points is zero. This is reflected in the GP by the variance in the Gaussian distribution at the training point being zero. In this way, the variance in the model is used as a metric for the uncertainty in its prediction. Figure II.1 shows a simple Gaussian process model. The expected values predicted by the GP model pass directly through each of the training points and the 95% confidence bounds grow larger as points move further from the training data.

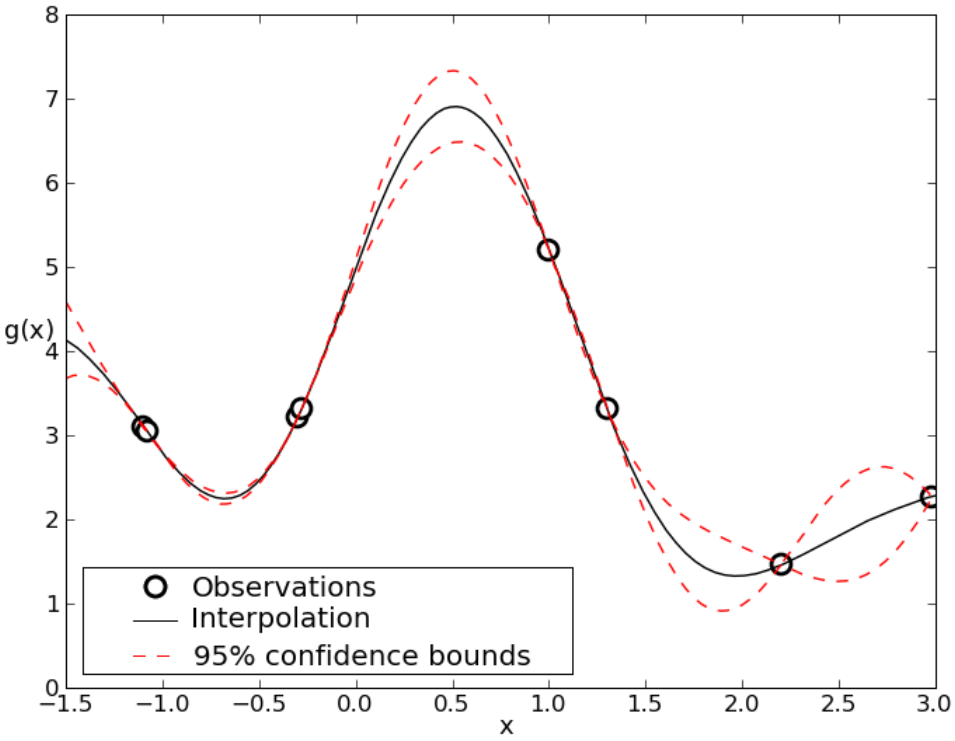


Figure II.1: Plot of simple Gaussian process model.

The expected improvement function utilizes this measure of the uncertainty to search for better solutions to the objective function, where "better" refers to lower solutions since EGO is constructed as a minimizer.

II.1.2 Expected Improvement Function

The expected improvement function is used to select the location at which a new training point should be added. The EIF is defined as the expectation that any point in the search space will provide a better solution than the current best solution (i.e. a solution with a smaller value than the smallest found so far since the method is seeking the minimum) based on the expected values and variances predicted by the GP model. An important feature of the EIF is that it provides a balance between exploiting areas of the design space where good solutions have been found, and exploring areas of the design space where the uncertainty is high. First, recognize that at any point in the design space, the GP prediction $\hat{g}()$ follows a Gaussian distribution:

$$\hat{g}(\mathbf{x}) \sim N\left[\mu_{\hat{g}}(\mathbf{x}), \sigma_{\hat{g}}(\mathbf{x})\right] \tag{II.8}$$

where the mean $\mu_{\hat{g}}()$ and the variance $\sigma^2_{\hat{g}}()$ were defined in Eqs. II.4 and II.5, respectively. The EIF is defined as:[47]

$$EI\big(\hat{g}(\mathbf{x})\big) \equiv E\left[\max\left(g(\mathbf{x}^*) - \hat{g}(\mathbf{x}), 0\right)\right] \tag{II.9}$$

where $g(\mathbf{x}^*)$ is the current best solution chosen from among the true function values at the training points (henceforth referred to as simply g^*). This expectation can then be computed by integrating over the distribution of $\hat{g}(\mathbf{x})$ with g^* held constant:

$$EI\big(\hat{g}(\mathbf{x})\big) = \int_{-\infty}^{g^*} \left(g^* - g\right)\, f_{\hat{g}}\, dg \tag{II.10}$$

where g is a realization of $f_{\hat{g}}$. This integral can be expressed analytically as:[47]

$$EI\big(\hat{g}(\mathbf{x})\big) = \left(g^* - \mu_{\hat{g}}\right)\, \Phi\left(\frac{g^* - \mu_{\hat{g}}}{\sigma_{\hat{g}}}\right) + \sigma_{\hat{g}}\, \phi\left(\frac{g^* - \mu_{\hat{g}}}{\sigma_{\hat{g}}}\right) \tag{II.11}$$

where it is understood that μ_g and σ_g are functions of $\mathbf{x}$. A detailed derivation of this equation is provided in Appendix A.1.

The point at which the EIF is maximized is selected as an additional training point. With the new training point added, a new GP model is built and then used to construct another EIF, which is then used to choose another new training point, and so on, until the value of the EIF at its maximized point is below some specified tolerance.

While not mentioned in Ref. 47, it is clear from Eq. II.11 that the magnitude of the EIF will be affected by the magnitude of the underlying objective function $g(\mathbf{x})$. Because the magnitude of the maximum EIF is used as the convergence criterion, this can cause problems in assessing the convergence of EGO. If the objective function has a very small response, EGO may appear to converge very rapidly; large response values may lead the method to never converge. To overcome this, a scaling factor is introduced. Instead of basing convergence on the EIF as shown, it is first scaled by the absolute value of the constant term from the trend of the underlying GP model, i.e. $EI = EI/|\boldsymbol{\beta}(1)|$.

It is important to understand how the use of this EIF leads to optimal solutions. Eq. II.11 indicates how much the objective function value at $\mathbf{u}$ is expected to be less than the predicted value at the current best solution. It contains a balance between exploiting regions of the design space where good solutions have been discovered, and exploring regions that have not been well sampled and thus have greater uncertainty. Because the GP model provides a Gaussian distribution at each predicted point, expectations can be calculated. Points with good predicted values and even a small variance will have a significant expectation of producing a better solution (exploitation), but so will points that have relatively poor predicted values and greater variance (exploration). The following simple example provides an illustration of how the EGO algorithm leads to optimal solutions.

II.1.3 Simple EGO Example

This problem involves minimizing the function

$$f = 10\,\frac{\sin\frac{5x}{2} + 2}{x^2 + 4} \tag{II.12}$$

over the bounds $-1.5 \leq x \leq 3.0$ as shown in Figure II.2. The first step is to randomly draw a small number of samples from this function and construct a GP from those samples. Here, four initial samples are used. The resulting GP model is shown in Figure II.3 where the solid line is the mean prediction, the dotted line is the underlying true function (unknown to the optimizer), and the dashed lines are the 90% confidence bounds of the GP predictor. Next, the point at which the EIF is maximized is found. Figure II.4 adds the initial EIF to the previous plot.

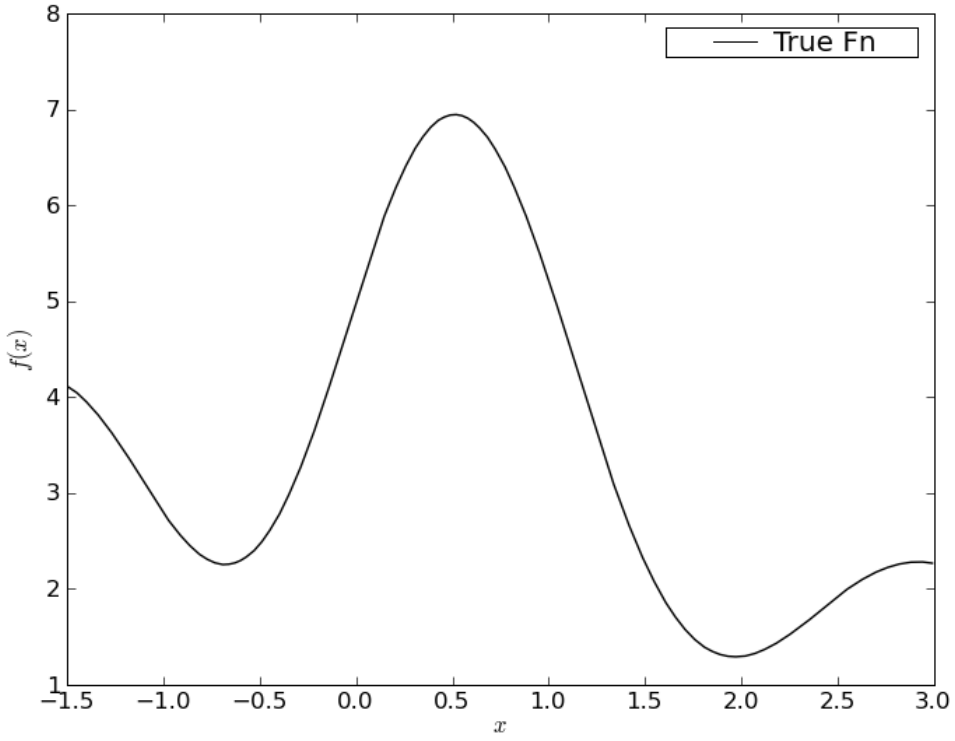


Figure II.2: Simple EGO example - True function to be minimized.

This EIF is clearly multimodal with four local optima (at approximately -1.1, -0.2, 2.3, and 3.0), with the global optimum at approximately -1.1. This point appears favorable to the EGO algorithm because there is a known good value nearby and there

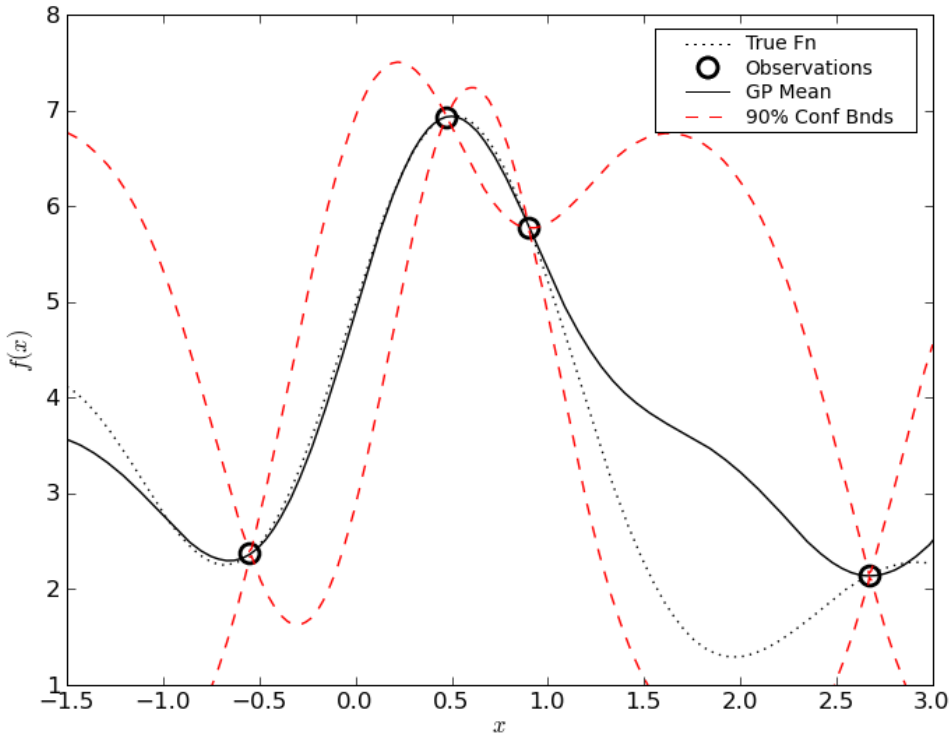


Figure II.3: Simple EGO example - Initial GP model.

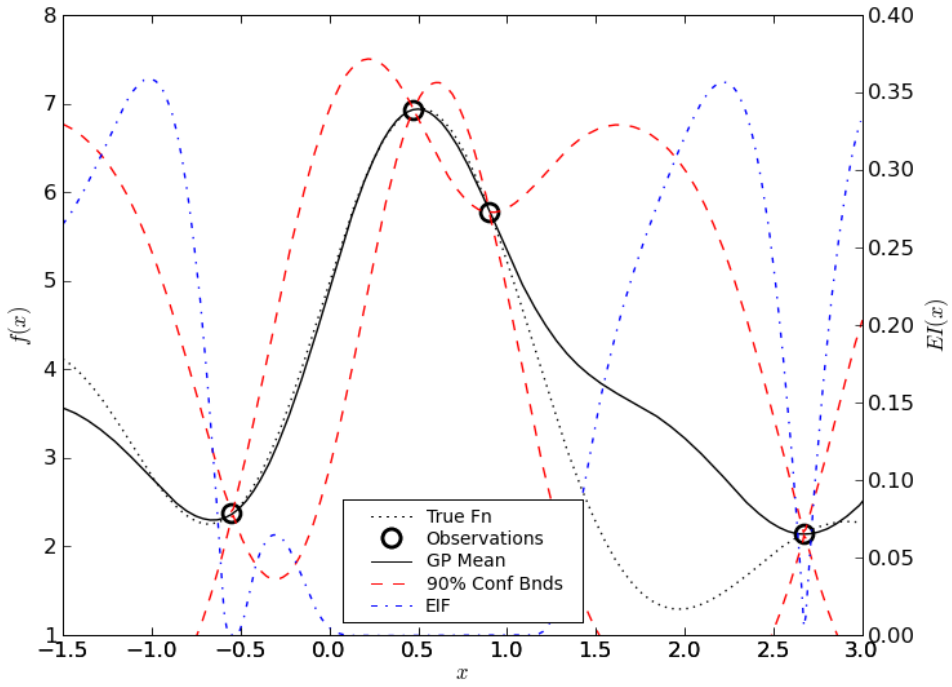


Figure II.4: Simple EGO example - Initial EIF.

is considerable variance in the GP model. The true function is evaluated at this maximization point, its response is added to the training data, and a new GP and EIF are

constructed. This completes the first iteration of the EGO algorithm and the new state of knowledge is shown in Figure II.5. Note that after the new point was added, the EIF in this region has been drastically reduced because the true value at this point is worse than the current optimal solution. This "bad" value, plus the now-reduced variance, lead to a low expectation of finding an improved value in this region. The maximum EIF point is now around 2.2. The predicted value in this region is near the current best point, and the variance in the GP is high; thus the EIF is high.

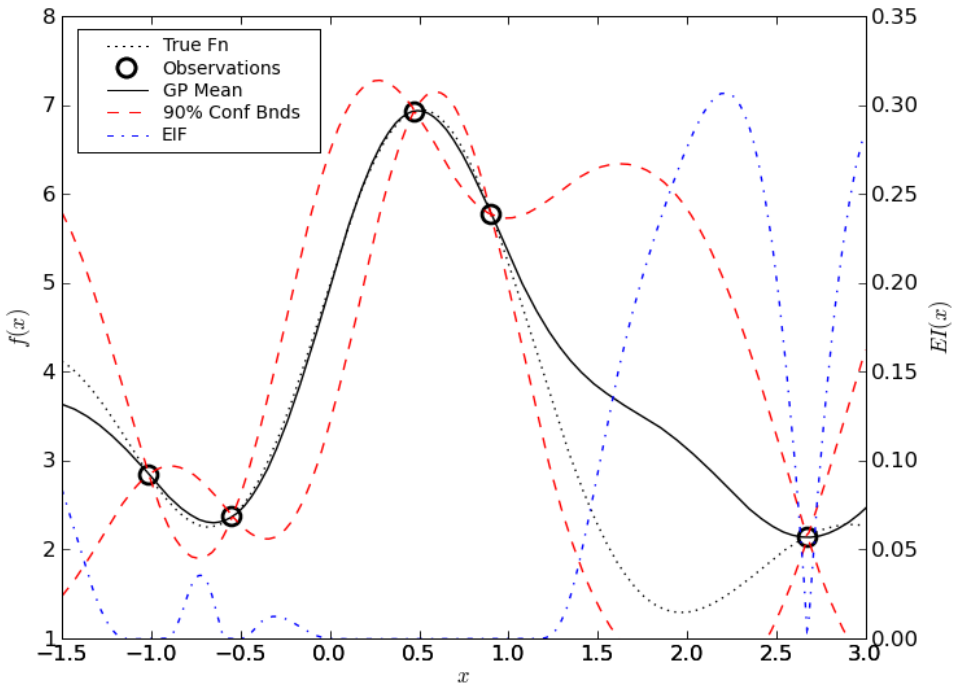
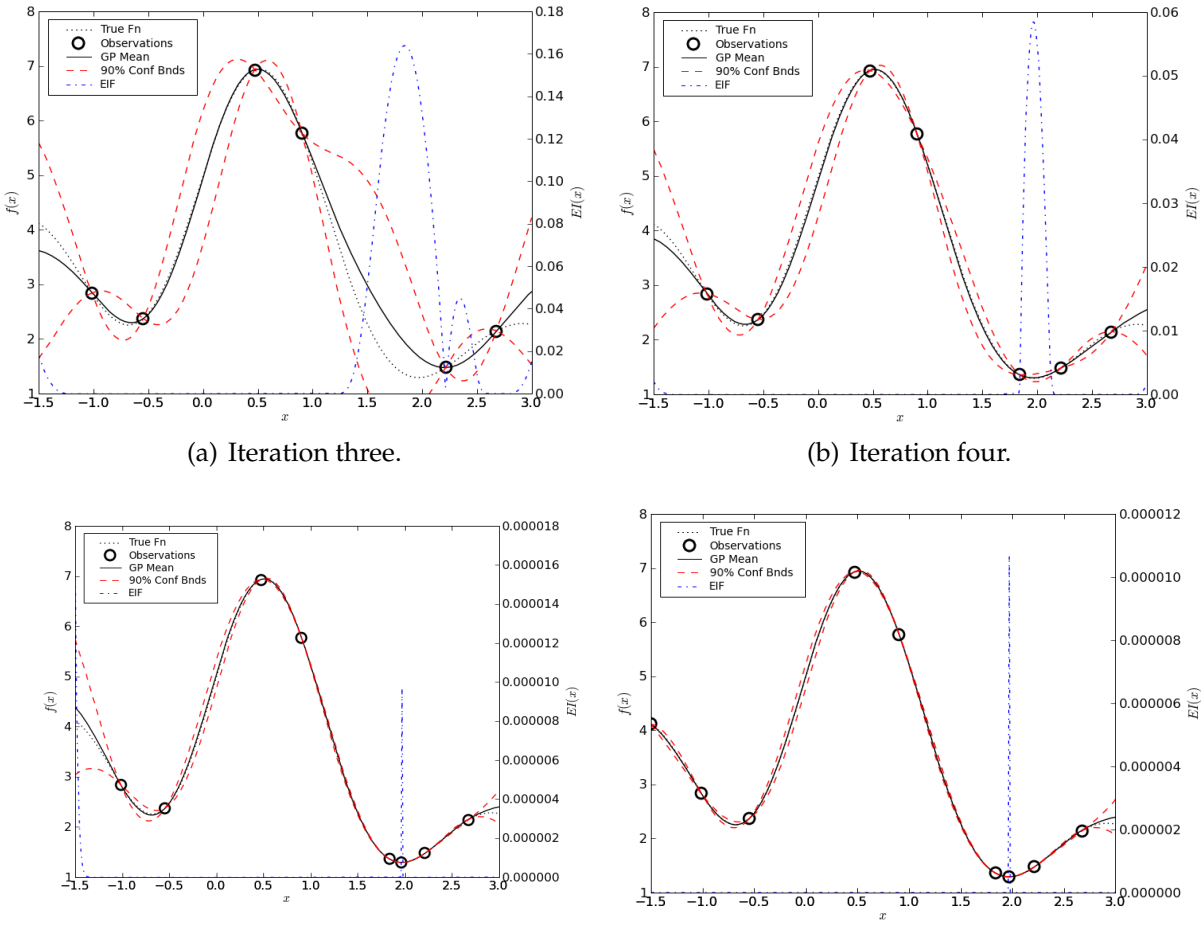


Figure II.5: Simple EGO example - Iteration two.

Figure II.6 shows the next four iterations. At each iteration, the EIF is formed and maximized, the maximizing point is evaluated on the true function, this data is added to the GP and a new model is built. Note that at each iteration, the value of the EIF at the maximizing point (shown on the scale on the right of each plot) is reduced in each iteration. After just six iterations (ten total function evaluations), the maximum EIF value is sufficiently low for EGO to converge, and the global minimum of this function has been found. Figure II.7 shows the final state of knowledge for this problem.

The application of EGO to reliability analysis, however, is made more complicated



(a) Iteration three.

(b) Iteration four.

(c) Iteration five.

(d) Iteration six.

Figure II.6: Simple EGO example - Iterations 3-6.

because the response function appears within the equality constraint rather than the objective (see Eq. I.8). In this case, the maximization of the EIF is inappropriate because feasibility is the main concern. This application is therefore a significant departure from the intentions of EGO and requires a new formulation. For this problem, the expected feasibility function is introduced below.

II.2 Expected Feasibility Function

The expected improvement function provides an indication of how much the true value of the response at a point can be expected to be less (or more) than the current best solution. It therefore makes little sense to apply this to the forward reliability problem where the goal is not to minimize the response, but rather to find where

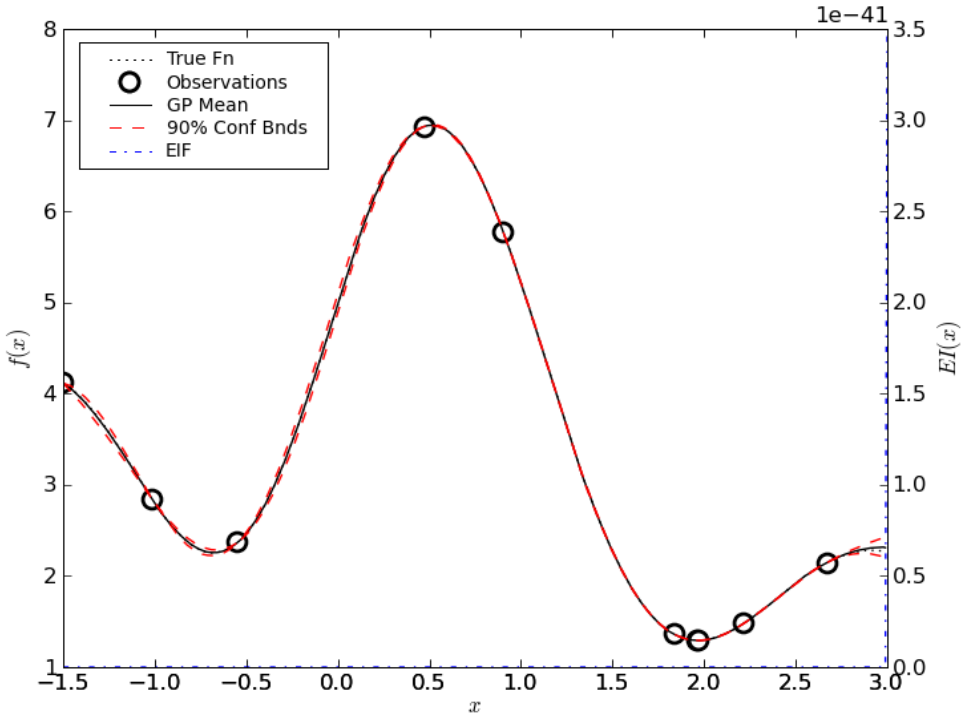


Figure II.7: Simple EGO example - Final iteration.

it is equal to a specified threshold value. The expected feasibility function (EFF) is introduced here to provide an indication of how well the true value of the response is expected to satisfy the equality constraint $g(\mathbf{x}) = \bar{z}$. Inspired by the contour estimation work in Ref. 66, this expectation can be calculated in a similar fashion as Eq. II.10 by integrating over a region in the immediate vicinity of the threshold value $\bar{z} \pm \epsilon$:

$$EF\big(\hat{g}(\mathbf{x})\big) = \int_{\bar{z}-\epsilon}^{\bar{z}+\epsilon} \big[\epsilon - |\bar{z} - g|\big]\, f_{\hat{g}}\, dg \tag{II.13}$$

where g denotes a realization of the distribution $f_{\hat{g}}$, as before. Allowing z^+ and z^- to denote $\bar{z} \pm \epsilon$, respectively, this integral can be expressed analytically as:

$$\begin{aligned} EF\big(\hat{g}(\mathbf{x})\big) = (\mu_g - \bar{z}) &\left[2\,\Phi\left(\frac{\bar{z}-\mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right)\right] \\ &- \sigma_g \left[2\,\phi\left(\frac{\bar{z}-\mu_g}{\sigma_g}\right) - \phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) - \phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right)\right] \\ &+ \quad \epsilon \left[\Phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^- - \mu_g}{\sigma_g}\right)\right] \end{aligned} \tag{II.14}$$

where ϵ is proportional to the standard deviation of the GP predictor ($\epsilon \propto \sigma_g$). In this case, z^-, z^+, μ_g, σ_g, and ϵ are all functions of the location $\mathbf{x}$, while $\bar{z}$ is a constant. The derivation of this function is detailed in Appendix A.2. Note that the EFF provides the same balance between exploration and exploitation as is captured in the EIF. Points where the expected value is close to the threshold ($\mu_g \approx \bar{z}$) and points with a large uncertainty in the prediction will have large expected feasibility values. Like the EIF, magnitude of the EFF will be affected by the magnitude of the underlying response function $g(\mathbf{x})$. To prevent this from influencing the convergence, the function is scaled in the same way that the EIF was, i.e. $EF = EF/|\boldsymbol{\beta}(1)|$.

II.3 Efficient Global Reliability Analysis Algorithm

The following process makes up the EGRA algorithm:

1. Generate a small number of random samples from the true response function.
   (a) Only $\frac{(n+1)(n+2)}{2}$ samples are used (where n is the number of random variables). This initial selection is arbitrary, but the number of samples required to define a quadratic polynomial is used as a convenient rule of thumb.
   (b) The samples uniformly span the random variable space over the bounds $\pm 5\sigma$, though the bounds of this search space can be adjusted if needed.
   (c) Latin hypercube sampling (LHS) is used to generate the samples.
2. Construct an initial Gaussian process model from these samples.
3. Find the point with maximum expected feasibility.
   (a) The expected feasibility function is built with $\epsilon = 2\sigma_g$.
   (b) To locate the global optimum of this multimodal function is, the DIRECT[34] method is used.
   (c) If the scaled maximum expected feasibility is less than 1E-5, the model has converged. Go to step 6.

4. Evaluate the true response function at this point.

5. Add this new sample to the previous set of training data and build a new GP model. Go to step 3.

6. This surrogate model is then used to calculate the probability of failure using any sampling method.

In Step 3, DIRECT[34] is used to maximize the EFF. This method performs a relatively exhaustive search, first subdividing the search space, then preferentially performing further subdivisions in regions where good solutions have been found, but also dividing regions that are much larger than others to ensure that these regions are searched. This iterative procedure of DIviding RECTangles gives the method its name. By balancing the exploitation of promising regions and the exploration of unsampled regions (much like EGO), this method has been shown to reliably locate the global optimum. Locating the true global optimum, however, is not strictly necessary. As the EGO demonstration problem in the previous section showed, points that are good local solutions at a given iteration will likely remain good solutions in subsequent iterations. The order in which these local optima are added to the GP model is not important, so at any iteration, locating a good local optimum would be sufficient. While gradient-based methods are generally capable of accurately locating local optima (and the analytic gradient of the EIF/EFF can be derived) they cannot be applied to this problem because of the large regions of the EIF/EFF with zero gradient (see Figure II.6). In short, DIRECT is used in this work, but any gradient-free global optimization method capable of finding a good local optima could be used in its place.

The Gaussian process model created by EGRA provides a unique opportunity for applying multimodal adaptive importance sampling (MAIS) because the model possesses multiple points on or near the limit state with which to construct the multimodal sampling density. Previous uses of MAIS have started with only the MPP, and require multiple iterations of searching just to locate the representative points.[20,94]

Additionally, MAIS would not be easy with methods that use a random selection of true samples with which to construct the GP model because several iterations would be necessary to locate the limit state. The main use of MAIS has been to reduce the sampling cost,[20,94] but it can also be more accurate than using even a large number of Monte Carlo or LHS samples if enough evaluations can be afforded to allow the method to converge. Because all of the MAIS samples are evaluated using the GP model, they can be provided at little cost.

While EGO/EGRA will typically need far fewer evaluations of the true objective/response function than other methods, they both require a non-trivial amount of overhead computation to construct the GP models and to solve the global optimization problem to maximize the EIF/EFF. If the computational expense of the objective/response function is small in relation to this overhead, then using these methods to reduce the number of times this function is evaluated makes little sense. However, for computationally expensive models (such as large finite element models), reducing the number of evaluations is paramount. It is this class of problems to which the application of EGO/EGRA is intended.

II.4 Computational Experiments

This section presents the application of EGRA to several test problems, and compares its performance in terms of efficiency and accuracy to several previously available methods including FORM, SORM, AMV^2+, TANA, and exhaustive LHS sampling. For each test, once the GP model has converged, two types of sampling are explored: MAIS and LHS using one million samples. As discussed, because MAIS is a convergent method, it may provide additional accuracy. Using LHS sampling gives a more direct comparison to the LHS tests on the true response functions.

II.4.1 Multimodal Function

The first problem has a highly nonlinear and multimodal response defined by:

$$g(\mathbf{x}) = \frac{\left(x_1^2 + 4\right)(x_2 - 1)}{20} - \sin\frac{5x_1}{2} - 2 \tag{II.15}$$

The distribution of x_1 is Normal($\mu = 1.5$, $\sigma = 1$) and x_2 is Normal($\mu = 2.5$, $\sigma = 1$); the variables are uncorrelated. The response level of interest for this study is $\bar{z} = 0$ with failure defined by $g > \bar{z}$. This problem can be formulated as:

$$p_f = P\left[g(\mathbf{x}) > 0\right] \tag{II.16}$$

Figure II.8 shows a contour plot in x-space of this response function throughout the ± 5 standard deviation search space. This function clearly has several local optima to the forward-reliability MPP search problem (see Eq. I.8).

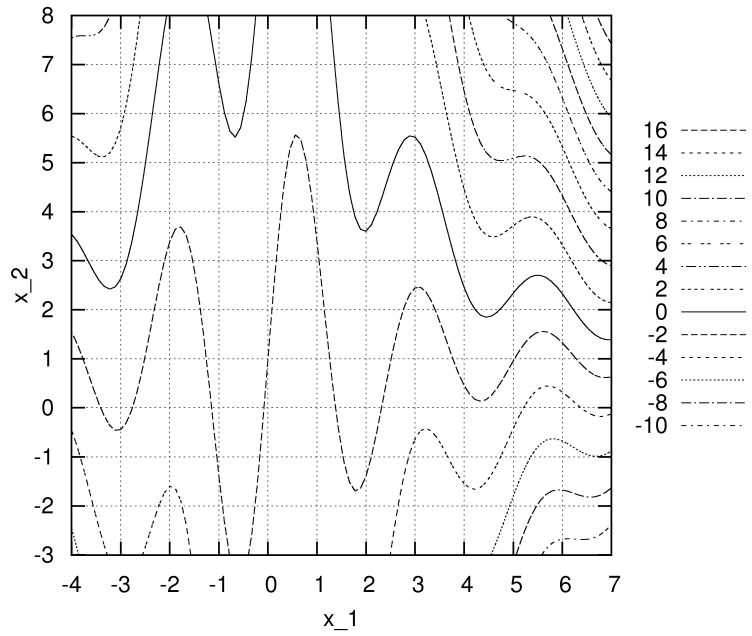


Figure II.8: Contour plot of the multimodal function. The solid line is $g = \bar{z} = 0$.

To illustrate how the EGRA method works, this problem is explored in detail. A GP model was built with 10 randomly selected samples. Note that this is larger than the recommened number of intial samples; this is done only to enhance visualization

in the early stages of the process. Figure II.9 shows contours of the mean value, the variance, and the expected feasibility function for this set of samples.

Comparing the mean value plot to the true contours in Figure II.8 shows that this is a poor approximation. Note that the variance is very low near the training data, but is large throughout the rest of the space. The ⋆ [near (-3.5,2.4)] on the expected feasibility function contour shows the point that maximizes the function and is the new sample that will be added at the next iteration. Selecting this point is a combination of the mean value and the variance, but at this stage the large variance dominates and the new point is chosen to explore the search space. Figure II.10 shows the same contours after this new point has been added to the training data.

This is clearly still a poor approximation to the multimodal function. Again, the uncertainty in the model dominates the expected feasibility and the maximum point [at (6.5,7.5)] is chosen to explore. The next few samples follow in a similar fashion, all chosen to explore due to the high uncertainty from building a GP model with such a small amount of data. Figure II.11 shows the mean, variance, and expected feasibility contours after 5 points have been added (15 total samples in the training data).

The new point selected at this iteration (indicated by the ⋆ on the expected feasibility contour plot) is at a point with low variance, but with an expected value very near the limit state (the solid line on the mean value contour plot). The uncertainty in the model has dropped to a point where the exploitive terms of the EFF have a considerable effect. Note also that the value of the expected feasibility is much smaller than it was with just 5 fewer samples, showing that the method is converging. Figure II.12 shows the contours when there are 30 total samples.

At this point, the mean value plot is very close to the true contour and the variance is very low. The expected feasibilty function is shown without the samples so as not to hide the very tight contours. The Gaussian process model "knows" where the limit state is, so all subsequent samples are "exploitation" samples. The samples chosen are selected by the little uncertainty that still remains. Because the variance is larger as

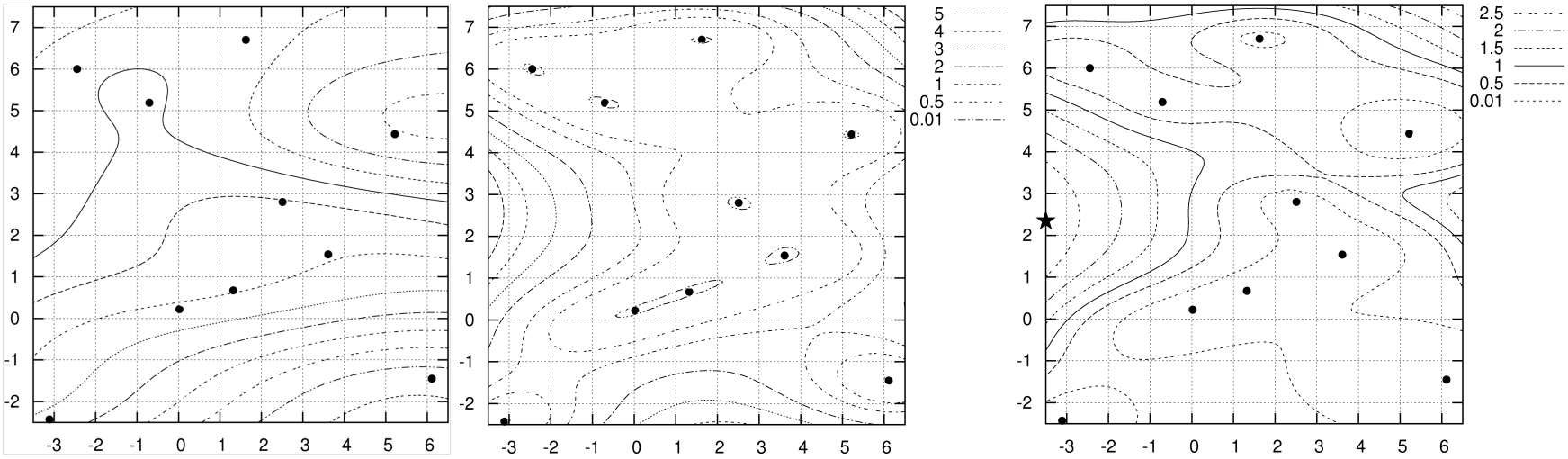


Figure II.9: Contours of the mean value (solid line is the limit state), variance, and expected feasibility function for 10 initial samples. Dots represent the samples used to create the GP; ⋆ is the max(EFF) point.

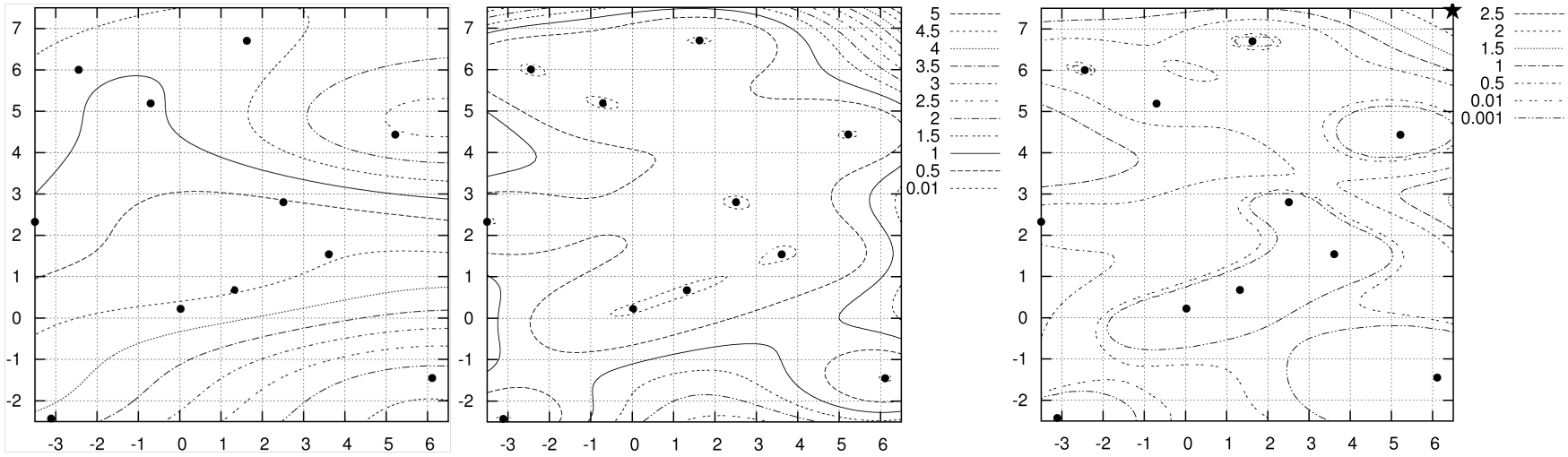


Figure II.10: Contours of the mean value (solid line is the limit state), variance, and expected feasibility function for 11 samples. Dots represent the samples used to create the GP; ⋆ is the max(EFF) point.

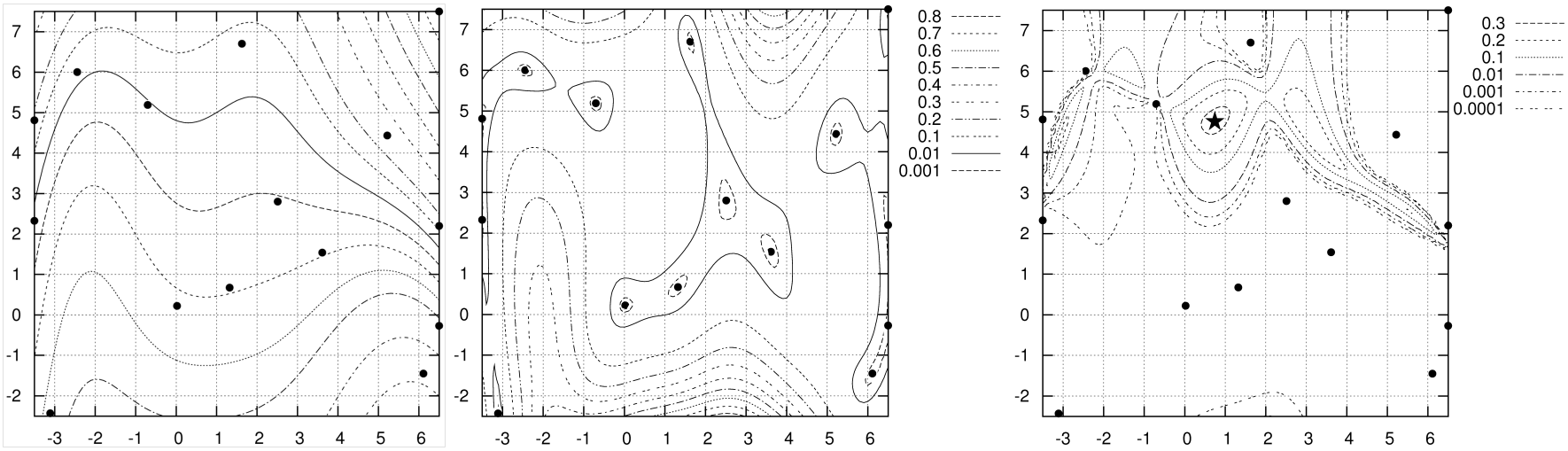


Figure II.11: Contours of the mean value (solid line is the limit state), variance, and expected feasibility function for 15 samples. Dots represent the samples used to create the GP; ⋆ is the max(EFF) point.

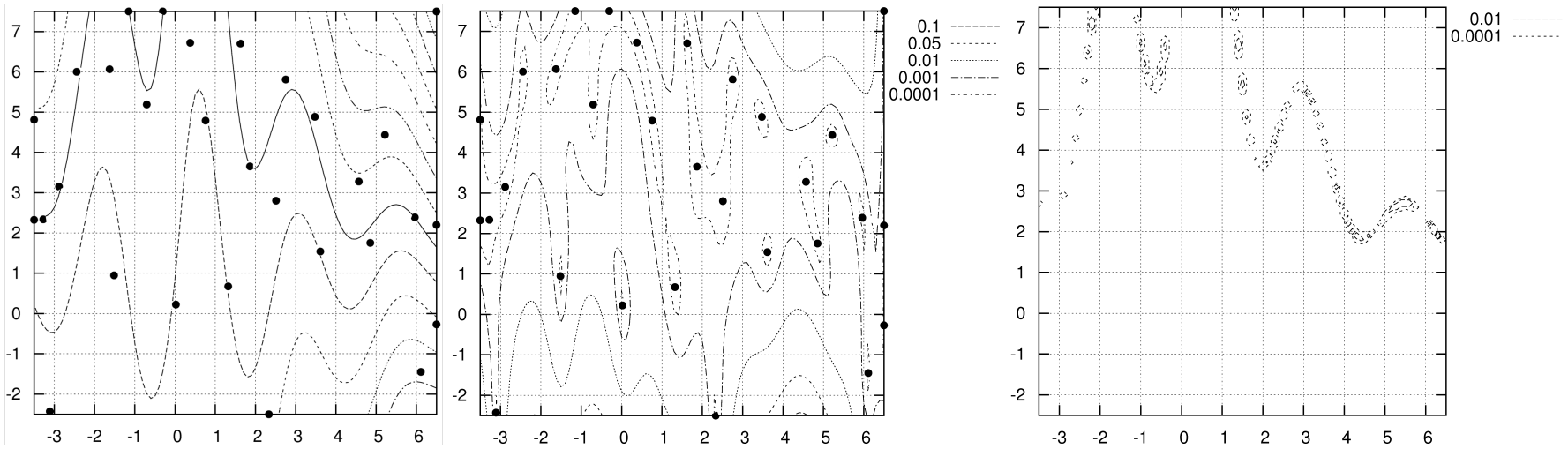


Figure II.12: Contours of the mean value (solid line is the limit state), variance, and expected feasibility function for 30 samples. Dots represent the samples used to create the GP.

one moves away from the sample data, this ensures that samples are not chosen too close to one another, protecting the structure of the GP. Figure II.13 shows the final contour plot of the expected values of the GP model. Note that all of the samples added since Figure II.12 lie almost exactly on the limit state and in general, the much larger density of samples near the limit state than elsewhere in the search space.

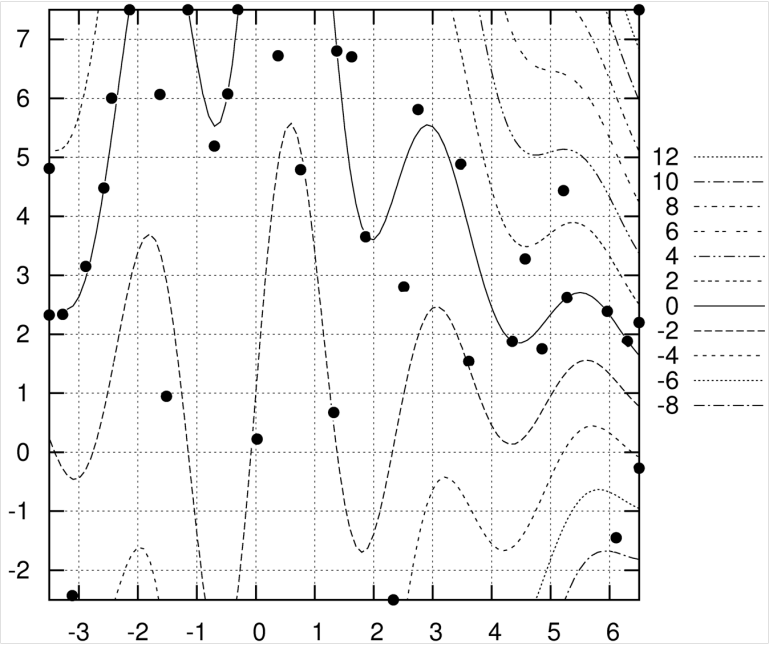


Figure II.13: Final contour of the mean value (solid line is the limit state) with 37 samples. Dots represent the samples used to create the GP.

This problem was also solved using reliability methods that reduce the cost of the MPP search through the use of local surrogate models. Two response function approximation methods were investigated:[24] second-order iterated Advanced Mean Value (AMV^2+) and Two-point Adaptive Nonlinear Approximation (TANA).[82,93]

A case using no response function approximation was also investigated. Combining this with first-order or second-order integration results in the traditional FORM and SORM formulations. To produce results consistent with an implicit response function, numerical gradients and quasi-Newton Hessians from Symmetric Rank 1 updates were used. For each method, at the converged MPP, both first-order and second-order integration were used to calculate the probability.

For each method, the only algorithmic variation explored here is that the local

surrogate models can be built in either x-space or u-space. Determining which space is appropriate depends upon the form of the response and the space transformation. This choice can have significant effects on both accuracy and efficiency for methods that use low-order approximations to the response function. For instance, if a linear approximation is used for a response that is linear in x-space but nonlinear in u-space, then building the approximation in x-space will yield better results. Gaussian process models are not greatly affected by this choice because they do not rely on curve-fitting or any assumptions on the shape of the response, so EGRA is only performed on models built in x-space.

Table II.1 gives a summary of the results from all methods. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Because EGRA is stochastic, it was also run 20 times and the average probabilities are reported. Two sampling methods are explored in combination with EGRA - MAIS and LHS using one million samples. For each of the 20 runs of EGRA, both sampling methods were used on the same converged model. This way, the only difference between their results is the sampling method employed, eliminating any variation in the GP models, providing a more direct comparison of the sampling methods. To measure the accuracy of the method, two errors are reported for the EGRA results: the error in the average probability, and the average of the absolute errors from 20 independent simulations. For comparison, the same errors are given for 20 runs of LHS studies comprised of 10,000 and 100,000 samples.

Most of the MPP search methods converge to the same MPP and thus report the same probability. Note, however, that the x-space TANA results are not included because the method failed to converge. The probabilities are more accurate when second-order integration is used, but still have significant errors (20%). For this multimodal problem, EGRA is more expensive than AMV^2+, but cheaper than all the

Table II.1: Results for the multimodal problem.

Reliability Method	Function Evaluations	First-Order p_f (% Error)	Second-Order p_f (% Error)	Sampling p_f (% Error, Avg. Error)
No Approximation	70	0.11797 (277.0%)	0.02516 (-19.6%)	—
x-space AMV^2+	26	0.11797 (277.0%)	0.02516 (-19.6%)	—
u-space AMV^2+	26	0.11777 (277.0%)	0.02516 (-19.6%)	—
u-space TANA	131	0.11797 (277.0%)	0.02516 (-19.6%)	—
LHS solution	10k	—	—	0.03117 (0.385%, 2.847%)
LHS solution	100k	—	—	0.03126 (0.085%, 1.397%)
LHS solution	1M	—	—	0.03129 (“truth”, 0.339%)
EGRA (MAIS)	34.9*	—	—	0.03129 (0.006%, 0.954%)
EGRA (1M LHS)	34.9*	—	—	0.03132 (0.084%, 0.370%)

*The average number of evaluations is reported for EGRA.

other methods. More importantly, EGRA is far more accurate than all but the most expensive sampling method. EGRA coupled with either the MAIS or LHS sampling provides accurate results (both have an average absolute error <1%), but the average absolute error from the LHS samples is approximately the same as that generated by the same number of LHS samples on the true function. This indicates that the GP model produced by EGRA is an extremely accurate representation of the true function. Any uncertainty that may remain in the converged model is far outweighed by the sampling variance.

II.4.2 Cubic Function

The second example is a two-dimensional nonlinear function from Ref. 94.

$$g(\mathbf{x}) = x_1^3 + x_2^3 - 18 \tag{II.17}$$

The distribution of x_1 is Normal($\mu = 10$,$\sigma = 5$) and x_2 is Normal($\mu = 9.9$,$\sigma = 5$); the variables are uncorrelated. The response level of interest for this study is $\bar{z} = 0$ with failure defined by $g < \bar{z}$. The problem formulation is then:

$$p_f = P[g(\mathbf{x}) < 0] \tag{II.18}$$

This problem was introduced by Zou et al.[94] to test a method that used a trust-region managed adaptive response surface method to locate the MPP and then used first-order integration, second-order integration, and multimodal adaptive importance sampling (MAIS) to calculate the probability of failure.

Table II.2 gives a summary of the results from the same methods investigated in the previous example plus the published results from Ref. 94. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Again, two errors are reported for

EGRA and LHS: the error in the average probability, and the average of the absolute errors from 20 independent simulations.

This problem only has one significant MPP, so the large disparity in some of the local search methods clearly indicates convergence to different points. It is interesting to note that for this problem, x-space TANA provides the most efficient solution while for the previous problem it failed to converge. Once again, second-order integration provides better results, but is still not an accurate approximation to the true shape of the limit state, so there are still large errors. Because this test problem is not multimodal, performing MAIS with only the MPP as a starting point (as is done by Zou et al.[94]) is sufficient to capture the higher level of nonlinearity in the limit state and generate an excellent result. However, if this method were applied to the previous test problem, it would likely be either much less accurate or require a substantial increase in cost in order to adequately locate and sample the other significantly probable regions of the space. It should also be pointed out that despite MAIS being a stochastic method, only the error for a single result is reported by Zou et al.[94] and not an average absolute error as is included for the other sampling methods. For this nonlinear problem, EGRA is less expensive than all the other methods, and provides much more accurate results. As with the previous problem, there are is not a large difference between using MAIS or LHS to perform the sampling once the GP model has converged.

II.4.3 Cantilever Beam

This problem involves the simple uniform cantilever beam[24,76,88] shown in Figure II.14.

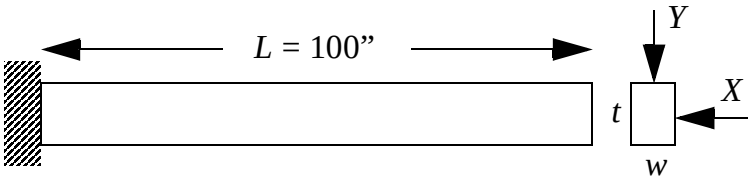


Figure II.14: Schematic of the cantilever beam example problem.

Table II.2: Results for the cubic function problem.

Reliability Method	Function Evaluations	First-Order p_f (% Error)	Second-Order p_f (% Error)	Sampling p_f (% Error, Avg. Error)
No Approximation	125	0.01301 (127.1%)	0.004164 (-27.3%)	—
x-space AMV^2+	66	0.01301 (127.1%)	0.004165 (-27.3%)	—
u-space AMV^2+	66	0.01301 (127.1%)	0.004165 (-27.3%)	—
x-space TANA	21	0.01301 (127.1%)	0.004157 (-27.4%)	—
u-space TANA	36	0.01301 (127.1%)	0.004165 (-27.3%)	—
Zou et al. MAIS	97	0.02560 (346.9%)	0.016200 (182.8%)	—
LHS solution	10k	—	—	0.005590 (2.413%, 7.279%)
LHS solution	100k	—	—	0.005686 (0.746%, 3.637%)
LHS solution	1M	—	—	0.005728 (“truth”, 0.757%)
Zou et al. MAIS	560	—	—	0.005750 (0.380%, no data)
EGRA (MAIS)	17.9*	—	—	0.005598 (2.268%, 2.457%)
EGRA (1M LHS)	17.9*	—	—	0.005571 (2.750%, 2.812%)

*The average number of evaluations is reported for EGRA.

Random variables in the problem include the yield stress R of the beam material, the Young's modulus E of the material, and the horizontal and vertical loads, X and Y, which are modeled with normal distributions using Normal($\mu = 40000$, $\sigma = 2000$), Normal($\mu = 2.9E7$, $\sigma = 1.45E6$), Normal($\mu = 500$, $\sigma = 100$), and Normal($\mu = 1000$, $\sigma = 100$), respectively. Problem constants include L = 100 in. and D_0 = 2.2535 in. The design of the beam is assumed to be $w = 2.6041$ and $t = 3.6746$. The constraints on beam response have the following analytic form:

$$stress = \frac{600}{wt^2}Y + \frac{600}{w^2t}X \leq R \tag{II.19}$$

$$displacement = \frac{4L^3}{Ewt}\sqrt{(\frac{Y}{t^2})^2 + (\frac{X}{w^2})^2} \leq D_0 \tag{II.20}$$

or when scaled:

$$g_S = \frac{stress}{R} - 1 \leq 0 \tag{II.21}$$

$$g_D = \frac{displacement}{D_0} - 1 \leq 0 \tag{II.22}$$

Tables II.3 and II.4 give summaries of the results from the same methods investigated in the previous examples. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Again, two errors are reported for EGRA and LHS: the error in the average probability, and the average of the absolute errors from 20 independent simulations.

The stress response is well captured by a linear approximation, which can be seen in the similarity between the first- and second-order integration results and their accuracy when compared to the LHS sampling. This simple functional form can be easily captured with a GP model, so EGRA rarely has to add any additional data beyond its initial samples to construct an accurate model ($\frac{(3+1)(3+2)}{2} = 10$). Even at this extremely

Table II.3: Results for the stress function of the cantilever beam problem.

Reliability Method	Function Evaluations	First-Order p_f (% Error)	Second-Order p_f (% Error)	Sampling p_f (% Error, Avg. Error)
No Approximation	63	0.00112 (-0.148%)	0.00112 (-0.148%)	—
x-space AMV^2+	82	0.00112 (-0.148%)	0.00112 (-0.148%)	—
u-space AMV^2+	82	0.00112 (-0.148%)	0.00112 (-0.148%)	—
x-space TANA	73	0.00112 (-0.148%)	0.00112 (-0.148%)	—
u-space TANA	91	0.00112 (-0.148%)	0.00112 (-0.148%)	—
LHS solution	10k	—	—	0.00116 (3.065%, 18.11%)
LHS solution	100k	—	—	0.00111 (1.397%, 7.139%)
LHS solution	1M	—	—	0.00112 (“truth”, 1.551%)
EGRA (MAIS)	10.3*	—	—	0.00111 (1.234%, 1.500%)
EGRA (1M LHS)	10.3*	—	—	0.00111 (1.084%, 2.635%)

*The average number of evaluations is reported for EGRA.

Table II.4: Results for the displacement function of the cantilever beam problem.

Reliability Method	Function Evaluations	First-Order p_f (% Error)	Second-Order p_f (% Error)	Sampling p_f (% Error, Avg. Error)
No Approximation	72	0.000183 (-8.974%)	0.000197 (-1.770%)	—
x-space AMV^2+	73	0.000183 (-8.974%)	0.000199 (-1.176%)	—
u-space AMV^2+	73	0.000183 (-8.974%)	0.000199 (-1.176%)	—
x-space TANA	82	0.000183 (-8.974%)	0.000198 (-1.346%)	—
u-space TANA	73	0.000183 (-8.974%)	—	—
LHS solution	100k	—	—	0.000208 (3.508%, 14.976%)
LHS solution	1M	—	—	0.000201 (“truth”, 4.810%)
EGRA (MAIS)	24.1*	—	—	0.000210 (4.367%, 4.367%)
EGRA (1M LHS)	24.1*	—	—	0.000212 (5.325%, 7.604%)

*The average number of evaluations is reported for EGRA.

low cost, the accuracy of EGRA is comparable to the one million LHS samples.

The displacement response is not linear, but the second-order integration provides a fairly accurate solution. Note, though, that the u-space TANA solution was unable to provide a second-order result due to numerical problems with the final Hessian. EGRA required only one third of the number of function evaluations of the cheapest MPP-based method, but again produced an accuracy comparable to exhaustive LHS. Because the probability of failure for this problem is small, using EGRA with MAIS provides a more accurate result (on average) than using one million LHS samples.

II.4.4 Short Column

This problem involves the plastic analysis of a short column with rectangular cross section (width b and depth h) having uncertain material properties (yield stress Y) and subject to uncertain loads (bending moment M and axial force P).[24,51] The response function is defined as:

$$g = 1 - \frac{4M}{bh^2Y} - \frac{P^2}{b^2h^2Y^2} \tag{II.23}$$

The distributions for P, M, and Y are Normal(500, 100), Normal(2000, 400), and Log-normal(5, 0.5), respectively, with a correlation coefficient of 0.5 between P and M (uncorrelated otherwise). The design of the column is assumed to be $b = 8.654$ and $h = 25.0$.

Table II.5 gives a summary of the results from the same methods investigated in the previous examples. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Again, two errors are reported for EGRA and LHS: the error in the average probability, and the average of the absolute errors from 20 independent simulations.

The MPP-based methods clearly all converge to the same MPP, and the second-order integration provides a very accurate probability of failure. EGRA outperforms

all methods, requiring about 40% fewer function evaluations than the most efficient MPP-based method. The accuracy of the EGRA results compare well to using 100k LHS samples, but clearly come at a much reduced cost.

II.4.5 Steel Column

This problem involves determining the probability that the stress on a steel column will exceed its yield stress.[51] The response function is dependent on nine random variables of various distributions.

$$g = F_s - P\left(\frac{1}{2BD} + \frac{F_0}{BDH}\frac{E_b}{E_b - P}\right) \tag{II.24}$$

where

$$P = P_1 + P_2 + P_3 \tag{II.25}$$

$$E_b = \frac{\pi^2 EBDH^2}{2L^2} \tag{II.26}$$

and F_s is the yield stress (Lognormal, $\mu/\sigma = 400/35$ MPa), P_1 is the dead weight load (Normal, $\mu/\sigma = 500/50$ kN), P_2 and P_3 are variable loads (Gumbel, $\mu/\sigma = 600/90$ kN), B is the flange breadth (Lognormal, $\mu/\sigma = 200/3$ mm), D is the flange thickness (Lognormal, $\mu/\sigma = 17.5/2$ mm), H is the profile height (Lognormal, $\mu/\sigma = 100/5$ mm), F_0 is the initial deflection (Normal, $\mu/\sigma = 30/10$ mm), E is the elastic modulus (Weibull, $\mu/\sigma = 21/4.2$ GPa), and L is the length of the column (Deterministic, 7.5 m). The response level of interest for this study is $\bar{z} = 0$ with failure defined by $g < \bar{z}$.

This problem presents a significant difficulty for EGRA. For certain combinations of the inputs (when the yield stress and flange thickness are small and the loads are large) the value of g is very small - much smaller than its value elsewhere in the search space. This type of "spike" in the response is impossible for the GP to accurately capture, and when the GP model breaks down, the behavior of EGRA is unpredictable. However, this combination of inputs is well within the failure region and since EGRA

is searching only for the limit state, including this region in the search space is not strictly necessary. Defining the search space as $\pm 5\sigma$ was relatively arbitrary and can easily be modified as needed and can be changed independently for each variable. For this problem, the space is defined as $[-4\sigma,5\sigma]$ for F_s, $[-5\sigma,3\sigma]$ for P_1, P_2, and P_3, and $[-3\sigma,5\sigma]$ for D; all other variables use the usual $\pm 5\sigma$.

Table II.6 gives a summary of the results from the same methods investigated in the previous examples. To establish an accurate estimate of the true solution, 20 independent studies were performed using one million Latin hypercube samples per study. The average probability from these studies is reported as the "true" solution. Again, two errors are reported for EGRA and LHS: the error in the average probability, and the average of the absolute errors from the 20 studies.

For this problem, none of the MPP-based methods provide very accurate results. The AMV^2+ methods converge relatively rapidly to a different (and seemingly erroneous) MPP than the other methods. None could provide second-order reliability estimates.

While EGRA needed some help in defining the search space in order to be able to solve this problem, with that help it produces accurate results. This shows that the size of the problem (nine random variables) is not a hindrance to the method, but that the form of the response function can cause problems if it produces a shape that cannot be modeled by a GP.

II.4.6 Bistable MEMS Device

This application problem involves the validation of previously reported optimal results to the reliability-based design optimization of a bistable MEMS device.[1,2,24,29] The RBDO problem is focused on the shape optimization of compliant bistable mechanisms, where instead of mechanical joints, material elasticity enables the bistability of the mechanism.[6,46,50] Figure II.15(a) contains an electron micrograph of a MEMS compliant bistable mechanism in its second stable position. The first stable position

Table II.5: Results for short column problem.

Reliability Method	Function Evaluations	First-Order p_f (% Error)	Second-Order p_f (% Error)	Sampling p_f (% Error, Avg. Error)
No Approximation	49	0.00639 (4.161%)	0.00614 (0.085%)	—
x-space AMV^2+	92	0.00639 (4.161%)	0.00614 (0.085%)	—
u-space AMV^2+	71	0.00639 (4.161%)	0.00614 (0.085%)	—
x-space TANA	57	0.00639 (4.161%)	0.00614 (0.085%)	—
u-space TANA	64	0.00639 (4.161%)	0.00614 (0.085%)	—
LHS solution	10k	—	—	0.00594 (3.171%, 9.298%)
LHS solution	100k	—	—	0.00616 (0.480%, 2.454%)
LHS solution	1M	—	—	0.00613 (“truth”, 0.808%)
EGRA (MAIS)	26.3*	—	—	0.00615 (0.171%, 2.993%)
EGRA (1M LHS)	26.3*	—	—	0.00614 (0.117%, 2.768%)

*The average number of evaluations is reported for EGRA.

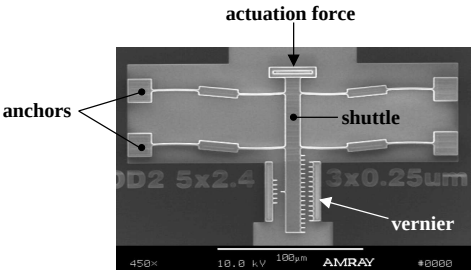
Table II.6: Results for the steel column test problem.

Reliability Method	Function Evaluations	First-Order p_f (% Error)	Second-Order p_f (% Error)	Sampling p_f (% Error, Avg. Error)
No Approximation	190	0.001033 (-31.35%)	—	—
AMV^2+x	58	0.000241 (-83.95%)	—	—
AMV^2+u	77	0.000416 (-72.37%)	—	—
TANA-x	210	0.001033 (-31.35%)	—	—
TANA-u	172	0.001033 (-31.35%)	—	—
LHS solution	10k	—	—	0.001570 (0.067%, 21.19%)
LHS solution	100k	—	—	0.001608 (2.320%, 5.422%)
LHS solution	1M	—	—	0.001571 (“truth”, 1.811%)
EGRA (MAIS)	80.7*	—	—	0.001530 (2.597%, 5.380%)
EGRA (1M LHS)	80.7*	—	—	0.001508 (3.958%, 5.428%)

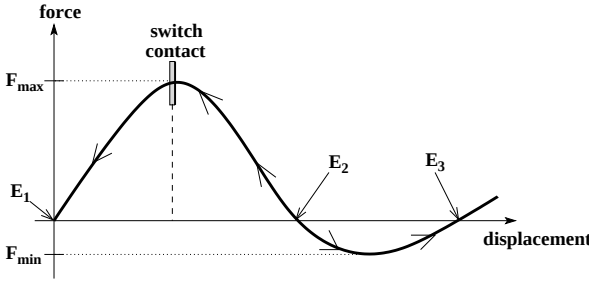
*The average number of evaluations is reported for EGRA.

is the as-fabricated position. One achieves transfer between stable states by applying force to the center shuttle via a thermal actuator, electrostatic actuator, or other means to move the shuttle past an unstable equilibrium, thus making it useful as a micro switch, relay, or nonvolatile memory.

Bistable switch actuation characteristics depend on the relationship between actuation force and shuttle displacement for the manufactured switch. Figure II.15(b) contains a schematic of a typical force–displacement curve for a bistable mechanism. The switch characterized by this curve has three equilibria: E_1 and E_3 are stable equilibria whereas E_2 is an unstable equilibrium (arrows indicate stability). A device with such a force–displacement curve could be used as a switch or actuator by setting the shuttle to position E_3 as shown in Figure II.15(a) (requiring large actuator force F_{max}) and then actuating by applying the comparably small force F_{min} in the opposite direction to transfer back through E_2 toward the equilibrium E_1. One could utilize this force profile to complete a circuit by placing a switch contact near the displaced position corresponding to maximum (closure) force as illustrated. Repeated actuation of the switch relies on being able to reset it with actuation force F_{max}.



(a) Scanning electron micrograph of a MEMS bistable mechanism in its second stable position. The attached vernier provides position measurements.

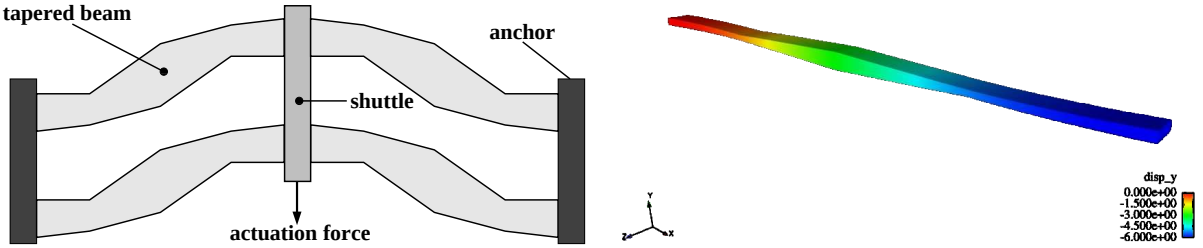


(b) Schematic of force–displacement curve for bistable MEMS mechanism. The arrows indicate stability of equilibria E_1 and E_3 and instability of E_2.

Figure II.15: Bi-stable MEMS mechanism.

The device design considered in Refs. 1, 2, 24, 29 is similar to that in the electron micrograph in Figure II.15(a), for which design optimization has been previously considered,[46] as has robust design under uncertainty with mean value methods.[85] The primary structural difference in the present design is the tapering of the legs, shown

schematically in Figure II.16(a). Figure II.16(b) shows a scale drawing of one tapered beam leg (one quarter of the full switch system). A single leg of the device is approximately 100 μm wide and 5–10 μm tall. This topology is a cross between the



(a) Schematic of a tapered beam bistable mechanism in as-fabricated position (not to scale).

(b) Scale rendering of tapered beam leg for bistable mechanism.

Figure II.16: Tapered beams for bistable MEMS mechanism.

fully compliant bistable mechanism reported in Ref. 46 and the thickness-modulated curved beam in Ref. 62. This tapered geometry offers many degrees of freedom for design, which are described in detail in Section V.3.5 where the full RBDO problem is considered.

Due to manufacturing processes, fabricated geometry can deviate significantly from design-specified beam geometry. As a consequence of photo lithography and etching processes, fabricated in-plane geometry edges (contributing to widths and lengths) can be $0.1 \pm 0.08\,\mu m$ less than specified. This variation in the manufactured geometry leads to substantial variation in the positions of the stable equilibria and in the maximum and minimum force on the force–displacement curve. The manufactured thickness of the device is also variable, though this does not contribute as much to variability in the force–displacement behavior. Variable material properties such as Young's modulus and residual stress also influence the characteristics of the fabricated beam. For this application, two key random variables are considered: ΔW (edge bias on beam widths, which yields effective manufactured widths of $W + \Delta W$) and S_r (residual stress in the manufactured device). The distribution of ΔW is Normal($\mu = -0.2$, $\sigma = 0.08$) measured in micrometers, and the distribution of S_r is Normal($\mu = -11$, $\sigma = 4.13$) measured in MPa.

Given a set geometric design variables $\mathbf{d}$ and the specified random variables $\mathbf{x} = [\Delta W, S_r]$, the response function is the minimum actuation force $F_{min}(\mathbf{d}, \mathbf{x})$ and failure is defined to be an actuation force with magnitude less than 5.0. The problem formulation is then:

$$p_f = P\left[F_{min}\left(\mathbf{d}, \mathbf{x}\right) + 5.0 > 0\right] \tag{II.27}$$

Figure II.17 displays the results of a parameter study of the response function $g(\mathbf{x}) = F_{min}(\mathbf{d}, \mathbf{x}) + 5.0$ as a function of the uncertain variables $\mathbf{x}$ for the optimal design reported in Ref. 24. The contour plot is scaled to a ± 3 standard deviation range in the transformed u-space. The limit state $g(\mathbf{x}) = 0$ (equivalent to $F_{min}(\mathbf{d}, \mathbf{x}) = -5.0$) is indicated by the solid line. For some design variable sets $\mathbf{d}$ (not depicted), the limit state is relatively well-behaved in the range of interest and first-order probability integrations would be sufficiently accurate. For the design variable set used to generate Figure II.17, the limit state has significant nonlinearity, and thus demands more sophisticated probability integrations. The most probable point converged to by the MPP search methods is denoted in Figure II.17 by the circle.

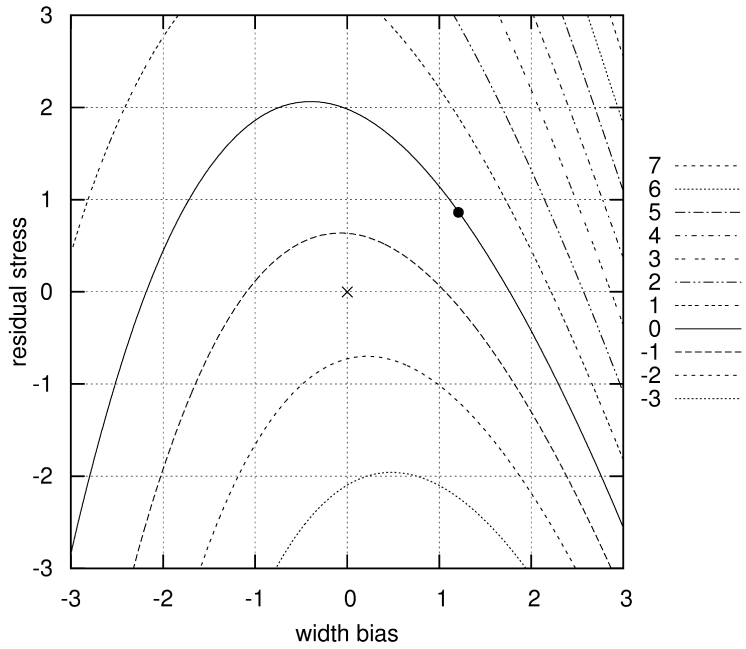


Figure II.17: Contour plot of $F_{min}(\mathbf{d}, \mathbf{x})$ as a function of uncertain variables $\mathbf{x}$. Solid line: limit state $g(\mathbf{x}) = 0.0$; $\times$: mean; circle: MPP.

Table II.7: Results for the bistable mems device problem.

Reliability Method	Function Evaluations	First-Order p_f	Sampling p_f (Avg. Error)
No Approximation	396	0.06815	—
x-space AMV^2+	50	0.06815	—
u-space AMV^2+	50	0.06815	—
x-space TANA	60	0.06816	—
u-space TANA	81	0.06815	—
LHS solution	1k	—	0.1130 (3.953%)
EGRA (MAIS)	15.3*	—	0.1099 (0.153%)

*The average number of evaluations is reported for EGRA.

Table II.7 gives a summary of the results from the same methods investigated in the previous examples. Second-order integration results are not provided because the quasi-Hessians that are computed during the MPP search, while helpful for informing the MPP search algorithm, do not provide adequate curvature information to generate accurate probability estimates. For this problem, the probabilities calculated by second-order integration were found to be less than those calculated by first-order integration. Inspection of Fig. II.17 shows that this is clearly not the case. Moreover, even when numerical Hessians are used, second-order integration is not possible for this problem because the large, negative principal curvatures create numerical difficulties. For the MPP search methods, the first-order probabilities are the most accurate available. Finite element analysis of this MEMS device is too expensive for the exhaustive LHS studies used in the previous examples, so 20 simulations with only 1,000 samples each were performed. Because the true solution to this problem is not known, the errors in the various solutions cannot be provided. However, for the stochastic methods, the average absolute error from among the 20 runs is reported. This error is calculated by comparing the individual solutions of a method to the mean solution of that method, so it is similar to the coefficient of variation of the solutions.

Because the true solution is unknown, no definitive argument can be made on the accuracy of the EGRA solution, but given the similarity to the LHS solution and the

fact that the MPP-search methods rely on low-order approximations to the shape of a limit state that is known to be multimodal (see Fig. II.17), it is reasonable to say that EGRA provides a more accurate estimate to the probability of failure than the MPP-search methods. A remarkable feature of these results is that EGRA requires less than one-third the number of function evaluations than even the cheapest of the MPP-search methods and the average error in the EGRA solutions is considerably smaller than for the LHS solutions. Given the larger variance in the LHS solutions, the accuracy of EGRA relative to the LHS results for the preceding examples, and the fact that the average EGRA solution here lies easily within the LHS error bounds, there is the implication that the average EGRA solution may actually be more accurate than the average LHS solution. Moreover, that EGRA solution comes at a cost of only about 15 true function evaluations versus the 1,000 required for LHS.

II.5 Summary

This chapter developed a new Efficient Global Reliability Analysis method and tested it through a wide array of reliability analysis problems. The results of these tests show that EGRA is generally less expensive than even the most efficient MPP-based methods while maintaining an accuracy that rivals even exhaustive sampling. This new method will allow engineers to estimate the reliability of a design more quickly, more accurately, and with greater confidence than any previously available method.

The test problems considered linear, nonlinear, and highly multimodal response functions with up to nine input random variables with varying distribution types and correlations. EGRA performed well on all of the tests, consistently proving more efficient than some or all of the MPP-based methods and producing accuracy similar to using one million LHS samples.

EGRA is not sensitive to the choice of input distribution or the correlation structure. This is because the largest computational effort in the EGRA algorithm is in

training the GP model, which is done independently of the input distributions. The surrogate model is constructed in the original x-space, so the distributions do not play a role until the sampling phase, which is easily handled. Because the GP model is built independently of the input distributions, EGRA can be expected to be equally accurate for any distribution or correlation. This also means that once the model is built, it will remain accurate for any *change* in the input distributions - a feature of EGRA that will be discussed in detail in Chapter IV.

The cost of EGRA is driven by three factors: the size of the search space, the shape of the response function, and the probability of failure. Larger spaces simply require more samples to fill the space and reduce the uncertainty. Additionally, because the number of initial samples is defined by the number of random variables, larger problems will require more samples just to start the process. For very large problems, this initial cost could be prohibitive (a problem with 100 random variables would require 5151 function evaluations for the initial samples alone). In such cases, it may be more efficient to use an MPP-based method (though the efficiency of these methods will suffer from the extra dimensionality if finite differences are required to calculate derivatives), but concerns regarding the accuracy of these methods will remain. At some point, as the number of variable increases, and the cost of EGRA rises, depending on the probability of failure and the level of accuracy required, it will become less expensive to use a sampling method since the cost of these methods does not scale with the size of the problem. However, for the size and reliability of most problems that engineers are typically concerned with, the efficiency and accuracy of EGRA remains compelling.

Simpler shapes can be captured by a GP with fewer points, thus reducing the cost of EGRA. This is apparent when comparing the stress response function in the cantilever beam test to the multimodal function. The linear stress function required only 10 samples (and could possibly be modeled with fewer, but this is the number of initial samples for this three-variable problem), while the multimodal function required

35 even though it had one fewer random variable.

The last factor driving the cost, the probability of failure, provides a significant advantage over other methods. EGRA is actually *less* expensive for lower probabilities of failure. If the probability of failure is low, then the size of the failure region relative to the search space is small. Smaller regions require shorter contours to contain them. This contour is the limit state for which EGRA is searching. If it is short, then it will require fewer samples to capture its shape, making EGRA less expensive. Because engineers are typically interested in problems with low probabilities of failure, EGRA provides a significant benefit when applied to real-world problems.

The "cost" of the method has been defined in this discussion as the number of required evaluations of the true response function when training the surrogate model. However, it should be noted that the computational expense of iteratively forming the GP model and then using an exhaustive global optimizer like DIRECT to find the maximum expected feasibility (thus requiring a large number of evaluations of the GP model) is non-negligible. For problems with quick-running response functions, this overhead cost can far outweigh the expense of a basic Monte Carlo analysis. EGRA is intended for application to problems with expensive response functions (such as the bistable MEMS problem explored in Section II.4.6) where this overhead cost pales in comparison to the expense of even a single analysis of the response function.

Another benefit to EGRA is the confidence that can be placed in its results. These test problems have shown that the MPP-based methods are inconsistent in terms of both accuracy and efficiency. For some cases, the different methods converge to different MPPs, and a few times the "No Approximation" method outperformed the methods that were designed to improve the efficiency of the MPP search. This inconsistency makes it difficult, if not impossible, for engineers to trust the results that these methods produce. On the other hand, for all test problems, the average error in the EGRA results was shown to be comparable or superior to using 100,000 LHS samples.

However, as the steel column test showed, EGRA is not without its problems. In

fact, if the underlying GP model fails, EGRA will fail, so any response function that cannot be well represented by a GP cannot be solved by EGRA. For the steel column problem, this could be easily dealt with by simply removing the "bad" part of the response function from the search space. This is an effective solution, but requires intervention by the user and some knowledge on if and where the response function might cause problems. Future versions of EGRA might automatically detect this behavior and adapt the search space as needed.

The development of EGRA is the major contribution of this dissertation. The remaining chapters discuss how EGRA can be used in the solution of traditionally difficult and/or expensive reliability problems beyond the basic reliability analysis problems presented here. Chapter III applies EGRA to system-level reliability analysis, Chapter IV explores unique ways EGRA can be used when performing reliability analysis using input distributions that are uncertain, and Chapter V applies EGRA to reliability-based design optimization.

CHAPTER III

SYSTEM-LEVEL RELIABILITY ANALYSIS

The previous chapters have been concerned with estimating the reliability given a single failure mode. However, in reality, even seemingly simple engineered systems can fail in multiple ways. For instance, consider the cantilever beam problem investigated in the previous chapters. In this problem, two failure modes were clearly identified - a displacement mode and a stress mode. But the problem was solved as essentially two separate problems rather than considering the total probability of failure of the system given that it could fail in either way. A system-level formulation of the problem would seek to find the probability that either the displacement limit is violated *or* the yield stress is exceeded.

The "or" condition is mathematically expressed through the union of the two individual failure events and defines their effect on overall system failure as a series system. A system in which all of the individual failure events must occur for the system to fail is a parallel system; the probability of failure of the parallel system is the probability of the intersection of all the failure modes.

Estimating the reliability with respect to a single failure mode is a challenging problem, necessitating the creation of EGRA to meet these challenges. Solving a system-level reliability analysis is, as should be expected, even more difficult. This chapter discusses how EGRA can be applied to this problem in a novel way to provide accurate assessments of the system reliability at a cost that rivals the reliability analysis for an individual failure mode (i.e., the reliability of a single component in the system).

III.1 Previous Methods

As with component-level analysis, methods for solving system-level reliability analysis can be generally broken into two groups: MPP-based methods, and sampling methods. And, again, the MPP-based methods are more efficient but rely on approximations that may be inaccurate while the sampling methods are generally more accurate, but only if enough samples are used, which can make them prohibitively expensive.

The probability of failure of a series system is defined as

$$p_f^{series} = P[\bigcup_i g_i(\mathbf{x}) \geq \bar{z}_i] \tag{III.1}$$

and for a parallel system,

$$p_f^{parallel} = P[\bigcap_i g_i(\mathbf{x}) \geq \bar{z}_i] \tag{III.2}$$

where each individual failure is defined as the response g_i exceeding the limit $\bar{z}_i$, but could also be defined as not exceeding $\bar{z}$, and could be different for each component i.

Solving either Eqs. III.1 or III.2 via Monte Carlo sampling is conceptually simple. For each random realization of $\mathbf{x}$, all of the component response functions g_i are evaluated. For a series system, if *any* component fails, then the random sample is counted as a system failure. Accordingly, for a parallel system, if *all* of the components fail, the sample is counted as a system failure. The system probability of failure is then simply the ratio of the number of system failures found to the total number of samples drawn, i.e. $p_f^{system} = N_f^{system}/N$.

Solving the system-level problem via MPP-based methods is more complicated. These methods are generally broken into two steps: First, the MPP for each component problem is located, which defines the reliability index β for that component. Second, various methods have been proposed to combine this component information to approximate the system reliability. Ref. 57 provides a good summary of these

methods.

In general, any of the methods previously discussed can be used to locate the MPP for each of the components. With these located, let $\mathbf{B}$ represent the vector of reliability indices for each of the components, and let $\mathbf{A}$ represent the correlation matrix between the component failure modes (calculated as the dot products of $\frac{-\nabla_{\mathbf{u}} G_i^T}{\|\nabla_{\mathbf{u}} G_i\|}$). Then, approximating the limit states as linear, for a series system, the probability of failure can be calculated as $1 - \Phi(\mathbf{B}, \mathbf{A})$; for a parallel system it is $\Phi(-\mathbf{B}, \mathbf{A})$. If only two failure modes are present, these can be evaluated using the bivariate normal distribution,[23] but the more general multivariate case is more difficult. Options include methods to bound the system probability of failure,[21] or various methods to approximately evaluate the distribution using importance sampling,[5] multiple linearizations,[42] or moment-based approximations.[61] Because all of these methods require that 1) the MPP be successfully located for each component, and 2) the component reliability can be accurately quantified with the applied approximation, these methods may not lead to accurate reliability estimates.

In short, no method that is both efficient and accurate currently exists for general system-level reliability analysis. The next section will discuss three possible ways that EGRA might be applied to this problem, followed by the application of the most promising of these three to a collection of example problems.

III.2 Formulations Using EGRA

There are multiple ways in which EGRA might be applied to the system-level reliability analysis problem. This section will explore three of them, detailing the advantages and disadvantages of each.

III.2.1 Component Solutions

The first way that EGRA can be used to estimate the reliability of a system is conceptually very simple. The basic idea is to first use EGRA to independently train a

GP model for each of the component response functions. Note that this step only constructs the models, it does not sample them to calculate the component reliability (though that could optionally be performed if the engineer desired this information). Once a GP model has been built for each component, these models can be sampled in the same way that sampling might be used to solve the system reliability problem directly on the "true" response functions (see Section III.1). Because EGRA has already been shown to be an efficient and accurate method for constructing surrogate models of the component limit states, this would similarly provide an efficient and accurate method for solving the system-level problem. However, consider the case where one (or more) component does not contribute to the system probability of failure because the probability of the system experiencing its failure mode is much smaller than that of other components. In such a case, the effort spent resolving the GP model for this component's limit state is essentially wasted. While this method of applying EGRA to the system-level problem would be accurate, its efficiency can be further improved. The next two methods present ways that EGRA can focus its efforts only on the limit states that bound the failure region of the system.

III.2.2 Composite Gaussian Process Model

In this second method, rather than construct an independent GP model for each of the components, EGRA attempts to train a single GP model to capture the so-called "composite" limit state. This limit state is made up of the portions of the component limit states that bound the system failure region. Consider the system problem depicted in Figure III.1 consisting of the following component response functions:

$$g_1(\mathbf{x}) = x_1^2 + x_2 - 8 \tag{III.3}$$

$$g_2(\mathbf{x}) = \frac{x_1}{5} + x_2 - 6 \tag{III.4}$$

The lines in this plot represent the component limit states (where $g_1 = 0$ and $g_2 = 0$); the shaded area is the system failure region. The composite limit state clearly has "sides" defined by the g_1 limit state and a "top" defined by the g_2 limit state.

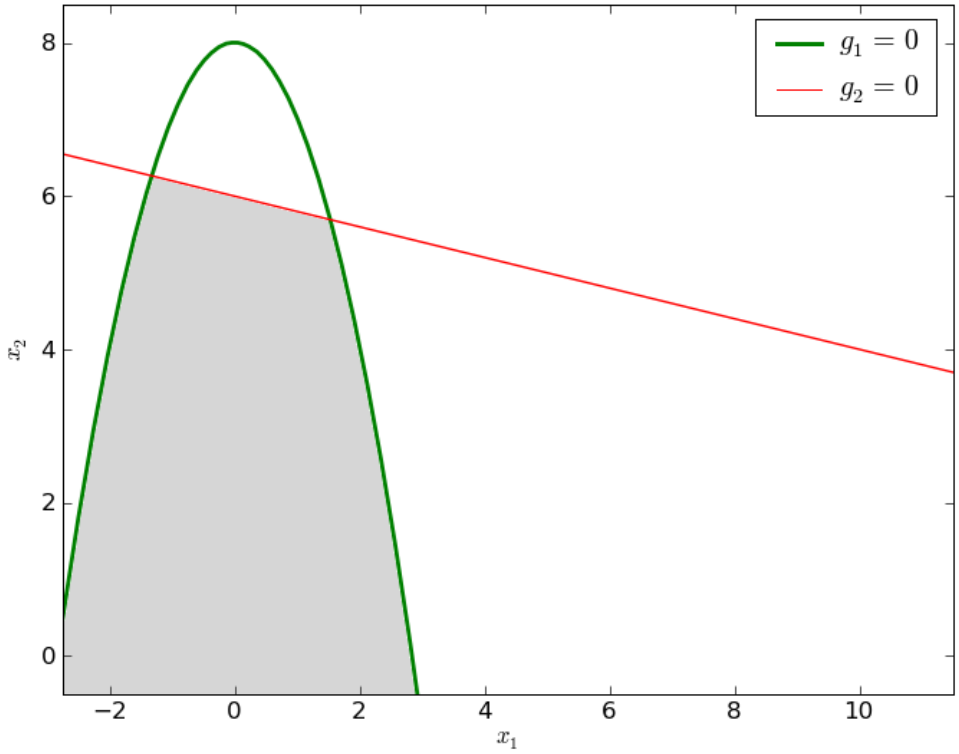


Figure III.1: Graphical depiction of the composite limit state. The lines are component limit states; the shaded area is the system failure region.

EGRA can be encouraged to search for this composite limit state through a simple modification of the response function. Let the system reliability problem be defined by:

$$p_f = P[g_1(\mathbf{x}) \geq 0 \cap\ g_2(\mathbf{x}) \geq 0] \tag{III.5}$$

Because both of these component limit states are defined via a $\geq$ operator and their limit state values are both $\bar{z} = 0$ (in general, these conditions can always be met through a simple rearrangement of the component response functions) this problem can be redefined as:

$$p_f = P[\min(g_1(\mathbf{x}), g_2(\mathbf{x})) \geq 0] \tag{III.6}$$

where this "min" operation is now the composite response function on which EGRA

will operate. Note that if this system was concerned with the union of these events rather than the intersection (a series rather than a parallel system), then the "min" could simply be replaced with a "max". Additionally, this concept can be scaled up for any number of response functions.

With this composite response function in place, EGRA now operates as in the case of an individual failure mode. For each point at which EGRA requests the response, all of the component response functions are evaluated and either the minimum or maximum (depending on the definition of the system) response is returned. Figure III.2 shows the results of this method applied to the system shown in Figure III.1.

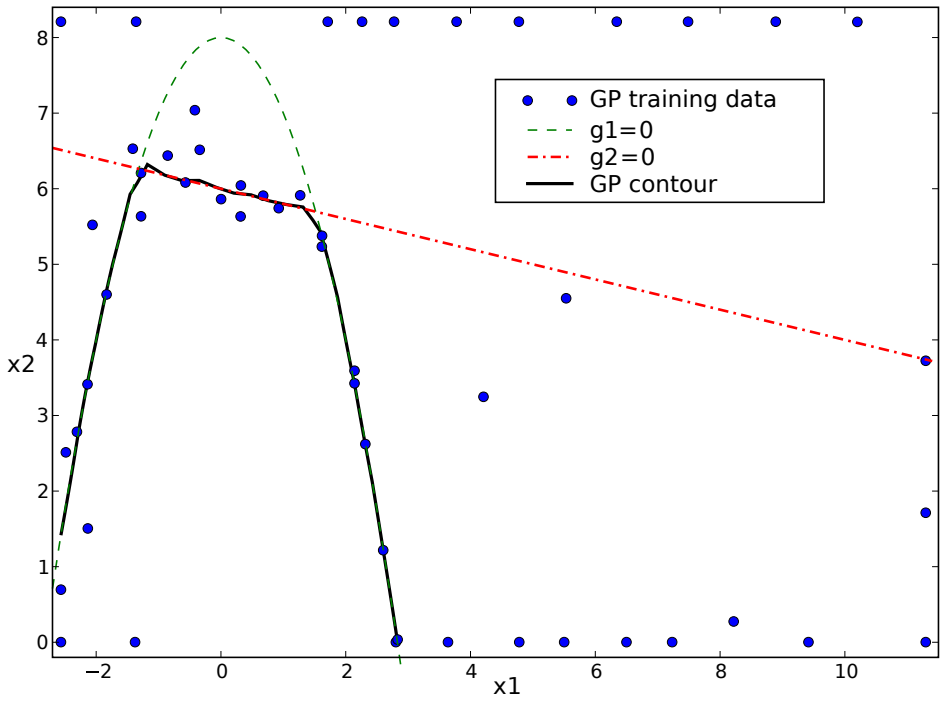


Figure III.2: Results from running EGRA on the composite response function. Note the error in the limit state contour near the corners.

Conceptually, this method should be more efficient than building GP models separately for each response function because EGRA might be able to "ignore" portions of the component limit states that do not contribute to the composite limit state. However, it is clear from the plot that this is not the case. While EGRA does do a good job of focusing the training data near the composite limit state, it requires a relatively substantial number of samples to do so (note the large number of wasted samples from

from the limit state) and the final form of the limit state contour has poor accuracy near the corners where the component limit states meet. There are two issues with solving the system problem in this way that create fundamental problems with the composite GP model.

The first issue is the relative scaling between the responses. At each sample point, EGRA receives the response value from only one of the components. Each of the components has the same limit state value, but they may diverge from this value as they move away from the limit state at much different rates. This creates discontinuities in the composite GP because points that are relatively near one another might have largely different responses because they actually come from different components. Because the GP relies on determining the correlation between points based on their distance from one another and the magnitude of their responses, it is clear how this kind of response creates difficulty in fitting a GP to the data. This leads to greater variance in the GP predictions, which leads to greater Expected Feasibility Function values, which leads EGRA to request more training data to converge the model, making this method far more expensive than hoped.

The second problem with fitting a composite GP model is the presence of the sharp corners in the composite limit state. GP models create smooth responses, so they are incapable of accurately modeling sharp changes in the limit state contour. For any case where the composite limit state is made up of portions of the component limit states, there will be corners where the separate component limit states cross. While the "sharpness" of these corners may differ, they will always be nonsmooth and thus create difficulties for the composite GP model.

An additional problem with this method is the computational expense. As was just discussed, the difficulty with fitting a GP model to the composite response requires EGRA to request a relatively large number of training points, so it is important to realize that at each of these training points *all* of the component response functions must be evaluated. So while the result in Figure III.2 shows that 52 training points

were requested, it actually represents 104 function evaluations.

The final method for applying EGRA to system-level reliability analysis retains the concept of the composite limit state, but keeps the GP models of the components independent.

III.2.3 Composite Expected Feasibility Function

The first system-level EGRA method provided an accurate reliability estimate, but might be overly expensive if one or more components does not contribute greatly to failure of the system. The second method introduced the concept of the composite limit state in an effort to reduce this expense, but attempting to form a composite GP model created problems that actually led to additional expense. The third method investigated here utilizes the composite limit state, but moves its effect on EGRA internally to the Expected Feasibility Function, rather than replacing the response function itself. By keeping the component GP models independent, this removes the difficulty in training the composite model while still retaining the cost benefit of seeking only the composite limit state.

First, recall Eq. II.14, which shows that the Expected Feasibility Function (EFF) is of the form:

$$\begin{aligned} EF(\hat{g}(\mathbf{x})) = (\mu_g - \bar{z}) & \left[2\,\Phi\left(\frac{\bar{z}-\mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right) \right] \\ & - \sigma_g \left[2\,\phi\left(\frac{\bar{z}-\mu_g}{\sigma_g}\right) - \phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) - \phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right) \right] \\ & + \epsilon \left[\Phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) \right] \end{aligned} \tag{III.7}$$

In Eq. II.14, this function was derived with only a single component GP model in mind, $\hat{g}$. Forming the *composite* EFF involves simply selecting the component GP model from which the mean and standard deviation will be used. This is done using the same min/max logic as was used to form the composite response function in

Eq. III.6. The resulting EFF might be rewritten as:

$$\begin{aligned} EF(\mathbf{x}) = \left(\mu_{\mathbf{g}}^* - \bar{z}\right) & \left[2\,\Phi\left(\frac{\bar{z} - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right) - \Phi\left(\frac{z^- - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right) - \Phi\left(\frac{z^+ - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right)\right] \\ & - \sigma_{\mathbf{g}}^* \left[2\,\phi\left(\frac{\bar{z} - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right) - \phi\left(\frac{z^- - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right) - \phi\left(\frac{z^+ - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right)\right] \\ & + \epsilon \left[\Phi\left(\frac{z^+ - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right) - \Phi\left(\frac{z^- - \mu_{\mathbf{g}}^*}{\sigma_{\mathbf{g}}^*}\right)\right] \end{aligned} \tag{III.8}$$

where $\mu_{\mathbf{g}}^*$ is the appropriately chosen min/max predicted response from among all the component GP model predictions at this particular point $\mathbf{x}$, and $\sigma_{\mathbf{g}}^*$ is the standard deviation from the corresponding model. Using this max/min relationship will create discontinuities in the EFF throughout the search space, but because the maximizing point of this function is sought using DIRECT[34] (see Section II.3), which is a gradient-free global optimizer, this does not create any problems for the method.

The point $\mathbf{x}$ that maximizes this function is the point at which there is the greatest expectation that it lies on the *composite* limit state. Of course, there may not be a large expectation that it lies on more than one *component* limit state. Once this point has been found, the EFF value for each of the components is calculated at this point. Only the components with a non-converged EFF value (i.e., EFF $>$ 1E-5) are evaluated and added to that component's GP model training data. By restricting the evaluation of the "true" response functions to only those components that have a large expectation of contributing to the composite limit state, the wasted evaluations that were found in the first method are eliminated.

Convergence of this method is measured based on the composite EFF value. When this value is less than the convergence tolerance, then the entire composite limit state has been located even though there may be substantial uncertainty remaining for portions of the various component limit states. The results of this method are explored in more detail in the discussion of the first computational experiment presented in the next section.

The third of these methods for using EGRA to solve system-level reliability analysis problems is clearly more promising than the others, so only this method is applied to the collection of test problems explored in this section. Where available, EGRA is compared to other methods from published literature, but in all cases it is compared to varying levels of LHS sampling to demonstrate its accuracy.

III.3.1 Multimodal System

This problem is an adaptation of the component multimodal example presented in Section II.4.1. Two additional nonlinear response functions have been added to form a system problem. The component response functions are described by:

$$g_1(\mathbf{x}) = \frac{\left(x_1^2 + 4\right)(x_2 - 1)}{20} - \sin\frac{5x_1}{2} - 2 \tag{III.9}$$

$$g_2(\mathbf{x}) = (x_1 + 2)^4 - x_2 + 4 \tag{III.10}$$

$$g_3(\mathbf{x}) = (x_1 - 4)^3 - x_2 + 2 \tag{III.11}$$

The distribution of x_1 is Normal($\mu = 2$, $\sigma = 1$) and x_2 is Normal($\mu = 5$, $\sigma = 1$); the variables are uncorrelated. The response level of interest for all response functions is $\bar{z} = 0$ with failure defined by $g_i < \bar{z}$.

Two formulations of this problem are explored. First, it is considered as a parallel system, followed by its solution as a series system.

Parallel Multimodal System

This parallel system problem can be formulated as:

$$p_f = P\left[g_1(\mathbf{x}) < 0 \cap g_2(\mathbf{x}) < 0 \cap g_3(\mathbf{x}) < 0\right] \tag{III.12}$$

Figure III.3 shows a plot in x-space of the three limit state contours throughout the

± 5 standard deviation search space. The shaded area represents the system failure region. It is clear from this plot that composite limit state is made up of only the g_2 component limit state and a portion of the g_1 limit state; the g_3 limit state does not bound the system failure region.

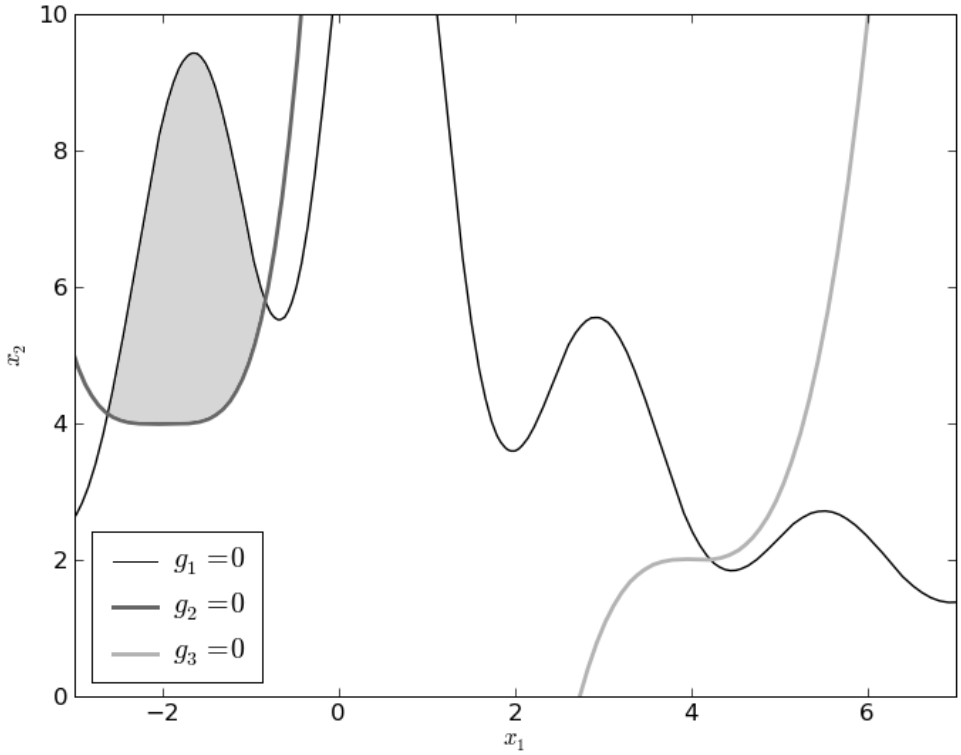


Figure III.3: Limit state contours of the multimodal system response functions. The shaded area is the system failure region for the parallel system.

Figure III.4 shows an example of the resulting GP models when EGRA is applied to this problem. The solid lines are the approximations to the limit state contours from the GP models; the dashed lines are the true limit states as shown in the previous plot. Consider, first, the limit state of g_2. This entire limit state is well-captured by the GP model because a significant portion of it bounds the system failure region. The limit state for g_1, however, is only accurately modeled in the region where it bounds the system failure region. Far from this region, it is highly inaccurate. This showcases EGRA's ability to not only seek accuracy of the model near the limit state, but to only the *portions* of that limit state where accuracy is needed.

The circles on this plot show the locations of the training data used by EGRA. These

are also color-coded and sized to represent the response function on which they were evaluated. There are six points at which all three response functions were evaluated. These are the initial points used to begin the EGRA process. Note that after these initial points, g_3 is never again evaluated. Because EGRA is searching only for the composite limit state, and this component limit state is far from the system failure region, EGRA needs no additional accuracy for this function and thus never requests another evaluation of it. There are several additional data points evaluated on g_1 and g_2, but it is clear that these are focused near the composite limit state.

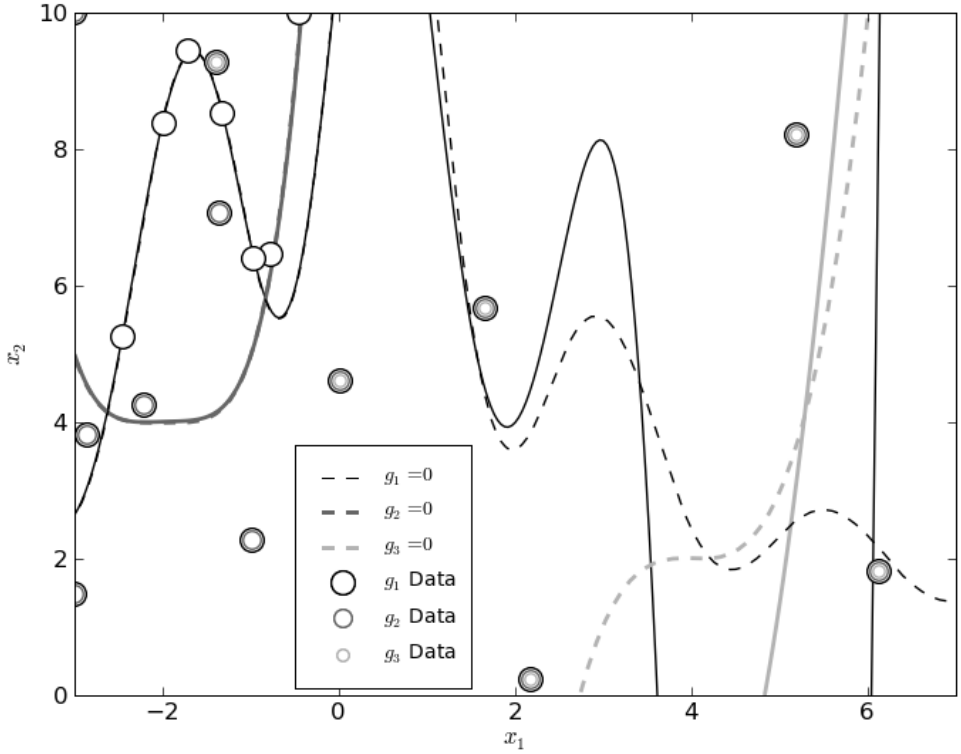


Figure III.4: Resulting GP models and training data from a run of EGRA on the parallel multimodal system. The solid lines represent the GP approximations to the limit state contours for the three response function.

Table III.1 gives a summary of the results for this test case. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Because EGRA is stochastic, it was also run 20 times and the average probabilities are reported. To measure the accuracy of

the method, two errors are reported for the EGRA results: the error in the average probability, and the average of the absolute errors from 20 independent simulations. For comparison, the same errors are given for 20 runs of LHS studies comprised of 10,000 and 100,000 samples.

Table III.1: Results for the parallel multimodal system problem.

Reliability Method	Fn Evals g_1	Fn Evals g_2	Fn Evals g_3	p_f (% Error, Avg. Error)
LHS solution	10k	10k	10k	0.001135 (0.203%, 13.38%)
LHS solution	100k	100k	100k	0.001156 (2.013%, 4.628%)
LHS solution	1M	1M	1M	0.001133 ("truth", 1.351%)
EGRA (1M LHS)	17.9*	9.1*	6.2*	0.001134 (0.141%, 1.666%)

*The average number of evaluations is reported for EGRA.

EGRA is able to provide an estimate of the system reliability that is as accurate as using three million LHS samples at an average cost of only 33.2 function evaluations - approximately 0.0011% of the cost. Recall that when using this same multimodal function in the component test in Section II.4.1, EGRA required an average of 34.9 function evaluations. Because EGRA is able to focus on only portions of the various component limit states, it is capable of solving this system problem with fewer total function evaluations than the component problem with the same response function.

Series Multimodal System

A series system problem can be formulated as:

$$p_s = 1 - P\left[g_1(\mathbf{x}) < 0 \cup g_2(\mathbf{x}) < 0 \cup g_3(\mathbf{x}) < 0\right] \tag{III.13}$$

where the probability of success $p_s = 1 - p_f$ is considered simply to provide a probability level that is easier to compare to other examples. This has no impact on the shape of the composite limit state or the behavior of EGRA. Figure III.5 shows the system failure region for this series system. It is bounded by the limit state of g_3 and a small portion of g_1, but the g_2 limit state does not bound the system failure region.

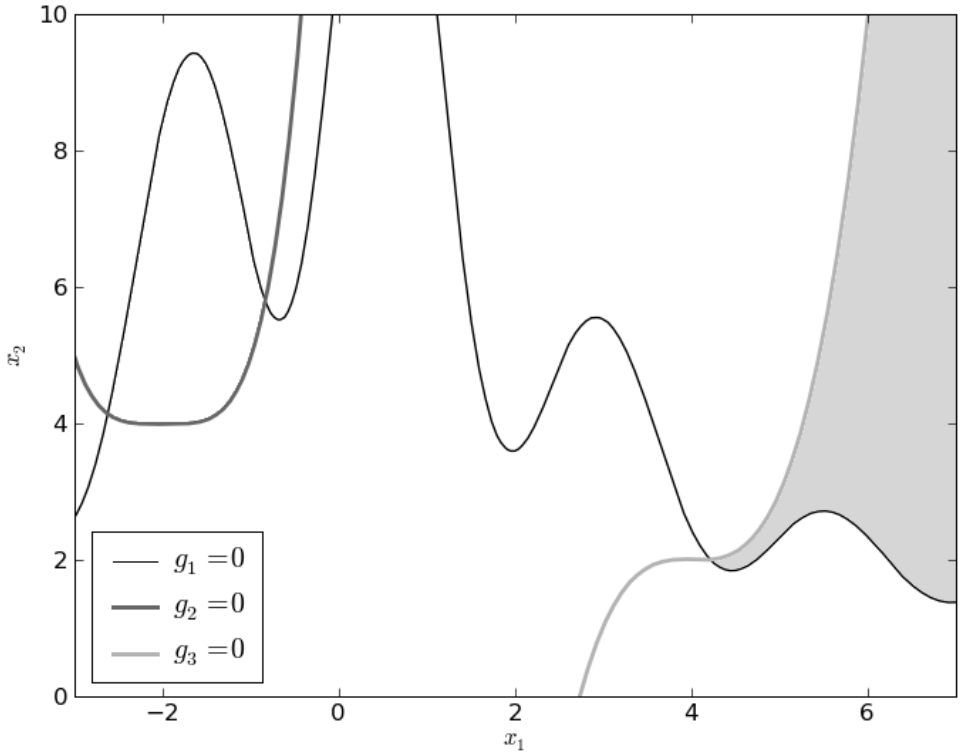


Figure III.5: Limit state contours of the multimodal system response functions. The shaded area is the system failure region for the series system.

Figure III.6 shows an example of the resulting GP models when EGRA is applied to this problem. Similar to the result for the parallel system, EGRA focuses the training data near only the portions of the component limit states that form the composite limit state (i.e., they bound the system failure region). The GP approximations to the limit state of g_2 and the portions of g_1 that are far from the failure region are significantly inaccurate, but EGRA does not attempt to improve them because accuracy of these models is not required to determine the reliability of the system.

Table III.2 gives a summary of the results for this test case. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Because EGRA is stochastic, it was also run 20 times and the average probabilities are reported. To measure the accuracy of the method, two errors are reported for the EGRA results: the error in the average probability, and the average of the absolute errors from 20 independent simulations.

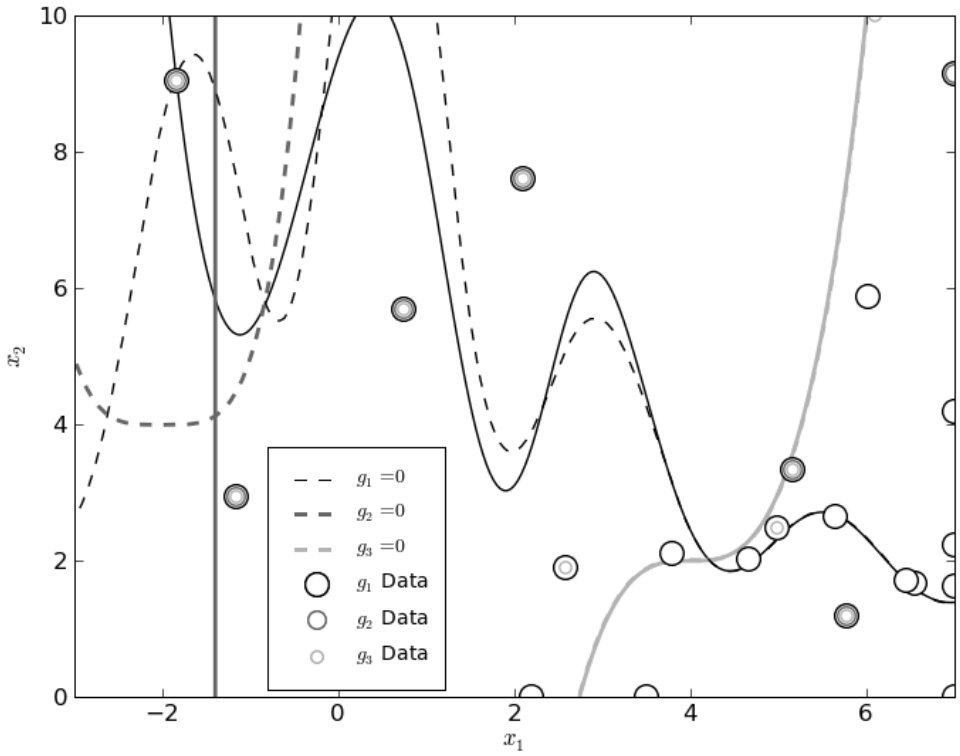


Figure III.6: Resulting GP models and training data from a run of EGRA on the series multimodal system. The solid lines represent the GP approximations to the limit state contours for the three response function.

Table III.2: Results for the series multimodal system problem.

Reliability Method	Fn Evals g_1	Fn Evals g_2	Fn Evals g_3	p_f (% Error, Avg. Error)
LHS solution	10k	10k	10k	4.000E-4 (3.964%, 26.78%)
LHS solution	100k	100k	100k	3.735E-4 (2.924%, 10.65%)
LHS solution	1M	1M	1M	3.848E-4 (“truth”, 2.229%)
EGRA (1M LHS)	19.8*	6.4*	9.2*	3.790E-4 (1.515%, 1.784%)

*The average number of evaluations is reported for EGRA.

EGRA is able to provide an estimate of the system reliability that is as accurate as using three million LHS samples at an average cost of only 35.4 function evaluations - approximately 0.0012% of the cost. The similarity in both the cost and accuracy of the EGRA solutions applied to the parallel and series formulations of this multimodal system demonstrate that EGRA is unconcerned with the definition of the system. The expense of EGRA is driven by the length and nonlinearity of the composite limit state, but whether it is operating on a series or parallel system is immaterial.

III.3.2 Cantilever Beam

This problem revisits the cantilever beam problem[76,88] investigated in the previous chapters, but combines the stress and displacement response functions to form a system reliability problem.

Random variables in this problem include the yield stress R of the beam material, the Young's modulus E of the material, and the horizontal and vertical loads, X and Y, which are modeled with normal distributions using N(40000, 2000), N(2.9E7, 1.45E6), N(500, 100), and N(1000, 100), respectively. Problem constants include L = 100 in. and D_0 = 2.2535 in. The design of the beam is assumed to be $w = 2.6041$ and $t = 3.6746$. The constraints on beam response have the following analytic form:

$$stress = \frac{600}{wt^2}Y + \frac{600}{w^2t}X \leq R \tag{III.14}$$

$$displacement = \frac{4L^3}{Ewt}\sqrt{(\frac{Y}{t^2})^2 + (\frac{X}{w^2})^2} \leq D_0 \tag{III.15}$$

or when scaled:

$$g_S = \frac{stress}{R} - 1 \leq 0 \tag{III.16}$$

$$g_D = \frac{displacement}{D_0} - 1 \leq 0 \tag{III.17}$$

The system problem is defined as:

$$p_f = P\left[g_S\left(\mathbf{x}\right) < 0 \cup g_D\left(\mathbf{x}\right) < 0\right] \tag{III.18}$$

Table III.3 gives a summary of the results for this test case. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Because EGRA is stochastic, it was also run 20 times and the average probabilities are reported. To measure the accuracy of the method, two errors are reported for the EGRA results: the error in the average

Table III.3: Results for the cantilever beam system problem.

Reliability Method	Fn Evals g_S	Fn Evals g_D	p_f (% Error, Avg. Error)
LHS solution	10k	10k	0.001180 (7.715%, 17.04%)
LHS solution	100k	100k	0.001263 (1.224%, 6.861%)
LHS solution	1M	1M	0.001279 ("truth", 1.964%)
EGRA (1M LHS)	10.0*	22.3*	0.001280 (0.070%, 1.680%)

*The average number of evaluations is reported for EGRA.

probability, and the average of the absolute errors from 20 independent simulations. For comparison, the same errors are given for 20 runs of LHS studies comprised of 10,000 and 100,000 samples.

III.3.3 Liquid Hydrogen Tank

The final test problem involves the reliability analysis of a liquid hydrogen fuel tank on a space launch vehicle.[57,75] The tank has a honeycomb sandwich design with top and bottom plates made of aluminum alloy AL2024 with the sandwich material made of Hexcell 1/8-in.-5052.0015. The tank is subjected to stresses caused by ullage pressure, head pressure, axial forces due to acceleration, and bending and shear stresses caused by the weight of the fuel.

The random variables for this problem include the thickness of the plate $t_{plate} \sim$ Normal(μ=0.07433, σ=0.005), the thickness of the honeycomb $t_h \sim$ Normal(μ=0.1, σ=0.01), and the loads on the tank $N_x \sim$ Normal(μ=13, σ=60), $N_y \sim$ Normal(μ=4751, σ=48), and $N_{xy} \sim$ Normal(μ=-684, σ=11).

Three modes of failure are considered for the tank: von Mises strength, isotropic strength, or honeycomb buckling. The honeycomb buckling response function is defined by a response surface generated from the structural sizing program HYPERSIZER,[16] which is given in Ref. 75 as:

$$g_{HB} = 0.847 + 0.96x_1 + 0.986x_2 - 0.216x_3 + 0.077x_1^2 + 0.11x_2^2 + 0.007x_3^2 + 0.378x_1x_2 - 0.106x_1x_3 - 0.11x_2x_3 \quad \text{(III.19)}$$

where x_1, x_2, and x_3 are defined as:

$$x_1 = 4\left(t_{plate} - 0.075\right) \tag{III.20}$$

$$x_2 = 20\left(t_h - 0.1\right) \tag{III.21}$$

$$x_3 = -6000\left(\frac{1}{N_{xy}} + 0.003\right) \tag{III.22}$$

The response functions for the von Mises and isotropic strengths are defined by:

$$g_{vM} = \frac{84,000\, t_{plate}}{\sqrt{N_x^2 + N_y^2 - N_x N_y + 3N_{xy}^2}} - 1 \tag{III.23}$$

$$g_{ISO} = \frac{84,000\, t_{plate}}{|N_y|} - 1 \tag{III.24}$$

This system problem is formulated as:

$$p_f = P\left[g_{vM}\left(\mathbf{x}\right) < 0 \cup g_{ISO}\left(\mathbf{x}\right) < 0 \cup g_{HB}\left(\mathbf{x}\right) < 0\right] \tag{III.25}$$

The particular value of the mean plate thickness used here is chosen because it is the optimal value reported in Ref. 57 from a reliability-based design optimization study performed on this liquid hydrogen tank system. To estimate the system reliability in that work, FORM is used to locate the MPP for each response, and then Pandey's method[61] is used to perform the multinormal integration.

Table III.4 gives a summary of the results for this test case. To establish an accurate estimate of the true solution, 20 independent simulations were performed using one million Latin hypercube samples per simulation. The average probability from these simulations is reported as the "true" solution. Because EGRA is stochastic, it was also run 20 times and the average probabilities are reported. To measure the accuracy of the method, two errors are reported for the EGRA results: the error in the average probability, and the average of the absolute errors from 20 independent simulations. For comparison, the same errors are given for 20 runs of LHS studies comprised of

Table III.4: Results for the liquid hydrogen tank system problem.

Reliability Method	Fn Evals g_{vM}	Fn Evals g_{ISO}	Fn Evals g_{HB}	p_f (% Error, Avg. Error)
LHS solution	10k	10k	10k	0.000700 (0.416%, 15.86%)
LHS solution	100k	100k	100k	0.000692 (0.803%, 4.866%)
LHS solution	1M	1M	1M	0.000697 ("truth", 1.334%)
FORM/Pandey[57]	—	—	—	0.001000 (43.45%, —)
EGRA (1M LHS)	18.6*	6.3*	10.0*	0.000702 (0.739%, 1.839%)

*The average number of evaluations is reported for EGRA.

10,000 and 100,000 samples.

The computational expense of a single analysis of the FORM/Pandey method used in Ref. 57 is not provided, but it should be noted that throughout the entire RBDO analysis, only 48 total function evaluations are used to estimate the reliability. This certainly indicates that this is an efficient method, particularly in a case where analytic gradients are available, but because it relies on linear approximations to the limit states, the method can be inaccurate. EGRA is capable of accurately estimating the system reliability for this problem at an average cost of only 34.9 total function evaluations.

III.4 Summary

This chapter presented the application of the Efficient Global Reliability Analysis method to system-level reliability analysis. Three formulations for applying EGRA to this class of problems were explored, but one was identified as the best option. This formulation uses independent Gaussian process models for each of the component response functions, but selects the training data for these models based on a search for the composite limit state. At each new training point selected by EGRA, only the response functions for which this point is expected to improve the approximation of its component limit state are evaluated, i.e. all component response functions are not evaluated at all points.

By focusing the GP training data (points at which the true response functions are

evaluated) near only the portions of the component limit states that bound the system failure region, locally accurate models can be built with very few samples. This makes EGRA a highly efficient way to perform system-level reliability analysis. The efficiency and accuracy of this method were demonstrated through its application to a collection of example problems. Both parallel and series systems were explored, and EGRA proved equally efficient and accurate for both formulations.

CHAPTER IV

RELIABILITY ANALYSIS WITH DISTRIBUTION UNCERTAINTY

Thus far, in all the example problems investigated throughout this dissertation, statistical descriptions of the inputs being treated as random variables have been provided and assumed known. However, when solving real-world problems, the probability density functions for these random variables are often estimated by fitting probability distribution models to observed test data. Because there can never be an infinite amount of test data, there is always some degree of uncertainty associated with the actual underlying probability density functions that generated the observed data. Consequently, there is also some amount of uncertainty associated with any probability of failure estimate that is computed with a reliability analysis based on the assumed probability distribution models.

Uncertainty associated with a probability distribution model can be broken into two parts: uncertainty about the probability distribution model form, and uncertainty about the model parameters. Several recent papers have illustrated approaches for quantifying the effect of distribution parameter uncertainty on reliability predictions.[56,74,81] However, little effort, especially within the reliability analysis community, has been made to quantify the effect of uncertainty about the *form* of the probability distribution model. Some recent work has addressed this objective through the use of so-called *generalized* probability distribution models, which have as special cases more well-known probability distribution models, such as the normal distribution.[56] However, this approach is still restricted by the assumption that the observed sample data came from a probability distribution model with a particular form, albeit a more general one.

This chapter demonstrates a rigorous approach founded in Bayesian inference for quantifying the uncertainty in both the distribution model form and its parameters.

Sometimes referred to as Bayesian Model Averaging, this type of approach allows one to compute the uncertainty associated with an output of interest (in this case, the reliability or probability of failure) by averaging over multiple possible models, based on their relative likelihoods, as indicated by observed data.

It is important to note that while this type of analysis is certainly *possible* with a different reliability analysis method, such as Monte Carlo sampling, it is the application of EGRA to this analysis that has made quantifying the uncertainty in reliability estimates due to the distribution uncertainty finally *practical* due to the vast computational savings.

IV.1 Bayesian Inference and Model Averaging

Loosely speaking, probability distribution model uncertainty refers to the possibility that the chosen mathematical representation of a random variable differs from the actual real-world data-generating process. The extent of the potential for difference between the postulated model and the reality (this difference is based on currently available information or lack thereof), as well as the potential impact on some quantity of interest being predicted, are of interest to the analyst employing probabilistic methods. In this section, a rigorous theory, based on Bayesian inference, is presented to quantify these effects.

For the remainder of this chapter, the term *model* (in the context of a probability distribution) is used to refer to a set of probability distributions. This is synonymous with the term *model form* as used in the introduction. For example, a Gaussian (or normal) model for a random variable X consists of all probability density functions for which X is normally distributed with some mean and standard deviation. Following Ref. 84, an individual model $\mathscr{M}$ can be written mathematically as

$$\mathscr{M} = \{p_\theta(x)\} \tag{IV.1}$$

where $\boldsymbol{\theta}$ is a vector of model parameters. Eq. IV.1 says that $\mathscr{M}$ consists of a set of

probability densities for a random variable X that depend on the parameters $\boldsymbol{\theta}$.

IV.1.1 Bayesian Inference

First, consider the case where the probability distribution model is known, but the parameters of that model must be estimated based on some observed data, $\mathbf{d} = (x_1, \ldots, x_n)^T$. As such, the model parameters are subject to uncertainty. Under the Bayesian framework, this uncertainty is quantified by calculating the posterior distribution of the parameters, which is based on Bayes' theorem:

$$p(\boldsymbol{\theta} \mid \mathbf{d}) = \frac{\pi(\boldsymbol{\theta}) p(\mathbf{d} \mid \boldsymbol{\theta})}{\int \pi(\boldsymbol{\theta}) p(\mathbf{d} \mid \boldsymbol{\theta}) \, d\boldsymbol{\theta}}, \tag{IV.2}$$

where $p(\boldsymbol{\theta}|\mathbf{d})$ is the posterior distribution for $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta})$ is the so-called prior distribution, and $p(\mathbf{d}|\boldsymbol{\theta})$ is the likelihood function for $\boldsymbol{\theta}$. The integral in the denominator is simply a normalizing constant, so that the posterior distribution is proportional to the prior distribution multiplied by the likelihood function. The likelihood function, $p(\mathbf{d}|\boldsymbol{\theta})$, is what relates the data to the unknowns, and it is typically written as $L(\boldsymbol{\theta})$ because the data hold fixed values once observed.

The prior distribution, $\pi(\boldsymbol{\theta})$ is expressed as a probability distribution function for $\boldsymbol{\theta}$, and simply represents the state of knowledge about the unknowns before observing $\mathbf{d}$. In most cases, though, one would like the inference to be guided solely by the observed data, so that vague (a.k.a reference) prior distributions are employed. Such vague prior distributions typically capture the notion that *a priori*, any value of $\boldsymbol{\theta}$ is equally likely.

Consider a simple example in which the mean and variance of a normal distribution are being estimated using n independent samples from that distribution: $\mathbf{d} = x_1, \ldots, x_n$. The unknowns are thus the mean and variance: $\boldsymbol{\theta} = (\mu, \sigma^2)$. The first step is to specify the prior distribution; the standard vague reference prior for this case is[53]

$$\pi(\mu, \sigma^2) \propto 1/\sigma^2 \tag{IV.3}$$

which is the product of two independent priors, $\pi(\mu) \propto 1$ and $\pi(\sigma^2) \propto 1/\sigma^2$. Because the underlying distribution is Gaussian, the likelihood function for $\boldsymbol{\theta}$ is simply

$$L(\boldsymbol{\theta}) = p(\mathbf{d}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x_i - \mu)^2}{2\sigma^2}\right], \tag{IV.4}$$

or equivalently

$$L(\boldsymbol{\theta}) = \left(\sigma^2\right)^{-n/2} \exp\left[-\frac{S + n(\bar{x} - \mu)^2}{2\sigma^2}\right], \tag{IV.5}$$

where $\bar{x}$ is the sample mean and $S = \sum(x_i - \bar{x})^2$.

Because the reference prior distribution that was chosen forms a "conjugate pair" with the likelihood function, the posterior distribution for $\boldsymbol{\theta} = (\mu, \sigma^2)$ can be constructed analytically (see Ref. 53 for the details). If only the marginal uncertainty in a subset of parameters is of interest, this is expressed mathematically as an integration over the parameters which are not of interest. For example, the marginal distribution for μ can be computed as

$$p(\mu|\mathbf{d}) = \int p(\mu, \sigma^2|\mathbf{d})\, d\sigma^2 \tag{IV.6}$$

After some simplification, it can be shown that

$$\frac{\mu - \bar{x}}{s/\sqrt{n}} \sim t_{n-1} \tag{IV.7}$$

where $\bar{x}$ and s are the sample mean and standard deviation of the observations (treated here as summary statistics of $\mathbf{d}$, not random variables), respectively, and t_{n-1} is a random variable with Student's t distribution on $n - 1$ degrees of freedom. This is the same distribution from which confidence intervals on the population mean are traditionally computed.

The difference between the Bayesian perspective and the classical perspective is that the latter treats the unknown population mean as a constant and the sample mean as a random variable, whereas Bayesian analysis treats the unknown popula-

tion parameters as random variables. By treating the unknowns as random variables, a rigorous, more complete representation about the uncertainty can be obtained. Further, Bayesian analysis allows for examination of distribution parameter uncertainty for *any* probability distribution which has a computable (at least to a proportionality constant) density function. The drawback is that the posterior distribution will not be analytically available in many cases, and numerical methods will be required. However, Markov Chain Monte Carlo sampling (MCMC) has been shown to be an effective method for simulating (through random sampling) the posterior distribution. A detailed treatment of MCMC can be found in Ref. 58.

IV.1.2 Bayesian Model Averaging

Now consider that instead of assuming that one particular probability distribution model is correct, k candidate models, $\mathscr{M}_j$, $j = 1, \ldots, k$, are under consideration, each with associated unknown parameters, $\boldsymbol{\theta}_j$. Here the goal is to quantify the relative support from the data for each candidate model. Bayes' theorem can be applied again, here to compute the posterior probability associated with each candidate model:

$$P(\mathscr{M}_j \mid \mathbf{d}) = \frac{p(\mathbf{d} \mid \mathscr{M}_j) P(\mathscr{M}_j)}{\sum_r p(\mathbf{d} \mid \mathscr{M}_r) P(\mathscr{M}_r)}, \tag{IV.8}$$

where $P(\mathscr{M}_j \mid \mathbf{d})$ is the posterior probability of $\mathscr{M}_j$, and $P(\mathscr{M}_j)$ is the prior probability of $\mathscr{M}_j$. In most cases, equal prior probabilities are desired for each candidate model, in which case the posterior probabilities become

$$P(\mathscr{M}_j \mid \mathbf{d}) = \frac{m_j}{\sum_r m_r}, \tag{IV.9}$$

where $m_j = p(\mathbf{d} \mid \mathscr{M}_j)$ is the likelihood of $\mathscr{M}_j$, which can be shown to be

$$m_j = \int \mathscr{L}_j(\boldsymbol{\theta}_j) \pi_j(\boldsymbol{\theta}_j) \, d\boldsymbol{\theta}_j, \tag{IV.10}$$

where $\mathscr{L}_j(\boldsymbol{\theta}_j) = p_{\theta_j}(\mathbf{d})$ is the likelihood function for $\boldsymbol{\theta}_j$ under $\mathscr{M}_j$. Thus, the posterior probability of each candidate model is based on the relative likelihood of observing the sample data under that model, where the unknown parameters are integrated out.

In practice, there are two difficulties associated with computing this posterior distribution: (1) it is necessary to specify a prior distribution, $\pi_j(\boldsymbol{\theta}_j)$ for each model, and (2) the integral of Eq. IV.10 can be difficult to evaluate. With regards to the prior distribution of the unknown parameters, the ideal approach in most cases would be to employ a noninformative reference prior, as is typically done when making inference on $\boldsymbol{\theta}$ alone. This poses a problem here, however, because most noninformative priors are improper and are thus only defined up to a constant, which results in an arbitrary constant in m_j. Note that when computing the posterior of $\boldsymbol{\theta}$, this is not a problem because the constant will get canceled; when computing the posterior of $\mathscr{M}_j$, however, the constant does not get canceled.

A popular approach for handling both the prior distribution problem and the integration problem is to compute an approximation to m_j that does not depend on the prior. Let $\ell(\boldsymbol{\theta}) = \log \mathscr{L}(\boldsymbol{\theta})$, and let $\hat{\boldsymbol{\theta}}$ denote the maximum likelihood estimator. A fairly accurate approximation to m_j is $\hat{m}_j$,[49,84] where

$$\hat{m}_j = \exp\left(\ell_j(\hat{\boldsymbol{\theta}}_j) - \frac{d_j}{2}\log n\right) \tag{IV.11}$$

and d_j is the number of components of $\boldsymbol{\theta}_j$.

Now posterior probabilities of each candidate model form $\mathscr{M}_j$ can be obtained, as well as posterior distributions for the parameters associated with each candidate model, $\boldsymbol{\theta}_j$. Typically, however, one is not interested in these posterior probabilities as such. Instead, one is interested in the uncertainty that they imply about some dependent quantity of interest.

Say that the quantity of interest is γ, which is a function of the probability distribution model, $\mathscr{M}$, and its parameters. The quantity of interest, γ, could be, say, a probability of failure obtained from a reliability analysis. For any particular probabil-

ity distribution model, $\mathscr{M}_j$, one could compute what could loosely be referred to as a posterior distribution for γ, $p(\gamma \mid \mathbf{d}, \mathscr{M}_j)$, by propagation of the posterior $p(\boldsymbol{\theta} \mid \mathbf{d}, \mathscr{M}_j)$ through the reliability analysis process.

Adding in model form uncertainty, it is easy to show that the posterior distribution of γ is given by

$$p(\gamma \mid \mathbf{d}) = \sum_r p(\gamma \mid \mathbf{d}, \mathscr{M}_r) P(\mathscr{M}_r \mid \mathbf{d}), \tag{IV.12}$$

where $p(\gamma \mid \mathbf{d}, \mathscr{M}_r)$ is the posterior distribution of γ under $\mathscr{M}_r$, which is simply a function of $p(\boldsymbol{\theta}_r \mid \mathbf{d})$. The posterior distribution of the quantity of interest, γ, is then a comprehensive representation of the uncertainty associated with γ due to uncertainty in both the probability distribution model form and parameters associated with the input random variable. This posterior distribution could be used, among other things, to display a histogram, compute an expected value, or compute confidence intervals. For example, the posterior expected value of γ is given by

$$\mathrm{E}[\gamma \mid \mathbf{d}] = \int \gamma \, p(\gamma \mid \mathbf{d}) \, d\gamma \tag{IV.13}$$

IV.2 Reliability Analysis Under Uncertainty

The goal of reliability analysis is to determine the probability that an engineered device, component, system, etc. will fail in service given that its performance is a function of some random inputs. This performance is defined by a response function $g(\mathbf{x})$, where $\mathbf{x}$ represents the vector of random variables defined by known probability distributions. Failure is then said to occur when the response function exceeds (or fails to exceed) some threshold value $\bar{z}$. The probability of failure, p_f, is then defined by

$$p_f = \int_{g > \bar{z}} p(\mathbf{x}) \, d\mathbf{x} \tag{IV.14}$$

where $p(\mathbf{x})$ is the joint probability density function of the random variables $\mathbf{x}$, and the integration is performed over the failure region where $g > \bar{z}$.

Clearly, the probability of failure depends on the specified probability density function for the inputs, $p(\mathbf{x})$. In practice, though, $p(\mathbf{x})$ may need to be estimated using observed sample data, in which case it is subject to uncertainty, as outlined in Section IV.1. Since the probability density $p(\mathbf{x})$ is uncertain, the computed probability of failure is also uncertain, hence the term reliability analysis *under uncertainty*.

The approach employed here to account for this uncertainty is to first quantify the uncertainty associated with $p(\mathbf{x})$ using the methods outlined in Section IV.1, and then to propagate this uncertainty through to p_f. Conceptually, one can think of randomly generating "realizations" of $p(\mathbf{x})$ from its uncertainty model; for each realization, the corresponding failure probability, p_f, is computed based on Eq. IV.14.

Section IV.1 describes the process for using Bayesian inference to construct what is essentially a hierarchical uncertainty model for $p(\mathbf{x})$. Uncertainty associated with model form is quantified by posterior probabilities $P(\mathscr{M}_j \mid \mathbf{d})$, while uncertainty associated with model parameters, conditional on a particular model form, is quantified by the posterior distribution $p(\boldsymbol{\theta}_j \mid \mathbf{d}, \mathscr{M}_j)$. It turns out that if the common practice of calculating the posterior $p(\boldsymbol{\theta}_j \mid \mathbf{d}, \mathscr{M}_j)$ using Markov Chain Monte Carlo sampling is adopted, then it is a simple matter to generate "realizations" of $p(\mathbf{x})$. This is because the MCMC sampling process itself provides random samples of $\boldsymbol{\theta}_j$ from $p(\boldsymbol{\theta}_j \mid \mathbf{d}, \mathscr{M}_j)$.

To put the above concepts together, the uncertainty in p_f due to uncertainty in the specification of $p(\mathbf{x})$ can be quantified by constructing an uncertainty distribution using the following sampling scheme:

1. Generate a random realization of the probability density function $p(\mathbf{x})$:

   (a) Randomly select a probability distribution model $\mathscr{M}_j$ based on the model form posterior probabilities, $P(\mathscr{M}_r \mid \mathbf{d})$, $r = 1, \ldots, k$.

   (b) Using the MCMC sampler, generate a random realization of the parameter vector $\boldsymbol{\theta}_j$ from $p(\boldsymbol{\theta}_j \mid \mathbf{d}, \mathscr{M}_j)$.

2. Compute the corresponding value of p_f, which represents one sample from its

uncertainty distribution.

The results from this sampling scheme could be used to derive a variety of metrics that are useful in conveying the impact of probability distribution model uncertainty on the results of the reliability analysis. For example, one could compute a confidence interval or conservative upper bound for p_f at a given confidence level.

There is a fundamental challenge, however, with using the above scheme for practical problems. This is that in practice, the performance function $g(\mathbf{x})$ is typically not a closed-form function, but instead it may only be observable by exercising a (possibly expensive) computer simulation, such as a finite element model. As such, direct computation of the failure probability given by Eq. IV.14 is rarely possible, and a variety of computational methods have been developed specifically for this purpose. Widely used methods include those based on finding the so-called *Most Probable Point* (MPP) and sampling methods.[39]

Because the sampling scheme outlined above will require multiple reliability analyses, an efficient reliability analysis technique is clearly of paramount importance. This efficient technique is provided in EGRA. Chapters II and III have demonstrated several features of EGRA that make it an extremely attractive choice in this context. First, EGRA is very efficient and has been shown to require a number of function evaluations on the same order as MPP-based methods such as Advanced Mean Value. Second, EGRA has been shown to be very accurate, far-surpassing the accuracy of MPP-based methods on several nonlinear, multimodal test problems, even rivaling the accuracy of exhaustive LHS sampling. The most compelling argument for the use of EGRA in this situation, though, is that no new function evaluations will be needed each time the distributions of the input random variables change. As such, although the construction of the uncertainty distribution for p_f may require thousands of reliability analyses, EGRA will only require evaluations of $g(\mathbf{x})$ during the first reliability analysis.

The EGRA method achieves its efficiency and accuracy by constructing a Gaussian

process surrogate model to $g(\mathbf{x})$. What makes EGRA different from other surrogate-based methods is that the training points used to construct the surrogate are chosen deliberately to achieve high accuracy near the limit state, and only near the limit state (the limit state is the contour of $g(\mathbf{x})$ for which $g = \bar{z}$). Once the surrogate model has been constructed, it is then sampled exhaustively to compute p_f; because the surrogate model does not depend on the specific probability distribution models of the inputs, it does not need to be reconstructed when the distribution models and/or parameters change. This feature makes the method very well-suited for comprehensive uncertainty analysis with Bayesian inference.

IV.3 Computational Experiments

This section presents three example problems. The first revisits the cantilever beam problem, but now assuming that the input distribution parameters are estimated from varying amounts of test data. Note, however, that the distribution *forms* are assumed known in this problem. The second demonstrates the convergence of the Bayesian Model Averaging to the "correct" distribution as additional test data is gathered. The final problem revisits the Bistable MEMS problem and performs the reliability analysis assuming that the distribution model form and its parameters are uncertain for one of the inputs.

IV.3.1 Cantilever Beam with Parameter Uncertainty

The cantilever beam problem (previously investigated in Sections II.4.3 and III.3.2, as well as Refs. 24, 26, 63, 76, 81, 88 and others to test reliability analysis and design methods) is used as an example. This problem has two response functions of interest; one concerning the stress, g_S and one concerning the displacement, g_D:

$$g_S = R - \frac{600}{wt}\left(\frac{Y}{t} + \frac{X}{w}\right) \tag{IV.15}$$

$$g_D = D - \frac{4L^3}{Ewt}\sqrt{\frac{Y^2}{t^4} + \frac{X^2}{w^4}} \tag{IV.16}$$

The limit states for these response functions are at $g_S = 0$ and $g_D = 0$; all variables are described in Table IV.1.

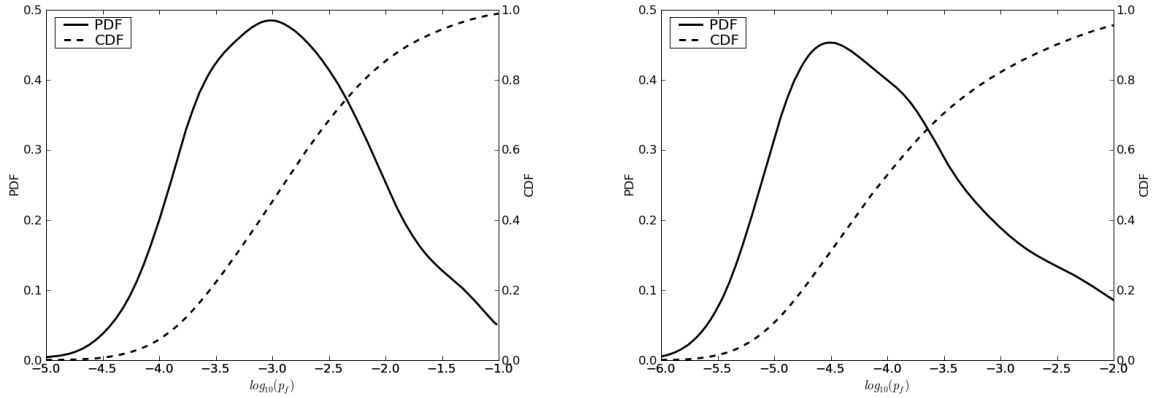
Table IV.1: Variable detail for the cantilever beam example. The values for t, w, and D are taken from Ref. 81.

Variable	Mean	COV	Distribution
Horizonal Load, X	500	0.2	Normal
Vertical Load, Y	1,000	0.1	Normal
Yield Strength, R	40,000	0.05	Normal
Modulus of Elasticity, E	29e6	0.05	Normal
Length, L	100	—	Deterministic
Beam Width, w	2.6041	—	Deterministic
Beam Thickness, t	3.6746	—	Deterministic
Max Displacement, D	2.25	—	Deterministic

The probabilities of failure at this design for the stress and displacement functions are: $P(g_S \leq 0) = 1.111 \times 10^{-3}$ and $P(g_D \leq 0) = 2.093 \times 10^{-4}$ (determined from the average probability of failure reported from 20 independent runs of the EGRA algorithm). Of course, these numbers are assuming that the distribution parameters are known. If the variables describing the material properties E and R are known to follow normal distributions, but their means and standard deviations must be estimated from a limited number of material tests, Bayesian inference can be used to quantify the uncertainty in these probabilities of failure.

For this problem, it was assumed that only 5 observations of the material properties E and R are available. These "observations" are drawn from random sampling of the *assumed* unknown distributions listed in Table IV.1. Using these data, 1000 samples of the unknown distribution parameters are drawn using Markov Chain Monte Carlo sampling. This is used as the "outer loop" of a nested sampling method similar to that used in Ref. 81. On the "inner" loop, EGRA is used to calculate the probability of failure given the particular set of distribution parameters. For each iteration, EGRA searches for the limit state over $\mu_E^i \pm 5\sigma_E^i$ and $\mu_R^i \pm 5\sigma_R^i$, which will clearly differ for each iteration but will largely overlap. Subsequent iterations inherit the GP model created in previous iterations, but may require training that model in regions

of the random variable space not previously visited and thus may require additional function evaluations. For this example, because only 5 observations have been made, there is considerable uncertainty in the distribution parameters, making the search space rather large. In total, for the 1000 MCMC samples, EGRA required 11 function evaluations for the stress response function g_S and 74 for the displacement response function g_D. The end result is the posterior distribution of the probability of failure for each response function, which are shown in Figure IV.1.



(a) Stress function.

(b) Displacement function.

Figure IV.1: Posterior distributions of the probability of failure for the stress and displacement response functions, respectively, assuming only 5 observations for the input distributions of the material properties E and R.

From these distributions, several statistics could be calculated, e.g. the expected probability of failure, upper and lower confidence bounds, or the confidence that the probability of failure will lie below some value of interest such as a regulatory limit. It is important to point out that these curves are as accurate as an exhaustive nested sampling method often used for this type of problem, but at a cost of only 11 and 74 evaluations of the true response functions instead of the possibly billions that might be required by the nested sampling. This is clearly a substantial cost savings.

Moreover, assume now that further observations of the material properties are made. These additional data provide additional certainty of the input variable distributions, i.e. the distributions of their parameters have less variance. Because the

GP model was built independent of the input distributions and their parameters it does not have to be rebuilt to accomodate this change in the input distributions. The same MCMC-MAIS nested sampling can be performed to generate an updated probability of failure distribution at the cost of zero additional true function evaluations. Figure IV.2 shows the posterior distributions of the probability of failure assuming 5, 10, 25, 50, 100, and 1000 samples of the input distributions for the stress response function. Figure IV.3 shows the same for the displacement response function.

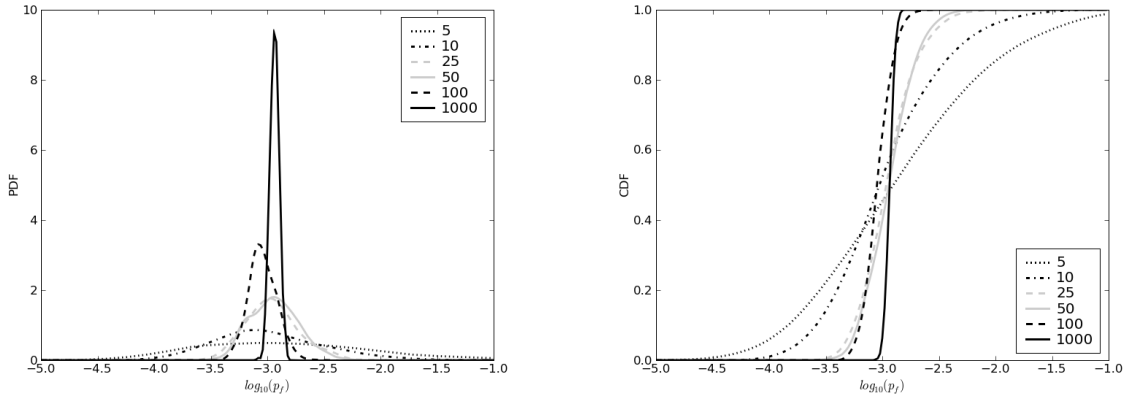


Figure IV.2: Posterior probability density and cumulative distribution functions for the stress response function assuming varying numbers of initial samples.

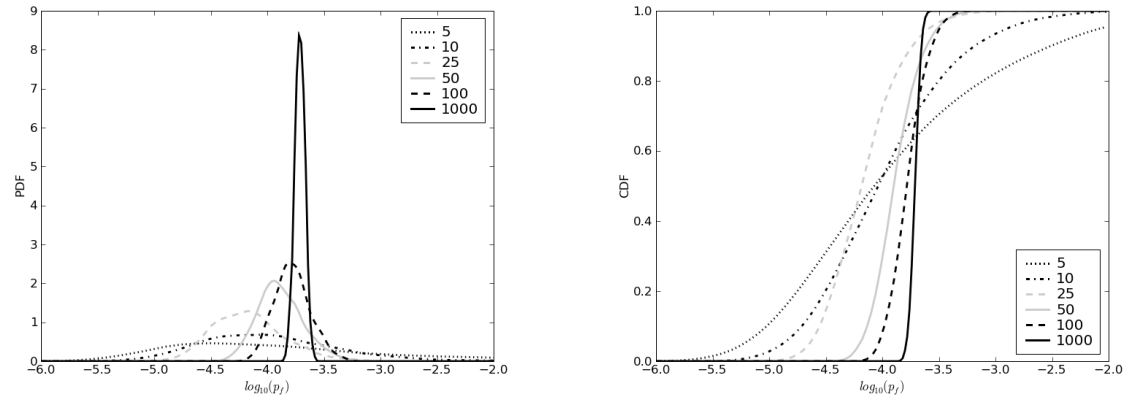
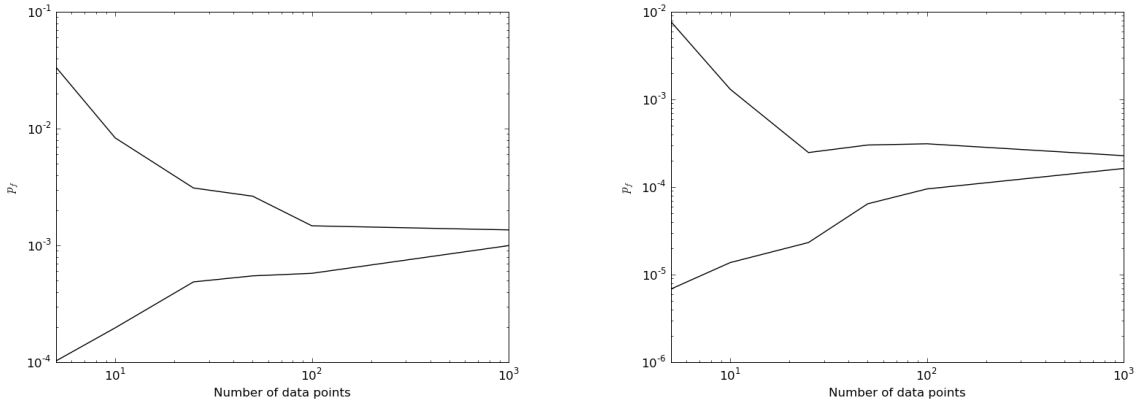


Figure IV.3: Posterior probability density and cumulative distribution functions for the displacement response function assuming varying numbers of initial samples.

These plots show a clear convergence of the posterior distributions toward the true probability of failure. This can be easier seen by plotting the convergence of the

confidence bounds. Figure IV.4 shows the 5% and 95% confidence bounds on the probability of failure distribution for both response functions.



(a) Stress function. (b) Displacement function.

Figure IV.4: Convergence of the 5% and 95% confidence bounds on the probability of failure distributions for the stress and displacement response functions, respectively.

IV.3.2 Convergence of Model Form with Additional Data

The model form (posterior) probability is a representation of the degree of belief that a random variable may be represented using a particular probability distribution model form, based on a set of observed data. As such, one would intuitively expect that as more and more data are collected, the model form probability should converge to either zero or one, representing an absence of uncertainty about the "correct" model form. When the amount of data is small, the observations may not be sufficient to discriminate between candidate model forms, in which case one would expect the model form probabilities to be more evenly distributed among the hypothesized models. In this section, examples are presented to explore the extent to which these properties manifest themselves during the practical application of Eqs. IV.9 and IV.11.

Candidate Models: Normal and Lognormal

First, consider the simple case in which two candidate model forms are available with which to represent a random variable: the normal model and the lognormal

model. In practice, these two are often used interchangeably, and each may be justifiable on the basis of the central limit theorem. The probability density functions for the normal and the lognormal model are, respectively:

$$p_{\theta=(\mu,\sigma)}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{IV.17}$$

$$p_{\theta=(\lambda,\zeta)}(x) = \frac{1}{\zeta x\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\log x-\lambda}{\zeta}\right)^2\right],\ x>0 \tag{IV.18}$$

For a given set of observed data, the approximation of Eq. IV.11 is used to compute the model form probabilities. This involves evaluating the log of the likelihood function at the maximum likelihood estimates of the parameters. Under the assumption that the observed data, $\mathbf{d} = (x_1, \ldots, x_n)$, are independent, the log likelihood function is given by

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_\theta(x_i). \tag{IV.19}$$

For the normal model, the parameters $\hat{\boldsymbol{\theta}} = (\hat{\mu}, \hat{\sigma})$ that maximize the likelihood function (and also the log likelihood function) can be shown to be

$$\hat{\mu} = \bar{x} \tag{IV.20}$$

$$\hat{\sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{IV.21}$$

where the sample mean $\bar{x} = \left(\sum_i x_i\right)/n$. The maximum likelihood estimates for the parameters of the lognormal distribution are computed similarly, but with x replaced by $\log x$.

To illustrate the effect of sample size, data sets were randomly generated with sample sizes varying from 10 to 400 from a normal distribution with $\mu = 500$ and $\sigma = 100$. For each data set, the model form posteriors for the normal and lognormal models were computed using Eqs. IV.9 and IV.11. Because the observed data are random, this experiment was repeated 1,000 times at each sample size. Figure IV.5 plots the median

value of the posterior probability for the normal model as a function of sample size, along with 90% confidence intervals.

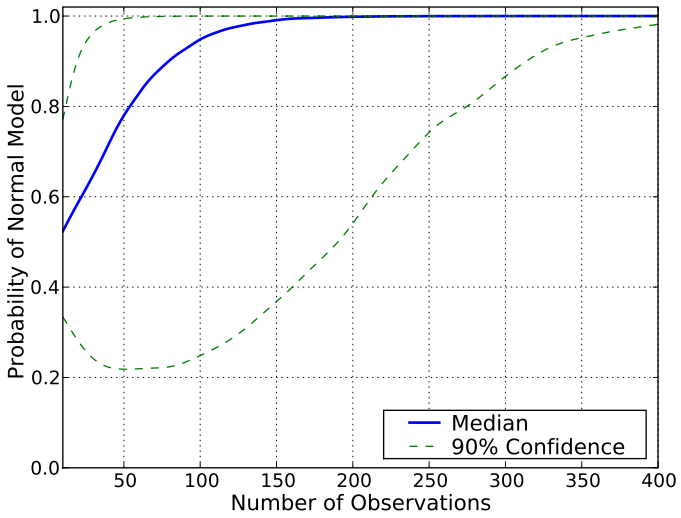


Figure IV.5: Posterior probability for the normal model (candidate models being the normal and lognormal) as a function of sample size. Sample observations are generated from the normal distribution with $\mu = 500$ and $\sigma = 100$.

Since the data are in fact generated from a normal distribution, the posterior probability of the normal model is expected to approach one (and conversely, the probability of the lognormal model to approach zero) as the sample size becomes large. Note that it takes a sample size of about 150 before the median probability reaches (approximately) one. However, with 150 observations, the 5th percentile of the probability (i.e. the lower bound based on a two-sided 90% confidence interval) is only 0.36, which is an indication that even at this sample size, the occasional spurious sample may be better approximated by the lognormal model than the normal. Further, sample sizes of 25 or less tend not to provide much evidence that the normal model is better than the lognormal model. For most of the cases having small sample sizes, the computation of the model form probability suggests (correctly) that there is substantial residual uncertainty about the correct model form.

Candidate Models: Normal, Lognormal, and Weibull

This example extends the previous one to the case in which there are three candidate model forms: normal, lognormal, and Weibull. The Weibull distribution is also commonly used in engineering applications, and for certain values of its parameters, it becomes a close approximation to the normal distribution. The probability density function for the Weibull distribution is defined for $x \geq 0$ as

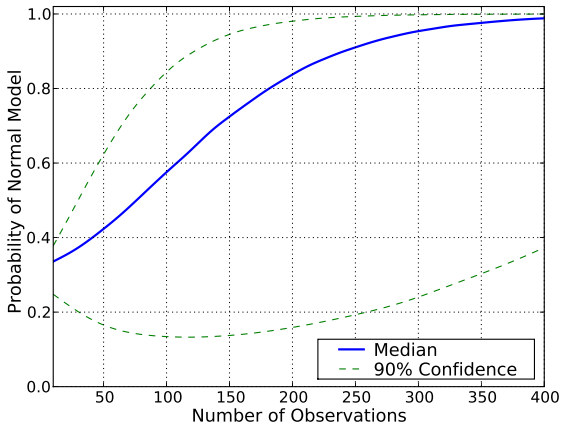
$$p_{\theta=(\alpha,\beta)}(x) = \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} e^{-(x/\alpha)^{\beta}} \tag{IV.22}$$

Unlike the normal and lognormal distributions, the maximum likelihood estimate of the Weibull parameters is not available in closed form. For the Weibull distribution, maximum likelihood estimation can be done using a gradient-based numerical optimization solver.

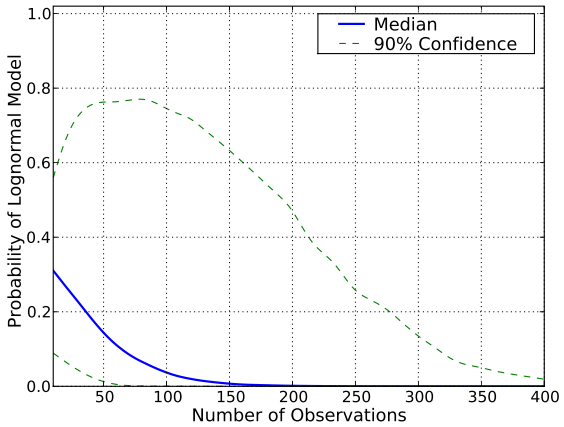
The previous analysis is repeated with the addition of the Weibull as a candidate probability distribution model form. The posterior probability for each model form is plotted in Figure IV.6 as a function of sample size, showing the median and 90% confidence bounds, as before. Based on the median, the probability of the normal model does show convergence towards one, but it is much slower than in the previous example. This is because the Weibull model can be a close approximation to the normal model, so a larger number of observations are needed before the two can be distinguished.

IV.3.3 Bistable MEMS

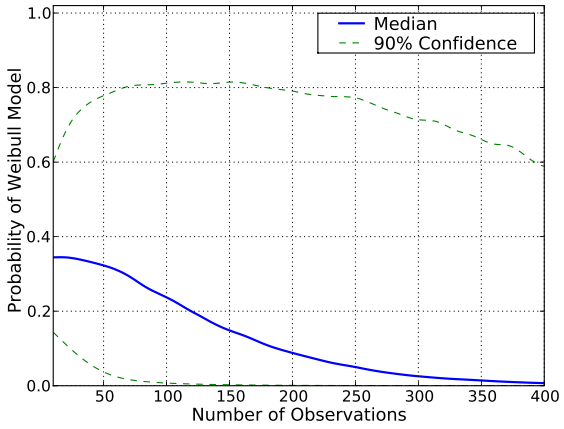
The preceding examples illustrate how Bayesian inference can be used to quantify distribution form uncertainty when modeling random variables, but the primary purpose of this work is to demonstrate how this type of inference can support reliability analysis for engineering systems. This example illustrates how distribution form uncertainty can be incorporated into the reliability analysis process.



(a) Normal model



(b) Lognormal model



(c) Weibull model

Figure IV.6: Posterior probabilities for the three candidate models as a function of sample size. Sample observations are generated from the normal distribution with $\mu = 500$ and $\sigma = 100$.

For this example, the reliability analysis of the bistable MEMS device presented in Section II.4.6 is repeated, where two key random variables are considered: ΔW (edge bias on beam widths) and S_r (residual stress in the manufactured device). The probability distributions used in the original analysis were $\Delta W \sim N(\mu = 0.2, \sigma = 0.08)$ and $S_r \sim N(\mu = 11, \sigma = 4.13)$. Note that for this analysis, the signs of the random variables have been reversed. This is done to provide more flexibility in exploring distribution model form uncertainty, since negative values preclude the use of several common models. The probability of failure is defined to be the probability that the magnitude of the minimum actuation force, $F_{\min}(\Delta W, S_r)$ is less than 5.0:

$$p_f = P\left[F_{\min}(\Delta W, S_r) < 5.0\right] \tag{IV.23}$$

The calculation of $F_{\min}(\Delta W, S_r)$ is achieved using a nonlinear, solution-verified finite element analysis solver.[2]

In order to illustrate the process of accounting for distribution model uncertainty, it is assumed that as opposed to being known, the probability distribution of the random variable ΔW must be estimated from a finite set of observed data. It is assumed that the distribution for ΔW follows one of three candidate models: normal, lognormal, or Weibull. Bayesian inference will be used, as outlined in Section IV.1, to account for both distribution model form uncertainty and distribution parameter uncertainty.

A hypothetical set of observed data of size $n = 20$ for ΔW is randomly simulated from its nominal distribution. For the purpose of the analysis, though, the observed data set is treated as the only available information about the distribution of ΔW. Figure IV.7 shows a normalized histogram of the observed data, along with the best fit estimates (using maximum likelihood) for each of the three candidate distribution models.

The normal and Weibull models appear to be very similar, and both appear to fit the data well. The lognormal model appears to be plausible as well, but does not appear to fit the data as well as the other two. The posterior probabilities associated

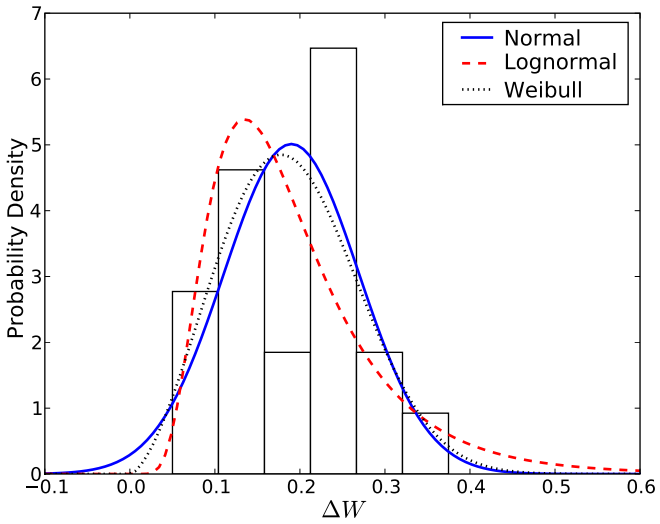


Figure IV.7: Histogram of observed data for random variable ΔW, along with best fit probability distribution models.

with each distribution model, computed using Eqs. IV.9 and IV.11 are tabulated in Table IV.2. The reason that the Weibull model shows a higher posterior probability than the normal model is probably attributable to its fit to the data in the lower tail. The lognormal model achieves a modest posterior probability, which is expected from the visual comparison.

The reliability analysis itself is conducted using EGRA, and the accuracy of this method has already been demonstrated for this particular example problem in Section II.4.6. As reported in that section, the probability of failure with the nominal random variable definitions is 0.1099, requiring an average of only 15.3 function evaluations to compute using EGRA. As previously mentioned in Section IV.2, one of the main advantages of using the EGRA method in this context (in addition to its excellent accuracy and efficiency) is that it allows the exploration of probability distribution uncertainty *without any additional function evaluations*. This makes reliability analysis with sparse data practical even for very expensive response functions.

An initial run with EGRA is performed to collect the "training data" with which to construct the targeted (for accuracy at the limit state) Gaussian process surrogate model of F_{min}. Once the training data are collected, this surrogate model can be sam-

Table IV.2: Model form posterior probabilities based on 20 observations of ΔW.

Model	$P(\mathscr{M} \mid \mathbf{d})$
Normal	0.34
Weibull	0.53
Lognormal	0.13

pled based on any feasible probability distribution model for the input random variables. Because this model is trained within ± 5 standard deviations of the mean for each random variable, it should be valid for all probability distribution models that the Bayesian inference suggests are feasible.

Once the surrogate model has been constructed, the uncertainty in the failure probability is quantified through repeated reliability analyses as outlined in Section IV.2. For each analysis, a probability distribution model form is randomly chosen for ΔW based on the posterior probabilities given in Table IV.2. The distribution parameters are then randomly chosen as well from their posterior distribution using Markov Chain Monte Carlo sampling. The resulting posterior distribution (based on 5,000 analyses) of p_f is illustrated in Figure IV.8, which shows a normalized histogram of the 5,000 samples of p_f, along with a kernel density estimate[72] based on those samples. The expected value is 0.146 (compare to the true failure probability of 0.1099), and the 95% confidence upper bound is 0.241.

The impact of distribution model form uncertainty can be further investigated by repeating the uncertainty analysis for each of the candidate model forms, considering only parameter uncertainty. The resulting individual posterior distributions are compared in Figure IV.9. Note that the posteriors for the normal and Weibull models are quite similar, whereas the posterior for the lognormal model suggests higher failure probabilities. This is consistent with Figure IV.7, where it can be seen that the lognormal model shows a significant difference from the normal and Weibull models. By accounting for the model form uncertainty using the Bayesian posterior probabilities, it is acknowledged that the underlying probability density function might be represented using different models, but at the same time the observed data is used to

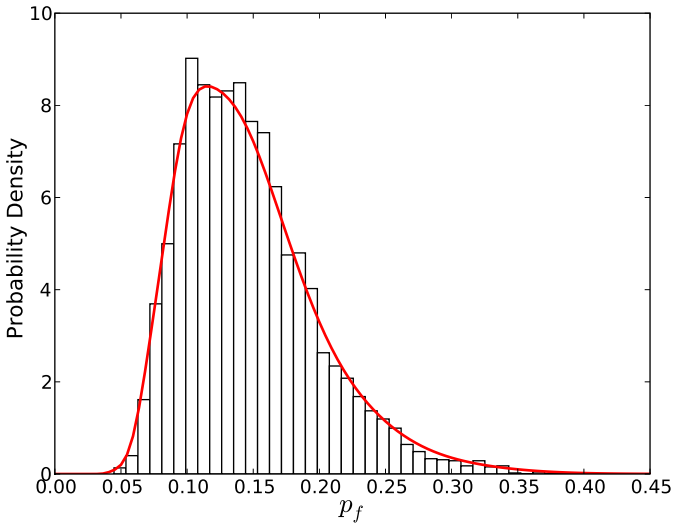


Figure IV.8: Histogram and kernel density estimate of posterior distribution of p_f considering uncertainty in both distribution model form and parameters for ΔW.

inform the degree of belief associated with each candidate model.

IV.4 Summary

Reliability analysis for engineering systems requires the specification of probability density functions for all of the random inputs. In practice, the importance of these input distributions is often overlooked, with more emphasis placed on the reliability method itself when reporting confidence in the results. Nevertheless, recent work has illustrated several rigorous approaches for quantifying the resulting uncertainty in reliability when input distribution parameters (such as the mean and standard deviation) must be estimated from observed sample data. This chapter extended these approaches, using Bayesian inference, to quantify the uncertainty associated with *both* probability distribution model form and model parameters.

An approach was presented whereby Bayesian inference is used to quantify the amount of evidence (in the form of posterior probabilities) that an observed sample gives to each of a set of candidate probability distribution models (e.g. the normal, lognormal, and Weibull models). For any given distribution model, the uncertainty associated with the distribution parameters is also quantified using Bayesian infer-

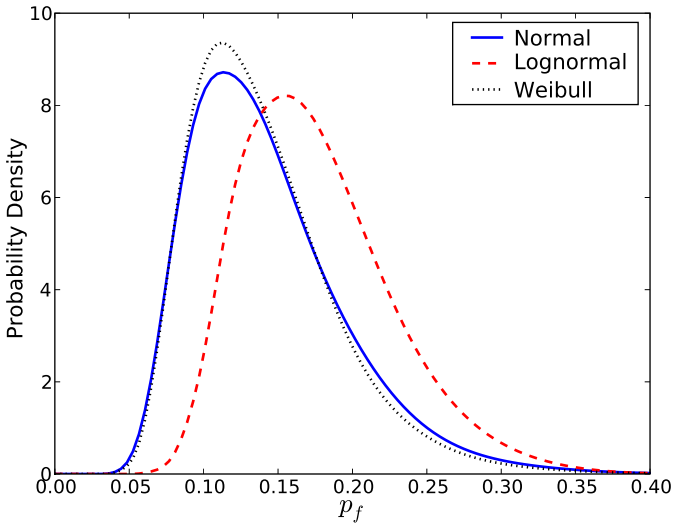


Figure IV.9: Posterior distributions of p_f, considering only distribution parameter uncertainty for ΔW.

ence.

This chapter also demonstrated how the resulting uncertainty in the input probability density functions can be used to characterize the uncertainty associated with the reliability estimate of interest. The approach is based on repeating the reliability analysis hundreds or thousands of times using feasible realizations of the input probability density functions suggested by the Bayesian inference. This approach can be theoretically expensive, but through the use of EGRA this uncertainty analysis can be performed without requiring many function evaluations beyond what would be needed for a single reliability analysis.

Several numerical examples are used to illustrate the approaches. First, the cantilever beam problem is used to demonstrate how the outlined methods can be used in conjunction with EGRA to investigate the impact of distribution parameter uncertainty on reliability estimates.

Next, the convergence of the model form posterior probabilities is illustrated using example problems with two and three candidate model forms. It is shown that in some cases, a very large number of observations may be needed to discriminate between similar models. In cases where both models fit the observed data well, the

difference in reliability computed using the two models may not be significant. For high-reliability applications, though, where tail behavior becomes important, distribution model form uncertainty may have a significant impact on reliability, even when the candidate models appear to fit observed data comparably well.

Finally, the quantification of uncertainty in the reliability analysis due to both distribution parameter and distribution form uncertainty is illustrated via the analysis of a bistable MEMS device. By assuming a hypothetical observed data set for one of the input random variables, the posterior distribution of failure probability is computed. This is done using 5,000 repeated reliability analyses with the EGRA method, but only 15 evaluations of the response function are needed. This example emphasizes that these seemingly expensive approaches can be brought to bear on even the most expensive reliability analysis applications.

CHAPTER V

RELIABILITY-BASED DESIGN OPTIMIZATION

Determining the optimal (lightest, least expensive, etc.) design for an engineered component or system that meets or exceeds a specified level of reliability is a problem of obvious interest across a wide spectrum of engineering fields. Various methods for this reliability-based design optimization (RBDO) problem have been proposed; several were discussed in Chapter I. Unfortunately, this problem is rarely solved in practice because, regardless of the method used, solving the problem is either too expensive or the final solution is too inaccurate to ensure that the reliability constraint is actually satisfied. This is especially true for engineering applications involving expensive, implicit, and possibly nonlinear performance functions (such as large finite element models). EGRA was developed in earlier chapters to improve both the accuracy and efficiency of reliability analysis for this type of performance function. This chapter explores how EGRA can be used in a design optimization context to create a method of sufficient accuracy and efficiency to enable the use of RBDO as a practical design tool.

RBDO is used to perform design optimization (such as minimizing the weight of a component) while accounting for reliability constraints. A general RBDO problem is typically of the form:

$$\begin{aligned} &\text{minimize} && f(\mathbf{d}) \\ &\text{subject to} && P\left[g(\mathbf{d},\mathbf{x}) \geq \bar{z}\right] \leq \bar{p}_f \end{aligned} \tag{V.1}$$

where the objective function f is a function of only the deterministic design variables $\mathbf{d}$, but the response function in the reliability constraint g is a function of $\mathbf{d}$ and $\mathbf{x}$, a vector of random variables defined by known probability distributions.

A common alternate formulation is the case where the design variables are actually parameters of the distributions of the random variables in $\mathbf{x}$. It is easy to imagine a situation where the designer could specify the nominal size of a component, but has no control over the manufacturing tolerance. In such a case, the specified size might be the mean value of a distribution with the variance set by the known variations in the manufacturing process. For this type of "variable insertion" problem, the RBDO problem could be stated as:

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{\theta}) \\ &\text{subject to} && P\left[g\left(\mathbf{x}(\boldsymbol{\theta})\right) \geq \bar{z}\right] \leq \bar{p}_f \end{aligned} \tag{V.2}$$

Another alternate formulation is to make the reliability the objective subject to some deterministic constraint. An example would be a situation where the designer desired to make a system as reliable as possible within some specified budget. For this type of formulation, the RBDO problem could be stated as:

$$\begin{aligned} &\text{minimize} && P\left[g\left(\mathbf{d}, \mathbf{x}\right) \geq \bar{z}\right] \\ &\text{subject to} && f\left(\mathbf{d}\right) \leq \bar{f} \end{aligned} \tag{V.3}$$

There are two essential ways that the computational expense of RBDO can be reduced. The first is to reduce the cost of performing the reliability analysis at a candidate design by reducing the number of evaluations of the response function g that are required. For this, EGRA can be used. The second is to reduce the number of design points at which the reliability analysis must be performed. To accomplish this, an efficient optimizer is needed. Moreover, a gradient-free optimizer is needed. Clearly, analytic derivatives of the probability of failure as a function of the design are not available. Equation I.30 provides an approximation based on the assumed shape of the limit state, but this derivative will be inaccurate if the approximation is poor. In general, this derivative would need to be calculated through numerical differentia-

tion, with a full reliability analysis required at each finite difference point; clearly this could require substantial cost.

The Efficient Global Optimization[47] (EGO) method discussed in Section II.1 is both efficient and gradient-free. However, EGO was originally created as an unconstrained minimizer, so without modification it is not applicable to the RBDO problem. The next section details a formulation to enable the solution of constrained problems using EGO.

V.1 Constraint Formulations for EGO

EGO was originally created as an unconstrained minimizer, and several methods have been investigated to add constraint support to the algorithm. Ref. 70 suggested multiplying the expected improvement at each point by that point's probability of being feasible (where that probability is determined from the variance in the GP model). Ref. 8 applied a penalty to the expected improvement value at points which are infeasible. Ref. 69 enforced the constraints by moving them into the $\max(EIF)$ problem.

The probability method[70] may be applicable to the RBDO problem, but was shown to leave slack in the constraints, causing it to converge to sub-optimal solutions.[69] The penalty method[8] and the constrained EIF method[69] cannot be used as implemented in those works because they assume that at any point, the true feasibility of that point can be easily determined. For RBDO problems, assessing the feasibility involves solving the reliability analysis problem, which will clearly be too computationally expensive. Instead, a new formulation for enforcing constraints in EGO is introduced here. This method constrains the $\max(EIF)$ problem using an augmented Lagrangian formulation, but uses the so-called "expected violation" since the true violation is unknown.

V.1.1 Augmented Lagrangian Formulation

Optimization methods based on merit functions are commonly used to solve constrained optimization problems. This type of method adds penalty terms to the objec-

tive function based on the constraint violations, thus making them less desirable to the optimizer. One very successful type of merit function is the augmented Lagrangian formulation. Using this formulation, the merit function for a constrained optimization problem can be generally written as:

$$\mathcal{M} = f + \lambda_g \Delta_g + r_p \Delta_g^2 + \lambda_h \Delta_h + r_p \Delta_h^2 \tag{V.4}$$

where λ_g and λ_h are Lagrange multipliers, r_p is the penalty coefficient, Δ_g is an inequality constraint violation ($\Delta_g = 0$ at feasible points), and Δ_h represents an equality constraint violation. Of course, $\mathcal{M}$, f, Δ_g, and Δ_h are all dependent on the point $\mathbf{x}$ at which they are evaluated. Note that this equation assumes the presence of only one of each type of constraint, but λ_g, λ_h, Δ_g, and Δ_h, could easily be replaced with vectors to form a more general formulation. For use with EGO, Gaussian process models stand in for f, Δ_g, and Δ_h:

$$\mathcal{M} = f + \lambda_g \hat{\Delta}_g + r_p \hat{\Delta}_g^2 + \lambda_h \hat{\Delta}_h + r_p \hat{\Delta}_h^2 \tag{V.5}$$

To drive the EGO algorithm, the expected improvement for this merit function is needed, but its calculation is not straightforward.

Recall that at any point, the terms $\hat{f}$, $\hat{\Delta}_g$, and $\hat{\Delta}_h$ do not represent single values, but Gaussian distributions. A linear combination of Gaussian distributions results in another Gaussian distribution, but the square term in Eq. V.5 means that the resulting distribution of $\mathcal{M}$ is not Gaussian. To accomodate the use of EGO in an augmented Lagrangian formulation, a deterministic value is needed in place of these distributions. The predicted violations μ_{Δ_g} and μ_{Δ_h} could be used, but they do not account for the uncertainty in these predictions. To provided a measure based on both the current prediction and the uncertainty in that prediction, an expected violation function can be used.

V.1.2 Expected Violation Function

The Expected Violation Function (EVF) for inequality constraints was originally introduced in Ref. 7, but was used differently in that work. Here it is used to replace the constraint violation terms Δ_g and Δ_h in Eq. V.5 with their *expected violation*. For an inequality constraint with an upper bound, the expected violation is calculated in a similar fashion as the previously derived expected improvement function:

$$EV_g = (\mu_g - \bar{g}) \left[1 - \Phi\left(\frac{\bar{g} - \mu_g}{\sigma_g}\right)\right] + \sigma_g \phi\left(\frac{\bar{g} - \mu_g}{\sigma_g}\right) \tag{V.6}$$

For an inequality constraint with a lower bound, this becomes:

$$EV_g = (\bar{g} - \mu_g) \Phi\left(\frac{\bar{g} - \mu_g}{\sigma_g}\right) + \sigma_g \phi\left(\frac{\bar{g} - \mu_g}{\sigma_g}\right) \tag{V.7}$$

And for an equality constraint:

$$EV_h = (\mu_h - \bar{h}) \left[1 - 2\Phi\left(\frac{\bar{h} - \mu_h}{\sigma_h}\right)\right] + 2\sigma_h \phi\left(\frac{\bar{h} - \mu_h}{\sigma_h}\right) \tag{V.8}$$

Using the proper expected violation for each constraint, the augmented Lagrangian merit function becomes:

$$\mathcal{M} = \hat{f} + \lambda_g EV_g + r_p EV_g^2 + \lambda_h EV_h + r_p EV_h^2 \tag{V.9}$$

Because this formulation adds deterministic values to the Gaussian distribution $\hat{f}$, the merit function is now truly Gaussian with parameters defined by:

$$\mu_{\mathcal{M}} = \mu_f + \lambda_g EV_g + r_p EV_g^2 + \lambda_h EV_h + r_p EV_h^2 \tag{V.10}$$

$$\sigma_{\mathcal{M}} = \sigma_f \tag{V.11}$$

These values can then be used in Eq. II.11 to calculate the expected improvement on the merit function.

When EGO converges, the GP models of the objective function and any constraints are theoretically accurate in the vicinity of the optimum solution due to the sample density in this region. Recognizing this, a gradient-based optimizer can be used to refine the EGO solution using these GPs rather than the true functions so that no additional true function evaluations are necessary. In this study, the Nonlinear Interior Point Solver (NIPS) in OPT++[59] is used to perform this solution refinement. Additional details on the implementation of the merit function can be found in Refs. 25 and 80. The scheme used to update the Lagrange multipliers λ and the penalty coefficient r_p comes from Ref. 17.

V.1.3 Simple Constrained EGO Example

To demonstrate the effectiveness of this new constrained EGO formulation before applying it to the RBDO problem, a simple test is presented here. This test involves the "mystery" function from Ref. 69 and is stated as:

$$\begin{aligned} \text{minimize} \quad & f = 2 + 0.001(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(2 - x_2)^2 + 7\sin 0.5x_1 \sin 0.7x_1x_2 \\ \text{subject to} \quad & g = -\sin(x_1 - x_2 - \tfrac{\pi}{8}) \leq 0 \\ & 0 < x_1, x_2 < 5 \end{aligned} \tag{V.12}$$

The EGO solution to this problem is compared to that from the NIPS method.[59] Note that because EGO is a stochastic method, an average solution is given with some additional information on the best solutions found for 10 runs of this method.

Table V.1: Results for the simple constrained EGO example.

Optimization Method	Avg. Obj. Fn (Best Obj. Fn)	Avg. Violation (# violations)	Avg. Fn Evals (Obj. Fn/Con. Fn)
NIPS	-1.174	8.44E-5	51/32
EGO	-1.162 (-1.174)	-4.22E-3 (0)	25.5/25.5

The average EGO solution is very close to the optimal value found by the NIPS method,[59] and is capable of locating the same optimal value (in fact, for this study, this

value was found 5 times in the 10 EGO runs). EGO clearly does a good job of enforcing the constraint; none of the solutions were infeasible. EGO is also less expensive than NIPS, requiring only 61% of the total function evaluations (on average).

This successful test shows that this constrained EGO method holds promise for solving the RBDO problem, which also involves a nonlinear inequality constraint. The next section details several ways that this RBDO problem can be formulated.

V.2 Formulations for RBDO with EGO/EGRA

A variety of ways to incorporate EGO and EGRA into an RBDO formulation are investigated.

V.2.1 Nested RBDO with Separate Surrogates

In this method, EGO is used as the optimizer and for each candidate design point a full reliability analysis is performed using EGRA. For each EGRA analysis, the design point $\mathbf{d}$ is set; EGRA operates only over the random variables, $\mathbf{x}$. Because this must be performed at every design point and there is no sharing of information to either guide the optimizer or assist subsequent EGRA runs, this is expected to be the most expensive option. However, due to the efficiency of both EGO and EGRA, this is still expected to require fewer evaluations of the response function g than previously investigated nested RBDO methods.

V.2.2 Nested RBDO with a Single Surrogate

In this method, EGRA is first used to locate the limit state and build a single GP of g over the entire $\mathbf{d}+\mathbf{x}$ space in one step. With this GP built, any optimizer could be used at the design level; to calculate the probability of failure corresponding to any design point, one needs only to sample the GP previously built by EGRA. This optimization requires no additional evaluations of the response function when evaluating the probabilistic constraints. Because EGRA is allowed to search over the full range

of $\mathbf{d}+\mathbf{x}$ at all times instead of being restricted to adding points with set values of $\mathbf{d}$, this method is expected to be more efficient than the nested method where the models at each $\mathbf{d}$ are formed separately. However, the disadvantage to this method is that it may "waste" function evaluations by resolving the limit state in regions of the design space that may never be visited by the optimizer.

For the tests of this method in Section V.3, a full EGRA analysis is run using the true response function to verify the reliability of the optimal design. The function evaluations used in this step are not counted in the method's cost, but the result is used in determining whether the method converged to a feasible solution.

V.2.3 Sequential RBDO

This formulation uses a GP of the response function in the probabilistic constraint that spans $\mathbf{d}+\mathbf{x}$, but rather than update that model for every iteration of the design optimizer, it is only updated at its convergence. The algorithm follows this outline:

1. Build an initial GP for the objective function. At each design point, use EGRA to solve for the p_f at that point using a GP for g that spans the design and random variable spaces.

2. Use EGO to fully solve the RBDO problem. At each design point, to calculate the corresponding p_f, sample the current GP of g, holding $\mathbf{d}$ constant.

3. When EGO converges, calculate the true p_f using EGRA at $\mathbf{d}^*$. The points used by EGRA to resolve the limit state are then added to the GP of g.

4. Re-solve the RBDO problem using this new GP. Iterate until the method converges.

This method has one major advantage over the nested method with a single surrogate model: because it only adds points to the GP at candidate optimal points, it does not waste time increasing the accuracy of the GP in regions of the design space

that are far from optimal. Assessing convergence of this method is difficult. Due to the stochastic nature of EGO, the "converged to the same point" technique typically used for sequential problems cannot be applied. Instead, this study uses a metric on the accuracy of the underlying constraint function GP models. At the optimal point from an EGO solution, if the value of all of the constraint functions' GP models are within 1% of the actual constraint function values after verification, the method is said to have converged. However, while the latest analysis may have been necessary to ensure this accuracy in the models, it is not necessarily the optimal solution found by the algorithm. All of the candidate optimal designs are post-processed to find the best solution.

V.3 Computational Experiments

Several test problems are investigated to compare the results of these various formulations of RBDO with EGRA to one another and to other RBDO algorithms. For each problem, two "baseline" tests were run using the most successful methods from Ref. 24. The first test is a nested formulation using the Nonlinear Interior Point Solver (NIPS) from the OPT++[59] library for the design optimizer and the second-order iterated Advanced Mean Value method[24] (AMV^2+) to perform the reliability analysis. This nested formulation is used to find the baseline solution, but is expected to be relatively expensive. To provide a more fair comparison on the expense of the methods, a second more efficient baseline is run using a sequential, trust-region based method, again using NIPS and AMV^2+. These are both gradient-based methods; to simulate their use on implicit performance functions, numerical differentiation is used to compute the gradients and, where needed, Hessian information is generated using quasi-Hessians with Symmetric Rank 1 updating.

Note that while the RBDO discussion above was in terms of the probability of failure p_f, in the examples here, the reliability constraint is actually enforced in the beta-space, using the generalized reliability index $\beta^* = -\Phi^{-1}(p_f)$. This scheme was found

to be more computationally efficient in the RBDO studies in Ref. 24 and is therefore repeated here.

V.3.1 Short Column

The first application problem revisits the short column problem previously investigated in Section II.4.4. This problem involves the plastic analysis and design of a short column with rectangular cross section (width b and depth h) having uncertain material properties (yield stress Y) and subject to uncertain loads (bending moment M and axial force P).[24,51] The response function is defined as:

$$g = 1 - \frac{4M}{bh^2Y} - \frac{P^2}{b^2h^2Y^2} \tag{V.13}$$

The distributions for P, M, and Y are Normal($\mu = 500$, $\sigma = 100$), Normal($\mu = 2000$, $\sigma = 400$), and Lognormal($\mu = 5$, $\sigma = 0.5$), respectively, with a correlation coefficient of 0.5 between P and M (uncorrelated otherwise).

An objective function of cross-sectional area and a target probability of failure are used in the design problem:

$$\begin{aligned} \min \quad & bh \\ \text{s.t.} \quad & \beta^* \geq 2.5 \\ & 5.0 \leq b \leq 15.0 \\ & 15.0 \leq h \leq 25.0 \end{aligned} \tag{V.14}$$

Table V.2 gives a summary of the results from all investigated methods. Because both EGO and EGRA are stochastic methods, average solutions are given with some additional information on the best solutions found for 10 runs of these methods.

The EGO/Separate EGRA test performed well. It used approximately 16% of the function evaluations required by the Sequential NIPS/AMV^2+ baseline test. The objective function values are competitive with the baselines, and none of the runs pro-

Table V.2: Results for the short column RBDO example.

Design/Reliability Methods	Avg. Obj. Fn (Best Feasible)	Avg. β^* Violation (# violations)	Avg. g Evals (Best Feasible)
Nested NIPS/AMV^2+	216.7	0.0	4123
Sequential NIPS/AMV^2+	216.7	0.0	2434
EGO/Separate EGRA	216.7 (216.2)	-0.013 (0)	396.2 (321)
EGO/Single EGRA	218.8 (217.2)	-0.010 (5)	161.1 (149)
Sequential EGO/EGRA	217.3 (216.2)	-0.036 (0)	222.8 (361)

duced an infeasible result.

The EGO/Single EGRA method is the least expensive of the new methods (requiring less than half of the function evaluations of the EGO/Separate EGRA method), but its objective function values do not compare as well to those of the other methods. The average constraint violation is comparable to the other methods, but half of the runs produced slightly infeasible results. This is likely due to the inaccuracy of the model at the optimal point. Because the model was trained over the entire $\mathbf{d}+\mathbf{x}$ space, the model will be less accurate than if it were trained with $\mathbf{d}^*$ in the training set as it is for the Nested and Sequential methods.

The Sequential EGO/EGRA test provides a compromise between the nested methods. It required, on average, just 9% of the function evaluations needed by the Sequential NIPS/AMV^2+ baseline test; the average objective function value is within 1% of the baseline tests' results and it produced no infeasible solutions.

V.3.2 Cantilever Beam

This test problem investigates the cantilever beam problem that has been previously investigated in Sections II.4.3, III.3.2, and IV.3.1, as well as Refs. 24, 26, 63, 76, 88 and others to test both reliability analysis and design methods. This problem demonstrates the ability of the EGRA-based RBDO formulations to enforce multiple constraints.

This problem has two response functions of interest; one concerning the stress, g_S

and one concerning the displacement, g_D:

$$g_S = R - \frac{600}{wt}\left(\frac{Y}{t} + \frac{X}{w}\right) \tag{V.15}$$

$$g_D = D - \frac{4L^3}{Ewt}\sqrt{\frac{Y^2}{t^4} + \frac{X^2}{w^4}} \tag{V.16}$$

The limit states for these response functions are at $g_S = 0$ and $g_D = 0$; all variables are described in Table V.3.

Table V.3: Variable detail for the cantilever beam example.

Variable	Mean	COV	Distribution
Horizonal Load, X	500	0.2	Normal
Vertical Load, Y	1,000	0.1	Normal
Yield Strength, R	40,000	0.05	Normal
Modulus of Elasticity, E	29e6	0.05	Normal
Length, L	100	—	Deterministic
Max Displacement, D	2.25	—	Deterministic

The design problem is to minimize the weight (or, equivalently, the cross-sectional area) of the beam subject to probabilistic constraints on the displacement and stress. This RBDO problem is stated as:

$$\begin{aligned} \min \quad & wt \\ \text{s.t.} \quad & \beta_S^* \geq 3.0 \\ & \beta_D^* \geq 3.0 \\ & 1.0 \leq w, t \leq 4.0 \end{aligned} \tag{V.17}$$

Table V.4 gives a summary of the results from all investigated methods. Because both EGO and EGRA are stochastic methods, average solutions are given with some additional information on the best solutions found for 10 runs of these methods.

All of the EGO/EGRA methods converged to better objective function values than the Sequential NIPS/AMV^2+ method, which, given the drastic difference in cost between this and the Nested NIPS/AMV^2+ method, appears to have pre-converged to

a poor solution.

The EGO/Separate EGRA method is again the most expensive of the new methods. In a few cases, this method converged to a slightly infeasible solution, but on average does a good job enforcing the constraints. The EGO/Single EGRA and Sequential EGO/EGRA methods are comparable in cost, but the Sequential method clearly does a better job locating feasible solutions. The Sequential EGO/EGRA method, on average, requires less than 5% and 10% of the function evaluations (for g_S and g_D, respectively) of the Sequential NIPS/AMV^2+ and clearly produces a better optimal value. The average optimal solution from Sequential EGO/EGRA is within 1% of that found by the Nested NIPS/AMV^2+ at less than 2% and 4% of the cost.

V.3.3 Liquid Hydrogen Tank

This test problem involves the design of a liquid hydrogen fuel tank on a space launch vehicle.[57,75] The reliability analysis component of this problem was investigated in Section III.3.3. This problem demonstrates a special feature of applying EGRA-based RBDO to the variable insertion problem and also its ability to enforce system-level probabilistic constraints.

The tank has a honeycomb sandwich design with top and bottom plates made of aluminum alloy AL2024 with the sandwich material made of Hexcell 1/8-in.-5052.0015. The tank is subjected to stresses caused by ullage pressure, head pressure, axial forces due to acceleration, and bending and shear stresses caused by the weight of the fuel.

The random variables for this problem include the thickness of the plate $t_{plate} \sim$ Normal(μ_{plate}, σ=0.005), the thickness of the honeycomb $t_h \sim$ Normal(μ=0.1, σ=0.01), and the loads on the tank $N_x \sim$ Normal(μ=13, σ=60), $N_y \sim$ Normal(μ=4751, σ=48), and $N_{xy} \sim$ Normal(μ=-684, σ=11). The mean plate thickness μ_{plate} is the design variable.

Three modes of failure are considered for the tank: von Mises strength, isotropic strength, or honeycomb buckling. The honeycomb buckling response function is de-

fined by a response surface generated from the structural sizing program HYPERSIZER,[16] which is given in Ref. 75 as:

$$g_{HB} = 0.847 + 0.96x_1 + 0.986x_2 - 0.216x_3 + 0.077x_1^2 + 0.11x_2^2 + 0.007x_3^2 + 0.378x_1x_2 - 0.106x_1x_3 - 0.11x_2x_3 \tag{V.18}$$

where x_1, x_2, and x_3 are defined as:

$$x_1 = 4\left(t_{plate} - 0.075\right) \tag{V.19}$$

$$x_2 = 20\left(t_h - 0.1\right) \tag{V.20}$$

$$x_3 = -6000\left(\frac{1}{N_{xy}} + 0.003\right) \tag{V.21}$$

The response functions for the von Mises and isotropic strengths are defined by:

$$g_{vM} = \frac{84,000\ t_{plate}}{\sqrt{N_x^2 + N_y^2 - N_xN_y + 3N_{xy}^2}} - 1 \tag{V.22}$$

$$g_{ISO} = \frac{84,000\ t_{plate}}{|N_y|} - 1 \tag{V.23}$$

This system probability of failure is defined as:

$$p_f = P\left[g_{vM}\left(\mathbf{x}\right) < 0\ \cup\ g_{ISO}\left(\mathbf{x}\right) < 0\ \cup\ g_{HB}\left(\mathbf{x}\right) < 0\right] \tag{V.24}$$

The RBDO problem is stated as:

$$\begin{aligned} \min \quad & \mu_{plate} \\ \text{s.t.} \quad & p_f \leq 0.001 \\ & 0.035 \leq \mu_{plate} \leq 0.1 \end{aligned} \tag{V.25}$$

Note that the bound constraint in Eq. V.25 is not part of the original problem formulation, but is added here to compensate for a weakness in constrained EGO. Similar

to EGRA that has problems when the underlying GP model breaks down due to singularities in the response (see the discussion in Section II.4.5), EGO cannot deal with the large "flat" spots in the constraint function that are caused by searching regions of the design space where EGRA reports the probability of failure as either zero or one (and β^* is either ∞ of $-\infty$). EGRA will return $p_f = 0$ (or 1) when the limit state is not found within its search space. Because this space is restricted (in this work at $\pm 5\sigma$) the true probability of failure may not actually be 0 or 1, the limit state just lies beyond these bounds, so this problem arises as an artifact of the EGRA truncation of the search space. The need to determine bounds on the design space is similar to selecting a starting point for gradient-based optimization methods, where the user must take care not to start the optimization in one of these "flat" regions where the constraint gradient is zero. However, the EGO limitation is more onerous to the user because rigorous bounds on all design variables must be known rather than a single reasonable starting position. Future versions of EGO might be able to detect these "flat" regions and automatically remove them from the design space. Alternatively, EGO might adapt the bounds of the EGRA search space as needed to mitigate the effect of the random variable truncation.

EGRA-based RBDO is especially efficient when applied to this type of variable insertion problem. As was discussed extensively in the previous chapter, because EGRA operates independently of the input distributions, the model it creates remains accurate for changes in those distributions. Because of this, a simple nested formulation of EGO/EGRA RBDO is extremely efficient. At the first design point, a full EGRA analysis is performed. At subsequent design points, the GP from the initial search can be re-used, thus requiring few (if any) additional true function evaluations to determine the probability of failure at this new point.

Table V.5 gives a summary of the results using the Nested EGO/EGRA RBDO formulation compared to results for this problem from Ref. 57. This work uses a novel single-loop RBDO formulation using FORM and Pandey's method[61] to calculate the

system reliability. Because both EGO and EGRA are stochastic methods, an average solution is given with some additional information on the best solution found for 10 runs of the method.

Because the RBDO method used in Ref. 57 relies on FORM (and therefore linear approximations to the limit states), it converges to a conservative solution. The true probability of failure at this μ_{plate} value is approximately 0.0007 rather than the desired value of 0.001. Moreover, this method required a total of 264 function evaluations to reach this result. The EGO/Separate EGRA method can find a more accurate result, and requires an average of only 38.4 function evaluations (only 14.5% of the cost).

V.3.4 Steel Column

This test problem involves the trade-off between cost and reliability for a steel column.[51] Calculation of the reliability at a given design was performed in Section II.4.5. The cost is defined as

$$Cost = \mu_B \mu_D + 5\mu_H \tag{V.26}$$

where μ_B, μ_D, and μ_H are the means of the flange breadth, flange thickness, and profile height, respectively. Nine uncorrelated random variables are used in the problem to define the yield stress F_s (Lognormal with μ/σ = 400/35 MPa), dead weight load P_1 (Normal with μ/σ = 500000/50000 N), variable load P_2 (Gumbel with μ/σ = 600000/90000 N), variable load P_3 (Gumbel with μ/σ = 600000/90000 N), flange breadth B (Lognormal with $\mu/\sigma = \mu_B/3$ mm), flange thickness D (Lognormal with $\mu/\sigma = \mu_D/2$ mm), profile height H (Lognormal with $\mu/\sigma = \mu_H/5$ mm), initial deflection F_0 (Normal with μ/σ = 30/10 mm), and Young's modulus E (Weibull with μ/σ = 21000/4200 MPa). The limit state has the following analytic form:

$$g = F_s - P\left(\frac{1}{2BD} + \frac{F_0}{BDH}\frac{E_b}{E_b - P}\right) \tag{V.27}$$

Table V.4: Results for the cantilever beam RBDO example.

Design/Reliability Methods	Avg. Obj. Fn (Best Feas.)	Avg. Viol. β_S^* (# viols)	Avg. Viol. β_D^* (# viols)	Avg. Evals g_S (Best Feas.)	Avg. Evals g_D (Best Feas.)
Nested NIPS/AMV^2+	9.520	0.0	0.0	4176	4599
Sequential NIPS/AMV^2+	11.076	0.0	0.0	1611	1593
EGO/Separate EGRA	9.550 (9.519)	-0.0495 (4)	-0.0513 (2)	122.6 (123)	305.8 (352)
EGO/Single EGRA	9.526 (9.523)	0.0222 (4)	-0.282 (3)	106.9 (88)	144.2 (174)
Sequential EGO/EGRA	9.592 (9.519)	-0.113 (0)	-0.152 (0)	70.4 (83)	160.1 (196)

Table V.5: Results for the liquid hydrogen tank RBDO problem.

RBDO Method	Avg. Obj. Fn (Best Feas.)	Avg. Viol. (# viols)	Avg. Evals g_{vM} (Best Feas.)	Avg. Evals g_{ISO} (Best Feas.)	Avg. Evals g_{HB} (Best Feas.)
Single-Loop FORM/Pandey[57]	0.07433		92	84	88
EGO/Separate EGRA	0.07376 (0.07377)	-6.7E-4 (3)	21.5 (22)	6.9 (7)	10 (10)

where

$$P = P_1 + P_2 + P_3 \tag{V.28}$$

$$E_b = \frac{\pi^2 EBDH^2}{2L^2} \tag{V.29}$$

and the column length L is 7500 mm.

This design problem demonstrates design variable insertion into random variable distribution parameters through the design of the mean flange breadth, flange thickness, and profile height. The following RBDO formulation maximizes the reliability subject to a cost constraint:

$$\begin{aligned} \max \quad & \beta^* \\ \text{s.t.} \quad & Cost \leq 4000. \\ & 200 \leq \mu_B \leq 250 \\ & 16 \leq \mu_D \leq 20 \\ & 100 \leq \mu_H \leq 150 \end{aligned} \tag{V.30}$$

Recall that in Section II.4.5, the search space for several variables had to be restricted to prevent problems with the GP model that prevented EGRA from being effective in solving this problem. Similarly, the search bounds for the RBDO problem must also be restricted as shown in Eq. V.30.

Table V.6 gives a summary of the results from the investigated methods. For this test, FORM is used (rather than AMV^2+ as in the previous RBDO tests) because the results in Section II.4.5 showed that the second order reliability analysis methods did not properly converge when applied to this problem. Also note that no sequential baseline is provided because the Sequential NIPS/FORM method failed to find an acceptable result. Because both EGO and EGRA are stochastic methods, average solutions are given with some additional information on the best solutions found for 10

runs of these methods.

Table V.6: Results for the steel column RBDO example.

Design/Reliability Methods	Avg. β^* (Best Feasible)	Avg. Cost (# violations)	Avg. g Evals (Best Feasible)
Nested NIPS/FORM	2.953	4000.0	9500
EGO/Separate EGRA	2.943 (2.986)	3998.08 (0)	156.1 (129)

This problem once again demonstrates the vast savings that EGRA can provide when applied to variable insertion problems. It is, on average, capable of producing a result within 0.5% of the baseline result at less than 2% of the cost.

However, this example also demonstrates a minor flaw in performing RBDO with EGO as the design optimizer. The optimal point found by the Nested NIPS/FORM method includes $\mu_B^* = \mu_B^{\min}$ and $\mu_H^* = \mu_H^{\min}$. Recall that the implementation of EGO used in this work utilizes the DIRECT[34] method to solve the max(EIF) problem. Because this method repeatedly subdivides the search space, it is impossible for it to find an optimal solution on the boundaries of that space. This in turn makes it impossible for EGO to find an optimal solution that involves the variable bounds.

V.3.5 Bistable MEMS

The final RBDO test involves the design of the bistable MEMS device presented in Section II.4.6. The tapered beam legs of the bistable MEMS mechanism are parameterized by the 13 design variables shown in Figure V.1, including widths and lengths of beam segments as well as angles between segments. For simulation, a symmetry boundary condition allowing only displacement in the negative y direction is applied to the right surface ($x = 0$) and a fixed displacement condition is applied to the left surface. With appropriate scaling, this allows the quarter model to reasonably represent the full four-leg switch system.

Given the 13 geometric design variables $\mathbf{d} = [L_1, L_2, L_3, L_4, \theta_1, \theta_2, \theta_3, \theta_4, W_0, W_1, W_2, W_3, W_4]$ and the specified random variables $\mathbf{x} = [\Delta W, S_r]$, a reliability-based design optimization problem is formulated to achieve a design that actuates reliably with

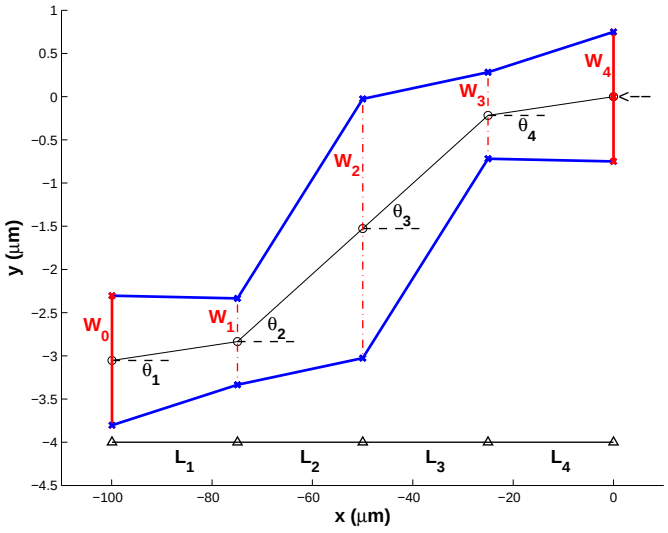


Figure V.1: Design parameters for the tapered-beam fully-compliant bistable mechanism (geometry not to scale). Displacement is applied in the negative y direction at the right face ($x = 0$), while at the left face, a fixed displacement condition is enforced.

at least 5 μN force. The RBDO formulation uses the limit state $g(\mathbf{x}) = F_{min}(\mathbf{x})$ and failure is defined to be actuation force with magnitude less than 5.0 ($F_{min}(\mathbf{x}) > -5.0$). A probability of failure $p_f \leq 0.02275$ is required.

$$
\begin{array}{lrcc}
\max & & F_{min}(\mathbf{d}, \mu_{\mathbf{x}}) & \\
\text{s.t.} & & P\left[F_{min}(\mathbf{d}, \mathbf{x}) > -5.0\right] & \leq 0.02275 \\
50 & \leq & F_{max}(\mathbf{d}, \mu_{\mathbf{x}}) & \leq 150 \\
 & & E_2(\mathbf{d}, \mu_{\mathbf{x}}) & \leq 8 \\
 & & S_{max}(\mathbf{d}, \mu_{\mathbf{x}}) & \leq 3000 \\
20 & \leq & L_1, L_2, L_3, L_4 & \leq 35 \\
0 & \leq & \theta_1, \theta_2, \theta_3, \theta_4 & \leq 5 \\
1 & \leq & W_0, W_1, W_3, W_4 & \leq 3 \\
2 & \leq & W_2 & \leq 5
\end{array}
\tag{V.31}
$$

where $\mu_{\mathbf{x}}$ represents the mean values of $\mathbf{x}$ and the probabilistic constraint is enforced through $\beta^* \geq 2.0$ rather than on the probability of failure as shown.

This problem further illustrates a weakness in EGO-based RBDO. As discussed in Section V.3.3, EGO cannot properly account for any "flat" regions in the design space

where the probability of failure is reported by EGRA as either zero or one. For that RBDO test, the bounds of only a single design variable had to be set to avoid this issue. However, defining appropriate bounds is considerably more difficult in this 13-variable design space. To further complicate matters, a given design may not exhibit the required bi-stability in its load-deflection curve, meaning that F_{min} is undefined and the numerical analysis of that design fails. In either event, whether EGRA gives $p_f = 0$ or 1 or F_{min} is undefined, it is essentially impossible to determine which variable or combination of variables need to be restricted to avoid the problem.

In order to optimize this design, a different strategy had to be employed. Initial bounds were first tightly set around the MVFOSM solution shown in Table V.7. An RBDO solution using the EGO/Separate EGRA formulation was then found over these restricted bounds. If this solution was on the boundary of the design space for any of the variables, the search space for those variables was translated and the analysis was repeated. This trust-region-like strategy allowed the EGO/EGRA method to locate a solution, but has several flaws. First, it requires far greater expense than a method that could consider the entire design space at once because the entire RBDO problem (expensive by itself) must be repeatedly solved. Second, by restricting EGO to a localized region, the method is no longer able to find the global optimum, reducing the usefulness of this method.

However, the benefits of applying EGRA to this RBDO problem are still clear. The analysis started at a previously converged optimal solution using a different reliability analysis method, which was demonstrated to be infeasible in Section II.4.6, and then moved away from this infeasible design because of the increased accuracy that EGRA provides. Table V.7 displays the results from solving this problem with EGRA compared with previous work using MVFOSM, AMV^2+, and FORM.[1,2,24,29] The comparatively small value of F_{min} in the EGRA solution coupled with the slack in the reliability constraint hints that this solution is unlikely to be the global optimum and that this bistable MEMS design could still benefit from further investigation.

Table V.7: Results for the bistable MEMS RBDO example.

variable/metric	MVFOSM	AMV^2+	FORM	EGRA
$L_1\,(\mu m)$	19.23	28.04	28.06	19.18
$L_2\,(\mu m)$	28.44	24.42	24.45	28.38
$L_3\,(\mu m)$	14.44	30.58	30.68	14.38
$L_4\,(\mu m)$	35.00	30.55	30.66	34.99
$\theta_1\,(deg.)$	2.733	4.200	4.189	2.749
$\theta_2\,(deg.)$	2.260	2.481	2.488	2.244
$\theta_3\,(deg.)$	2.719	2.465	2.478	2.737
$\theta_4\,(deg.)$	3.230	2.384	2.390	3.265
$W_0\,(\mu m)$	1.058	1.355	1.346	1.100
$W_1\,(\mu m)$	2.038	1.275	1.265	2.052
$W_2\,(\mu m)$	2.390	3.481	3.488	2.340
$W_3\,(\mu m)$	1.312	2.006	2.004	1.319
$W_4\,(\mu m)$	1.000	1.333	1.333	1.072
$F_{min}\,(\mu N)$	-5.896	-6.188	-6.292	-8.798
β	2.000	1.998	1.999	—
β^*	1.227	—	—	2.401
$F_{max}\,(\mu N)$	50.01	57.67	57.33	56.11
$E_2\,(\mu m)$	5.804	5.990	6.008	7.282
$S_{max}(MPa)$	1563	1333	1329	1144

V.4 Summary

While reliability analysis is a field of its own and is commonly treated as a stand-alone task, it is often part of a larger process. The reliability at a given design is first analyzed, then if that reliability is too low (or even too high) the design is adjusted and the reliability analysis is repeated. This process of searching for the optimal balance between some objective (cost, performance, etc.) and the reliability can be automated through reliability-based design optimization.

This chapter explored how EGRA can be used in solving this class of problems. Because EGRA does not lend itself to use with gradient-based design optimizers (it does not provide analytic derivatives in its result and its stochastic nature prevents the use of numerical differentiation) a gradient-free optimizer is needed. One such optimizer is the Efficient Global Optimization method on which EGRA is based. EGO was originally derived for unconstrained optimization, so in order to use it in RBDO, a constrained formulation was derived based on an augmented Lagrangian formulation

and the expected violation function.

By coupling EGRA with EGO, it is possible to formulate nested, single-loop, and sequential RBDO methods, all of which were applied to a collection of example problems. All of these EGRA-based methods proved to be considerably less expensive than their MPP-based counterparts, for the sizes of problems investigated. As was discussed in Section II.5, EGRA may not scale well for problems with large numbers of random variables and the same is true of EGO applied to problems with large numbers of design variables. Because both EGO and EGRA are stochastic methods, they cannot provide the exact solution like these gradient-based methods can, but the EGO/EGRA solutions are comparable.

The nested method with a single surrogate model is the most efficient, but also the least accurate and least consistent. Because this method does not attempt to fully resolve the limit state at a given design point, it is not uncommon for this method to converge to an infeasible solution due to the lack of fidelity in the underlying model.

The nested method with separate surrogate models is the most expensive of the EGO/EGRA methods because it performs a full reliability analysis at every design point visited by the optimizer. However, it is worth noting that even though it is the most expensive EGO/EGRA method, it is still considerably less expensive than the MPP-based methods with which it was compared; consistently requiring less than 20% of the function evaluations. This method occasionally converged to solutions that were slightly infeasible, but the resulting constraint violation was typically within the variance of the EGRA solution.

The sequential method typically required a computational expense somewhere between the nested methods, and consistently converged to feasible solutions. However, the optimal solutions to which this method converged were occasionally overly conservative, leaving slack in the constraint that could be used to improve the objective. In general, because the nested method with separate surrogates already provides substantial computational savings over traditional methods, if it can be afforded, it is

recommended. But for problems where efficiency is of the utmost importance, the sequential method provides an able alternative.

Using EGO as the design optimizer poses a problem due to difficulties with the GP models on which it operates. It is not uncommon for combinations of the design variables near their bounds to create regions where, due to EGRA's truncation of the random variable space over which it searches for the limit state, the probability of failure is reported as either zero or one. But to accommodate the use of EGO, these regions must be removed from the search space because these "flat" regions create problems for the Gaussian process model of the constraint violation. Finding the proper bounds for all of the design variables can be an arduous task that will require more study before these EGO/EGRA methods are adopted in practice.

Using EGRA to perform the reliability analysis in an RBDO process produces a significant advantage for so-called variable insertion problems where the design variables are distribution parameters. As was discussed in Chapter IV, because EGRA constructs the GP model independently of the input distributions, the model it trains does not necessarily have to be rebuilt when those distributions change. For this type of RBDO problem, the nested formulation can be used to perform the entire optimization at approximately the cost of a single reliability analysis. Moreover, because that reliability analysis is performed using EGRA, as Chapter II showed, that cost is dramatically less expensive than other available methods.

CHAPTER VI

CONCLUSIONS

Computational modeling and performing reliability analyses using those models are growing fields across a wide variety of engineering disciplines. As these models have gained in fidelity, they have become increasingly expensive to evaluate. A single analysis of a large model can take hours or even days to evaluate. This makes common reliability analysis methods like Monte Carlo sampling impractical due to their vast computational expense. To reduce this cost, analytical reliability analysis methods based on the concept of the Most Probable Point like FORM, AMV, AMV+, etc., have been developed. These methods, while efficient, are only accurate if the MPP is successfully located and the limit state is linear in the reduced normal space. Because the MPP is sought by solving an equality-constrained optimization problem with a gradient-based solver, it is not uncommon for this search to fail. This is especially true when applied to implicit response functions such as finite element models where the gradients must be calculated numerically and may therefore be unreliable. Additionally, few engineering applications are truly linear in their behavior; even when they are, they may not remain linear in the reduced normal space, a transformation which is generally nonlinear. Approximating the response as linear is convenient when estimating the reliability, but will lead to inaccuracies when the approximation is poor.

Throughout the computational experiments performed in this dissertation, MPP-based methods that incorporate second-order response function information were explored. These methods were typically more efficient in the search for the MPP than using first-order information alone. By accounting for some moderate nonlinearity in the limit state, they can also be more accurate. However, these methods occasionally failed to converge to the MPP or were forced to resort to first-order limit state approximations due to numerical exceptions in the second-order formulations. In general,

all MPP-based reliability analysis methods suffer from two basic problems: 1) potential failure in locating the MPP, and 2) a low-order approximation based solely on information in the vicinity of this single point on the limit state is rarely accurate.

Another alternative reliability analysis method involves the use of surrogate models. These methods collect data on the true response function, then use these to construct an approximate that is less expensive to evaluate. This new surrogate model is then used in place of the true function in a sampling method. Because these samples are evaluated using the surrogate model, and relatively few samples of the true function were required to construct that surrogate, this can be a very efficient reliability analysis method. However, the accuracy of this method depends on the accuracy of the surrogate model. Low-order polynomial functions are commonly used, but these can be inaccurate for the same reason that low-order polynomial approximations to the limit state can be inaccurate when used in MPP-based methods. Recent research has focused on more flexible surrogate models, such as Gaussian process models, that can be far more accurate.

The main contribution of this dissertation has been to greatly improve the efficiency with which these surrogate models are constructed. Common practice has been to seek global accuracy of these surrogate models. Error measures are calculated throughout the random variable space, and additional data is collected if the errors are too high. But global accuracy of the model is not needed when the model is only intended for use in calculating the probability of failure. When the model is sampled, the actual value of the response at a sample point is not important - only whether that value is greater or less than the response level of interest, i.e. whether the point is in the success or failure region. If the contour that separates these regions (the limit state) is accurately captured, then the accuracy of the model elsewhere in the space is immaterial. This type of locally accurate model can be constructed using a Gaussian process model by collecting training data in the region where accuracy is desired. Of course, the location of the limit state contour throughout the random variable space cannot

be known *a priori* (hence the search for it in the MPP-based methods), so focusing the training data for the surrogate model in this region is not straightforward.

The Efficient Global Reliability Analysis method was presented in Chapter II to meet this challenge. This method begins with a small number of randomly selected training points that are evaluated using the true response function. A Gaussian process model is fit to these data; because the data is scarce, the model is inaccurate with a large amount of variance. The predicted values and the variance in this model are combined through what has been termed the expected feasibility function to quantify the expectation that any given point in the random variable space lies on the limit state. A global optimizer is then used to find the point with the maximum expected feasibility. This point is then evaluated using the true response function and this new piece of data is added to the training data and a new Gaussian process model is constructed. This process is repeated until the maximum expected feasibility is acceptably small. By focusing the training data near the limit state, a locally accurate model can be constructed with a small amount of data, making this method highly efficient. By iteratively searching for the limit state using a metric that incorporates the variance in the model and seeking to reduce that variance in areas near the limit state, this method creates a highly accurate surrogate model. This model can then be sampled using any appropriate method to create a reliability analysis method that is both efficient for computationally expensive response functions and accurate for arbitrarily-shaped limit states.

The advantages of EGRA were demonstrated through several application problems. It was shown to consistently have a computational cost competitive with (and often superior to) the most efficient MPP-based methods, coupled with the accuracy of exhaustive Latin hypercube sampling. The method becomes more expensive as the number of random variables being considered increases, but appears to scale well (MPP-based methods also become more expensive with additional random variables when the numerical differentiation to calculate the gradients requires additional func-

tion evaluations). By constructing the surrogate model in the original space (rather than the reduced normal space), EGRA is completely independent of the distribution types and correlation structure describing the random variables, making it equally efficient and accurate for any combination of types or correlations.

The cost of EGRA is driven by two factors: the behavior of the limit state (highly nonlinear contours require more data to model), and the length of the limit state contour in the search space. This second factor leads to an interesting advantage of EGRA; solving high-reliability problems, which are the problems typically of interest to engineers, is actually less expensive using EGRA than solving low-reliability problems. A problem has a high reliability if it has a small failure region, which naturally requires a shorter contour to define than a large region would. This shorter limit state contour requires fewer samples to model, making EGRA less expensive.

The demonstration problems also illustrated a weakness in EGRA due to its reliance on Gaussian process models. These models create smooth interpolations of the training data. If the true response has abrupt changes or discontinuities, the model will fail to produce an accurate model, leading EGRA to be both more expensive and inaccurate. If this behavior in the response is localized (due to a singularity, for instance), then this region can simply be removed from the search space, thus restoring EGRA to the efficient and accurate method that is expected.

Chapter III discussed the application of EGRA to system-level reliability analysis. For system problems, the failure region is typically bounded by only portions of the various component limit states. These portions make up the so-called composite limit state. By reformulating the expected feasibility function to search for points on the composite limit state, EGRA is able to focus the training data near only the portions of the component limit states that bound the system failure region rather than attempting to model the entire limit state. Moreover, if a particular component does not contribute to the overall system failure (i.e., its limit state does not bound the system failure region), EGRA is capable of "ignoring" this component, requiring

no additional evaluations of its response beyond the initial study. Because the cost of EGRA will be driven, in part, by the length of the composite limit state contour, which bounds some small failure region, the expense of EGRA when applied to system-level problems is comparable to solving the much simpler component-level problem (there is a slight increase in cost due to the initial study that evaluates all of the component response functions). This makes EGRA a highly efficient way to perform system-level reliability analysis. The efficiency and accuracy of this method were demonstrated through its application to a collection of example problems. Both parallel and series systems were explored, and EGRA proved equally efficient and accurate for both formulations.

Chapter IV presented the challenges of reliability analysis when the input distributions are uncertain. This is a common problem in solving real-world problems because the input distributions are typically estimated from a limited set of test data. Unfortunately, the effects of this uncertainty are largely ignored in practice because available methods for quantifying them have suffered the same problems as existing reliability analysis methods: they are either too expensive or they rely on simplifying assumptions that can make them inaccurate. The nested approach based on Bayesian inference and Markov Chain Monte Carlo sampling that is used in this chapter is not altogether new, but the application of EGRA to this problem has made such a formulation practical. Using EGRA, the entire posterior distribution of the probability of failure can be calculated at a computational expense that is little more than performing a single reliability analysis (which, of course, using EGRA, is already more efficient than other methods).

Chapter V explored how EGRA can be used in solving reliability-based design optimization problems. By coupling EGRA with the Efficient Global Optimization method on which it is based, nested, single-loop, and sequential methods were formulated. All of these methods are considerably less expensive than their MPP-based counterparts, and while, because they are stochastic, they cannot provide the exact

solution like the gradient-based methods can, they are still very accurate. The single-loop method is the most efficient, but also the least accurate and least consistent; it is not uncommon for this method to converge to an infeasible solution. The nested method is the most expensive of the EGO/EGRA methods, but occasionally converged to solutions that were slightly infeasible. The sequential method typically had a computational expense between the nested and single-loop methods, and consistently converged to feasible solutions, but its optimal solution was occasionally overly conservative. The use of EGO as the design optimizer presents a problem due to potential complications with the GP models on which it operates. Regions of the design space where the probability of failure is either zero or one must be removed from the search space because these "flat" regions create problems for the Gaussian process model of the constraint violation. Finding the proper bounds of the design variables can be an arduous task that will require more study before these methods are adopted in practice. Using EGRA to perform the reliability analysis produces a significant advantage when the design variables are distribution parameters. As was discussed in Chapter IV, because EGRA constructs the GP model independently from the distributions, the model does not have to be rebuilt when those distributions change. For this type of RBDO problem, the entire optimization can be performed at approximately the cost of a single reliability analysis.

VI.1 Future Work

Efficient Global Reliability Analysis has shown great promise, but there are multiple opportunities to seek improvements and extensions to the method. The following sections will identify a few of these, as well as point out a few additional types of problems where EGRA might be successfully applied.

VI.1.1 Extensions to EGRA

Without major changes to the core algorithm, there are several ways that EGRA could be extended to improve its efficiency and provide additional information in its results.

The first extension is to develop parallel implementations of both EGRA and EGO. It is clear from Figures II.4 and II.9 that at each iteration of EGO and EGRA that there are multiple local optima to the max(EIF) and max(EFF) problems. In fact, it is because of these multiple local optima that a global optimizer like DIRECT is needed to locate the global optimum. Furthermore, Figures II.5 and II.10 show that after the global optimum was found in the previous iteration, points that were local optima remain local optima in the next iteration. A parallel implementation of EGO and EGRA would locate all of the local optima at each iteration. These new points would then be evaluated in parallel and added to the GP in the next iteration. By locating multiple points in each iteration, these methods will not necessarily require fewer evaluations of the response function, but because multiple new points can be evaluated and added to the GP simultaneously, fewer iterations will be required and the wall time of the methods will be reduced. The challenge in developing this parallel implementation is choosing an optimizer that can reliably locate all of the local optima. The DIRECT algorithm may be capable doing this, but the implementation from Ref. 34 that was used in this dissertation does not provide this information.

The next extension would allow EGRA to solve for multiple response levels in order to solve for the full CDF of the response rather than the probability of failure at a single response level. A straightforward way to find the probability at multiple response levels with EGRA would be to first train the GP for one response level, then use this GP as the starting point in the search for the next level, iterating until the limit states are accurately resolved for all levels of interest. The subsequent solves would likely require relatively few samples because the variance in the GP has been greatly reduced in searching for the previous limit states, so this would likely be an efficient

way to solve for the full CDF. However, it may be possible to extend the algorithm to solve for the multiple response levels all at once by modifying the expected feasibility function. If such a formulation is possible, it could improve the efficiency when multiple response levels are of interest.

The final extension would be to account for any remaining model error when reporting the result from EGRA. An important feature of EGRA is how it minimize the surrogate model error by using a convergence measure that is based in part on the variance of the GP. Many of the results shown in Chapter II demonstrated that the remaining error in the model was small enough to be insignificant when compared to the sampling variance. Of course, this may not be true for all problems; and even if it is, this is still useful information to report to the user. Quantifying the impact of the Gaussian process model error on the reported probability of failure is not entirely straightforward, but should not require any changes to the EGRA algorithm itself.

VI.1.2 Additional Applications of EGRA

Because it is a general-purpose reliability analysis methods, there are a wide variety of engineering applications to which EGRA might be applied. This section will focus on two classes of problems that EGRA might be successful in solving.

The first class of problems is inverse reliability analysis. Formulations for this type of analysis were discussed in Chapter I. These MPP-based methods relied on a $p \rightarrow \beta$ relationship based on either a first- or second-order approximation to the limit state. They then solved an optimization problem to find a point that lied this distance β from the mean response. EGO could easily be used to solve this optimization problem and locate this MPP, but for the arbitrarily-shaped limit states for which EGRA was created, there is no $p \rightarrow \beta$ relationship, so the response level at this MPP loses meaning - it is not necessarily the response level that corresponds to the desired probability. One way to solve this problem with EGRA would be to select some candidate response level, solve the full forward problem, then use the resulting probability to

update the target response level, and iterate until the level that produces the desired probability is found. Because the GP model could be reused in each iteration, this would likely be a sufficiently efficient inverse reliability method. However, fully resolving the limit state at these intermediate response levels is potentially wasteful. A more efficient method might update the response level for which EGRA is searching in each iteration rather than fully resolving the limit state at any intermediate level.

Chapter III explored the application of EGRA to system-level problems. Both series and parallel systems were analyzed, but combined systems were not. For instance, the three-component system investigated in Section III.3.1 could be formulated as a combined system by:

$$p_f = P\Big[\big[g_1(\mathbf{x}) < 0 \cap g_2(\mathbf{x}) < 0\big] \cup g_3(\mathbf{x}) > 0\Big] \tag{VI.1}$$

A composite limit state can still be formed from portions of the components, so conceptually, EGRA can be applied in the same way as it was for series and parallel systems. However, more sophisticated logic will be required to modify the expected feasibility function; the min/max logic used for the simpler systems is insufficient.

VI.1.3 Improvements to EGO and EGRA

While EGRA was successfully applied to a variety of test problems, there are some weaknesses in both it and EGO that need to be overcome to improve their robustness and generally applicability.

EGRA had problems with one of the example problems in Chapter II due to a singularity in the response function. The form of the GP model breaks down due to this sharp change in the response, causing EGRA to fail. This example problem was successfully solved by removing the portion of the search space near this singularity. But it is unreasonable to expect the user to know *a priori* where problem areas might exist in the response function, so EGRA needs to be improved to remove this weakness. Two possibilities are envisioned. First, it might be possible to reformulate the GP that

EGRA is training to better deal with potential abrupt changes in the response. Exactly how this might be done is a point of research, but possibilities might include multiple independent GPs for different regions of the search space, or a single GP with a discontinuous trend function. Second, EGRA could detect response values that might cause problems for the GP, choose not to add them to the training data, and automatically reduce the search space. This second idea would be simpler to implement, but if it removes an area of the search space that contains a significant portion of the limit state, it will create errors in the reliability estimates.

Another improvement, again involving the potential failure of a GP model, seeks to make EGO more generally applicable to RBDO problems. As was seen in several of the examples in Chapter V, the design space had to be severely restricted to prevent the optimizer from visiting areas where EGRA estimated the probability of failure as either zero or one because these "flat" regions cause problems for the GP of the constraint violation. Determining appropriate bounds can be difficult and time-consuming, making it difficult to use this method in practice. A potential method to eliminate this problem might be to introduce trust-region machinery. The user would need to select an initial point (as is already required for applying gradient-based methods) and some initial bounds in the vicinity of this point are constructed. The EGO/EGRA RBDO method (any of the nested, sequential, or single-loop methods should be possible) then proceeds in this limited space. If a "bad" region of this space is found, the trust region would be contracted in one or more dimension. If EGO converged on the bounds of the space, then the trust region would be translated and/or expanded. By not attempting to search the entire design space all at once, difficult regions of the space could be avoided and could possibly even make the method more efficient.

An additional way these "flat" regions might be handled is to allow EGO to adaptively select the bounds for the random variables over which EGRA will search for the limit state. For instance, default bounds of $\pm 5\sigma$ might be initially used, but when

EGRA returns $p_f = 0$ or 1, these bounds could be expanded and the analysis re-run to allow EGRA to resolve the smaller (or closer to 1) probability. This would expand the design space over which EGO could search, but would add to the expense of EGRA in these low (or high) probability regions where it might be sufficient (in terms of assessing feasibility of the design point) to estimate the probability as simply 0 or 1.

The final potential improvement seeks to reduce the already small number of function evaluations required by EGRA to model the limit state. It is evident from Figures II.13, III.4, and III.6 that EGRA commonly performs several evaluations of the response function on the boundaries of the search space. This is because the variance of the GP is typically large in this region where it is attempting to extrapolate information into this space rather than interpolating between the training data. However, accuracy of the limit state in this region is relatively unimportant because the likelihood of events in this region are so small. A scale factor could be added to the EFF to enforce that accuracy in the highly probable regions is more important than in the tails of the distributions. This probability bias will encourage EGRA to focus the training data near the portion of the limit state where accuracy is needed and could have a significant impact on the efficiency of the method. However, one disadvantage to this is that the locations of the sample points are no longer independent of the input distributions, meaning that many of ideas put forth in Chapter IV on propagating distribution uncertainty with EGRA will no longer be applicable.

CHAPTER A

DERIVATIONS

A.1 Expected Improvement Function

The expected improvement function is defined as:[47]

$$EI(\mathbf{x}) = E\left[\max\left(G(\mathbf{x}^*) - \hat{G}(\mathbf{x}),\, 0\right)\right] \tag{A.1}$$

where $G()$ is the true response function being modeled, $\mathbf{x}^*$ is the location of the current best solution found so far, and $\hat{G}()$ is the Gaussian process model approximation of the response function, which, at any point $\mathbf{x}$, provides a normally distributed random variable: $\hat{G}(\mathbf{x}) \sim N[\mu(\mathbf{x}), \sigma(\mathbf{x})]$.

For clarity, the dependence on $\mathbf{x}$ is removed from all arguments. The Gaussian process predictor is rewritten as $\hat{G}(g)$ to emphasize that it predicts values of the response function; g is then a realization of the normal probability density $\hat{G}$ and $g^* \equiv G(\mathbf{x}^*)$. It is clear that the maximization term in Eq. A.1 follows the relationship:

$$\begin{aligned} \max\left(g^* - g,\, 0\right) &= g^* - g \quad \text{for} \quad -\infty < \; g < \; g^* \\ &= 0 \qquad\quad \text{for} \quad g^* < \; g < \; \infty \end{aligned} \tag{A.2}$$

The expected value can then be computed by integrating over the interval where the function is nonzero:

$$\begin{aligned} EI &= \int_{-\infty}^{g^*} (g^* - g)\; \hat{G}(g)\; dg \\ &= \int_{-\infty}^{g^*} g^*\, \hat{G}(g)\; dg - \int_{-\infty}^{g^*} g\, \hat{G}(g)\; dg \\ &= g^*\, \Phi\left(\frac{g^* - \mu}{\sigma}\right) - \int_{-\infty}^{g^*} g\, \hat{G}(g)\; dg \end{aligned} \tag{A.3}$$

The second term can be written as:

$$\int_{-\infty}^{g^*} g\,\hat{G}(g)\,dg = \int_{-\infty}^{g^*} \frac{g}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{g-\mu}{\sigma}\right)^2\right] dg \tag{A.4}$$

Now allow the variable transformation $v = \frac{g-\mu}{\sigma}$, which gives $dg = \sigma\,dv$. Substituting these in:

$$\int_{-\infty}^{g^*} g\,\hat{G}(g)\,dg = \int_{-\infty}^{v^*} (v\,\sigma + \mu)\frac{1}{\sqrt{2\pi}} \exp\left[\frac{-v^2}{2}\right] dv \tag{A.5}$$

where $v^* = \frac{g^*-\mu}{\sigma}$. This equation can then be broken into two parts, giving:

$$\int_{-\infty}^{g^*} g\,\hat{G}(g)\,dg = \frac{\sigma}{\sqrt{2\pi}} \int_{-\infty}^{v^*} v\,\exp\left[-\frac{v^2}{2}\right] dv + \mu \int_{-\infty}^{v^*} \frac{1}{\sqrt{2\pi}}\,\exp\left[-\frac{v^2}{2}\right] dv \tag{A.6}$$

The second term is the integal of the standard normal PDF, which can be expressed using the CDF, giving:

$$\int_{-\infty}^{g^*} g\,\hat{G}(g)\,dg = \frac{\sigma}{\sqrt{2\pi}} \int_{-\infty}^{v^*} v\,\exp\left[-\frac{v^2}{2}\right] dv + \mu\,\Phi(v^*) \tag{A.7}$$

In the integral in the first term above, introduce the change of variables $w = \frac{v^2}{2}$, which gives $dv = \frac{1}{v}\,dw$. This reduces the integral to:

$$\int_{-\infty}^{v^*} v\,\exp\left[-\frac{v^2}{2}\right]\,dv = \int_{\infty}^{w^*} \exp\left[-w\right]\,dw \tag{A.8}$$

where $w^* = \frac{1}{2}(\frac{g^*-\mu}{\sigma})^2$. Note that the lower bound has changed from $-\infty$ to ∞ because the transformation involves squaring. Evaluating this gives:

$$\int_{-\infty}^{v^*} v\,\exp\left[-\frac{v^2}{2}\right]\,dv = -\exp\left[-w\right]\,|_{\infty}^{w^*} = -\exp\left[-w^*\right] \tag{A.9}$$

Now substitute Eq. A.9 into Eq. A.7 to get:

$$\int_{-\infty}^{g^*} g\,\hat{G}(g)\,dg = -\frac{\sigma}{\sqrt{2\pi}}\,\exp\left[-w^*\right] + \mu\,\Phi(v^*) \tag{A.10}$$

which can be written in terms of the standard normal PDF as:

$$\begin{aligned}\int_{-\infty}^{g^*} g\,\hat{G}(g)\,dg &= -\sigma\,\phi\,(v^*) + \mu\,\Phi(v^*) \\ &= -\sigma\,\phi\left(\frac{g^*-\mu}{\sigma}\right) + \mu\,\Phi\left(\frac{g^*-\mu}{\sigma}\right)\end{aligned} \tag{A.11}$$

Finally, substitute Eq. A.11 into Eq. A.3 to get:

$$EI = g^*\,\Phi\left(\frac{g^*-\mu}{\sigma}\right) + \sigma\,\phi\left(\frac{g^*-\mu}{\sigma}\right) - \mu\,\Phi\left(\frac{g^*-\mu}{\sigma}\right) \tag{A.12}$$

which can be rearranged to give:

$$EI = (g^*-\mu)\,\Phi\left(\frac{g^*-\mu}{\sigma}\right) + \sigma\,\phi\left(\frac{g^*-\mu}{\sigma}\right) \tag{A.13}$$

A.2 Expected Feasibility Function

Using the EGO concept to find points that lie on the limit state requires finding points that satisfy the equality constraint $G(\mathbf{x}) = \bar{z}$. To locate these points, the exected feasibility function is introduced:

$$EF = E\left[\epsilon - \min\left(|\hat{G}(g) - \bar{z}|,\, \epsilon\right)\right] \tag{A.14}$$

where $\epsilon = \alpha\sigma$ and α is some positive constant. It is clear that the minimization term follows the relationship:

$$\begin{aligned}\min\left(|\hat{G}(g) - \bar{z}|,\, \epsilon\right) &= \epsilon && \text{for } -\infty < \hat{G}(g) < \bar{z} - \epsilon \\ &= |\hat{G}(g) - \bar{z}| && \text{for } \bar{z} - \epsilon < \hat{G}(g) < \bar{z} + \epsilon \\ &= \epsilon && \text{for } \bar{z} + \epsilon < \hat{G}(g) < \infty\end{aligned} \tag{A.15}$$

The expected value can then be calculated by integrating over the interval where the function is nonzero:

$$\begin{aligned} EF &= \int_{z^-}^{z+} \left[\epsilon - |g - \bar{z}|\right] \hat{G}(g)\, dg \\ &= \epsilon \int_{z^-}^{z+} \hat{G}(g)\, dg - \int_{z^-}^{z+} |g - \bar{z}|\, \hat{G}(g)\, dg \end{aligned} \tag{A.16}$$

where $z^- = \bar{z} - \epsilon$ and $z^+ = \bar{z} + \epsilon$. Concentrating on the first term:

$$\begin{aligned} \epsilon \int_{z^-}^{z+} \hat{G}(g)\, dg &= \epsilon \int_{-\infty}^{z^+} \hat{G}(g)\, dg - \epsilon \int_{-\infty}^{z^-} \hat{G}(g)\, dg \\ &= \epsilon \left[\Phi\left(\frac{z^+ - \mu}{\sigma}\right) - \Phi\left(\frac{z^- - \mu}{\sigma}\right)\right] \end{aligned} \tag{A.17}$$

Let X represent the second term, $X = \int_{z^-}^{z+} |g - \bar{z}|\, \hat{G}(g)\, dg$, which becomes:

$$\begin{aligned} X &= \int_{z^-}^{\bar{z}} (\bar{z} - g)\ \hat{G}(g)\, dg \int_{\bar{z}}^{z^+} (g - \bar{z})\ \hat{G}(g)\, dg \\ &= \bar{z} \int_{z^-}^{\bar{z}} \hat{G}(g)\, dg - \int_{z^-}^{\bar{z}} g\, \hat{G}(g)\, dg + \int_{\bar{z}}^{z^+} g\, \hat{G}(g)\, dg - \bar{z} \int_{\bar{z}}^{z^+} \hat{G}(g)\, dg \\ &= \bar{z} \int_{-\infty}^{\bar{z}} \hat{G}(g)\, dg - \bar{z} \int_{-\infty}^{z^-} \hat{G}(g)\, dg - \int_{-\infty}^{\bar{z}} g\, \hat{G}(g)\, dg + \int_{-\infty}^{z^-} g\, \hat{G}(g)\, dg \\ &\quad + \int_{-\infty}^{z^+} g\, \hat{G}(g)\, dg - \int_{-\infty}^{\bar{z}} g\, \hat{G}(g)\, dg - \bar{z} \int_{-\infty}^{z^+} \hat{G}(g)\, dg + \bar{z} \int_{-\infty}^{\bar{z}} \hat{G}(g)\, dg \\ &= \bar{z}\, \Phi\left(\frac{\bar{z} - \mu}{\sigma}\right) - \bar{z}\, \Phi\left(\frac{z^- - \mu}{\sigma}\right) - \bar{z}\, \Phi\left(\frac{z^+ - \mu}{\sigma}\right) + \bar{z}\, \Phi\left(\frac{\bar{z} - \mu}{\sigma}\right) \\ &\quad - \int_{-\infty}^{\bar{z}} g\, \hat{G}(g)\, dg + \int_{-\infty}^{z^-} g\, \hat{G}(g)\, dg + \int_{-\infty}^{z^+} g\, \hat{G}(g)\, dg - \int_{-\infty}^{\bar{z}} g\, \hat{G}(g)\, dg \end{aligned} \tag{A.18}$$

From Eq. A.11 the following are known:

$$\begin{aligned} \int_{-\infty}^{\bar{z}} g\, \hat{G}(g)\, dg &= -\sigma\, \phi\left(\frac{\bar{z} - \mu}{\sigma}\right) + \mu\, \Phi\left(\frac{\bar{z} - \mu}{\sigma}\right) \\ \int_{-\infty}^{z^-} g\, \hat{G}(g)\, dg &= -\sigma\, \phi\left(\frac{z^- - \mu}{\sigma}\right) + \mu\, \Phi\left(\frac{z^- - \mu}{\sigma}\right) \\ \int_{-\infty}^{z^+} g\, \hat{G}(g)\, dg &= -\sigma\, \phi\left(\frac{z^+ - \mu}{\sigma}\right) + \mu\, \Phi\left(\frac{z^+ - \mu}{\sigma}\right) \end{aligned} \tag{A.19}$$

Substituting these in and combining terms:

$$\begin{aligned}\int_{z^-}^{z+} |g-\bar{z}|\,\hat{G}(g)\,dg = (\bar{z}-\mu)\left[2\,\Phi\left(\frac{\bar{z}-\mu}{\sigma}\right)-\Phi\left(\frac{z^- -\mu}{\sigma}\right)-\Phi\left(\frac{z^+ -\mu}{\sigma}\right)\right] \\ +\sigma\left[2\,\phi\left(\frac{\bar{z}-\mu}{\sigma}\right)-\phi\left(\frac{z^- -\mu}{\sigma}\right)-\phi\left(\frac{z^+ -\mu}{\sigma}\right)\right]\end{aligned} \tag{A.20}$$

Now, substituting the first and second term back into Eq. A.16, gives:

$$\begin{aligned}EF = (\mu-\bar{z})\left[2\,\Phi\left(\frac{\bar{z}-\mu}{\sigma}\right)-\Phi\left(\frac{z^- -\mu}{\sigma}\right)-\Phi\left(\frac{z^+ -\mu}{\sigma}\right)\right] \\ -\sigma\left[2\,\phi\left(\frac{\bar{z}-\mu}{\sigma}\right)-\phi\left(\frac{z^- -\mu}{\sigma}\right)-\phi\left(\frac{z^+ -\mu}{\sigma}\right)\right] \\ +\epsilon\left[\Phi\left(\frac{z^+ -\mu}{\sigma}\right)-\Phi\left(\frac{z^- -\mu}{\sigma}\right)\right]\end{aligned} \tag{A.21}$$

A.3 Expected Violation Function for Equality Constraints

The expected violation function for inequality constraints is similar to the expected improvement function, and the derivation follows closely to that shown in Section A.1. The case of equality constraints is presented here. The expected violation function for an equality constraint is defined as:

$$EV_h = E\left[|\hat{G}(g)-\bar{h}|\right] \tag{A.22}$$

This expected value can be calculated through the integral:

$$\begin{aligned}EV_h &= \int_{-\infty}^{\infty} |g-\bar{h}|\,\hat{G}(g)\,dg \\ &= \int_{-\infty}^{\bar{h}} (\bar{h}-g)\,\hat{G}(g)\,dg + \int_{\bar{h}}^{\infty} (g-\bar{h})\,\hat{G}(g)\,dg \\ &= \bar{h}\int_{-\infty}^{\bar{h}} \hat{G}(g)\,dg - \int_{-\infty}^{\bar{h}} g\,\hat{G}(g)\,dg + \int_{\bar{h}}^{\infty} g\,\hat{G}(g)\,dg - \bar{h}\int_{\bar{h}}^{\infty} \hat{G}(g)\,dg \\ &= \bar{h}\int_{-\infty}^{\bar{h}} \hat{G}(g)\,dg - \bar{h}\left[1-\int_{-\infty}^{\bar{h}} \hat{G}(g)\,dg\right] \\ &\quad - \int_{-\infty}^{\bar{h}} g\,\hat{G}(g)\,dg + \left[\int_{-\infty}^{\infty} g\,\hat{G}(g)\,dg - \int_{-\infty}^{\bar{h}} g\,\hat{G}(g)\,dg\right]\end{aligned} \tag{A.23}$$

where $\int_{-\infty}^{\infty} g\,\hat{G}(g)\,dg \equiv \mu$, and from the previous derivations, it is known that:

$$\int_{-\infty}^{\bar{h}} g\,\hat{G}(g)\,dg = -\sigma\,\phi\left(\frac{\bar{h}-\mu}{\sigma}\right) + \mu\,\Phi\left(\frac{\bar{h}-\mu}{\sigma}\right) \tag{A.24}$$

Substituting these into the previous equation and combining terms, gives:

$$EV_h = \left(\mu - \bar{h}\right)\left[1 - 2\Phi\left(\frac{\bar{h}-\mu}{\sigma}\right)\right] + 2\sigma\phi\left(\frac{\bar{h}-\mu}{\sigma}\right) \tag{A.25}$$

REFERENCES

[1] Adams, B.M., Eldred, M.S., Wittwer, J., and Massad, J., Reliability-Based Design Optimization for Shape Design of Compliant Micro-Electro-Mechanical Systems, *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA, Sept. 6-8, 2006.

[2] Adams, B.M., Bichon, B.J., Carnes, B., Copps, K.D., Eldred, M.S., Hopkins, M.M., Neckels, D.C., Notz, P.K., Subia, S.R., and Wittwer, J.W., Solution-Verified Reliability Analysis and Design of Bistable MEMS Using Error Estimation and Adaptivitiy, Sandia Technical Report SAND2006-6286, October 2006.

[3] Agarwal, H., Renaud, J.E., Lee, J.C., and Watson, L.T., A Unilevel Method for Reliability Based Design Optimization, paper AIAA-2004-2029 in *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, April 19-22, 2004.

[4] Allen, M. and Maute, K., Reliability-based design optimization of aeroelastic structures, *Structural and Multidisciplinary Optimization*, Vol. 27, 2004, pp. 228-242.

[5] Ambartzumian, R., Der Kiureghian, A., Ohanian, V., and Sukiasian, H., Multinormal Probability by Sequential Conditioned Importance Sampling, *Advances in Safety and Reliability, Proc. ESREL '97*, June 17-20, Lisbon, Vol. 2, pp. 1261-1268.

[6] Ananthasuresh, G.K., Kota, S., and Gianchandani, Y., A Methodical Approach to the Design of Compliant Micromechanisms, *Proc. IEEE Solid-State Sensor and Actuator Workshop*, Hilton Head Island, SC, 1994, pp. 189-192.

[7] Audet, C., Dennis, J.E., Moore, D.W., Booker, A., and Frank, P.D., A Surrogate-Model-Based Method for Constrained Optimization, *Proceedings of the 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization* paper AIAA-2000-4891, 2000.

[8] Björkman, M., and Holström, K., Global Optimization of Costly Nonconvex Functions using Radial Basis Functions, *Optimization and Engineering*, Vol. 1, 2000, pp. 373-397.

[9] Box, G. E. P. and Wilson, K.B., On the Experimental Attainment of Optimum Conditions (with discussion). *Journal of the Royal Statistical Society*, Series B, Vol. 13, 1951, pp. 145.

[10] Box, G.E.P. and Cox, D.R., An Analysis of Transformations, *Journal of the Royal Statistical Society*, Series B, Vol. 26, 1964, pp. 211-252.

[11] Breitung, K., Asymptotic Approximation for Multinormal Integrals, *Journal of Engineering Mechanics, ASCE*, Vol. 110, No. 3, 1984, pp. 357-366.

[12] Broyden, C.G., The Convergence of a Class of Double-rank Minimization Algorithms, *Journal of the Institute of Mathematics and its Applications*, Vol. 6, 1970, pp. 76-90.

[13] Buhmann, M.D., *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, Cambridge, UK, 2003.

[14] Chen, X. and Lind, N.C., Fast Probability Integration by Three-Parameter Normal Tail Approximation, *Structural Safety*, Vol. 1, 1983, pp. 269-276.

[15] Chiralaksanakul, A., and Mahadevan, S., First-Order Approximation Methods in Reliability-Based Design Optimization, *J. of Mech. Design*, Vol. 127, 2005.

[16] Collier, C., Yarrington, P., and Pickenheim, M., The Hypersizing Method for Structures, presented at *NAFEMS World Congress '99*, Newport, RI, Apr. 25-28, 1999.

[17] Conn, A.R., Gould, N.I.M., and Toint, P.L., *Trust-Region Methods*, MPS-SIAM Series on Optimization, 2000.

[18] Cressie, N.A.C., *Statistics for Spatial Data*, revised edition, 1993 (Wiley: New York).

[19] Der Kiureghian, A. and Liu, P.L., Structural Reliability Under Incomplete Probability Information, *Journal of Engineering Mechanics, ASCE*, Vol. 112, No. 1, 1986, pp. 85-104.

[20] Dey, A. and Mahadevan, S., Ductile Structural System Reliability Analysis using Adaptive Importance Sampling, *Structural Safety*, Vol. 20, 1998, pp. 137-154.

[21] Ditlevsen, O., Narrow Reliability Bounds for Structural Systems, *Journal of Structural Mechanics*, Vol. 7, No. 4, 1979, pp. 453-472.

[22] Du, X. and Chen, W., Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design, *Journal of Mechanical Design*, Vol. 126, 2004, pp.225-233.

[23] Dunnett, C.W. and Sobel, M., A Bivariate Generalization of Student's t-Distribution, with Tables for Certain Special Cases, *Biometrika*, Vol. 41, 1954, pp. 153-169.

[24] Eldred, M.S. and Bichon, B.J., New Second-Order Formulations for Reliability Analysis and Design, *AIAA Journal*, in preparation.

[25] Eldred, M.S. and Dunlavy, D.M., Formulations for Surrogate-Based Optimization with Data Fit, Multifidelity, and Reduced-Order Models, paper AIAA-2006-7117 in *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA, Sept. 6-8, 2006.

[26] Eldred, M.S., Agarwal, H., Perez, V.M., Wojtkiewicz, S.F., Jr., and Renaud, J.E., Investigation of Reliability Method Formulations in DAKOTA/UQ, (to appear) *Structure & Infrastructure Engineering: Maintenance, Management, Life-Cycle Design & Performance*, Taylor & Francis Group.

[27] Eldred, M.S., Giunta, A.A., Wojtkiewicz, S.F., Jr., and Trucano, T.G., Formulations for Surrogate-Based Optimization Under Uncertainty, paper AIAA-2002-5585

in *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, Sept. 4-6, 2002.

[28] Eldred, M.S., Giunta, A.A., Brown, S.L., Adams, B.M., Dunlavy, D.M., Eddy, J.P., Gay, D.M., Griffin, J.D., Hart, W.E., Hough, P.D., Kolda, T.G., Martinez-Canales, M.L., Swiler, L.P., Watson, J.-P., and Williams, P.J., DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 4.0 User's Manual. Sandia Technical Report SAND2006-6337, Revised October 2006, Sandia National Laboratories, Albuquerque, NM.

[29] Eldred, M.S., Bichon, B.J., Adams, B., and Mahadevan, S., Overview of Reliability Analysis and Design Capabilities in DAKOTA with Application to Shape Optimization of MEMS," *Structures and Infrastructures Series, Volume 1: Structural Design Optimization Considering Uncertainties*, edited by Tsompanakis, Y., Lagaros, N.D., and Papadrakakis, M., CRC Press/Balkema, Leiden, The Netherlands, 2008.

[30] Eldred, M.S., Webster, C.G., and Constantine, P., Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos, paper AIAA-2008-1892 in *Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Schaumburg, IL, April 7-10, 2008.

[31] Fadel, G.M., Riley, M.F., and Barthelemy, J.-F.M., Two Point Exponential Approximation Method for Structural Optimization, *Structural Optimization*, Vol. 2, No. 2, 1990, pp. 117-124.

[32] Fletcher, R., A New Approach to Variable Metric Algorithms, *Computer Journal*, Vol. 13, 1970, pp. 317-322.

[33] Friedman, J.H., Multivariate Adaptive Regression Splines, *Annals of Statistics*, Vol. 19, No. 1, 1991, pp. 167.

[34] Gablonsky, J.M., An Implementation of the DIRECT Algorithm, Technical Report CRSC-TR98-29, Center for Research in Scientific Computation, North Carolina State University, August, 1998.

[35] Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H., User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming, System Optimization Laboratory, Technical Report SOL 86-1, Revised July 1998, Stanford University, Stanford, CA.

[36] Giunta, A.A. and Eldred, M.S., Implementation of a Trust Region Model Management Strategy in the DAKOTA Optimization Toolkit, paper AIAA-2000-4935 in *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, September 6-8, 2000.

[37] Giunta, A.A., McFarland, J.M., Swiler, L.P., Eldred, M.S., The Promise and Peril of Uncertainty Quantification via Response Surface Approximations, *Structure and Infrastructure Engineering*, Vol. 2, No. 3, 2006, pp. 175-189.

[38] Goldfarb, D., A Family of Variable Metric Updates Derived by Variational Means, *Mathematics of Computation*, Vol. 24, 1970, pp. 23-26.

[39] Haldar, A. and Mahadevan, S., *Probability, Reliability, and Statistical Methods in Engineering Design*, 2000 (Wiley: New York).

[40] Harbitz, A., An Efficient Sampling Method for Probability of Failure Calculation, *Structural Safety*, Vol. 3, No. 2, 1986, pp. 109-115.

[41] Hohenbichler, M. and Rackwitz, R., Sensitivity and Importance Measures in Structural Reliability, *Civil Engineering Systems*, Vol. 3, 1986, pp. 203-209.

[42] Hohenbichler, M. and Rackwitz, R., First-Order Concepts in Systems Reliability, *Structural Safety*, Vol. 1, 1987, pp. 177-188.

[43] Hohenbichler, M. and Rackwitz, R., Improvement of Second-Order Reliability Estimates by Importance Sampling, *Journal of Engineering Mechanics, ASCE*, Vol. 114, No. 12, 1988, pp. 2195-2199.

[44] Hong, H.P., Simple Approximations for Improving Second-Order Reliability Estimates, *Journal of Engineering Mechanics, ASCE*, Vol. 125, No. 5, 1999, pp. 592-595.

[45] Huang, D., Allen, T.T., Notz, W.I., and Zeng, N., Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models, *Journal of Global Optimization*, Vol. 34, 2006, pp. 441-466.

[46] Jensen, B.D., Parkinson, M.B., Kurabayashi, K., Howell, L.L., and Baker, M.S., Design Optimization of a Fully-Compliant Bistable Micromechanism, *Proc. 2001 ASME Intl. Mech. Eng. Congress and Exposition*, New York, NY, November 11-16, 2001.

[47] Jones, D., Shonlau, M., and Welch, W., Efficient Global Optimization of Expensive Black-Box Functions, *INFORMS J. Comp.*, Vol. 12, 1998, pp. 272.283

[48] Karamchandani, A. and Cornell, C.A., Sensitivity estimation within first and second order reliability methods, *Structural Safety*, Vol. 11, 1992, pp. 95-107.

[49] Kass, R.E. and Wasserman, L., A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion, *Journal of the American Statistical Association*, Vol. 90, 1995, pp. 928-934.

[50] Kemeny, D.C., Howell, L.L., and Magleby, S.P., Using Compliant Mechanisms to Improve Manufacturability in MEMS, *Proc. 2002 ASME DETC*, No. DETC2002/DFM-34178, 2002.

[51] Kuschel, N. and Rackwitz, R., Two basic problems in reliability-based structural optimization. *Math. Method Operations Research*, Vol. 46, 1997, pp. 309-333.

[52] Lee, J.O., Yang, Y.O., and Ruy, W.S., A comparative study on reliability index and target performance based probabilistic structural design optimization, *Computers and Structures*, Vol. 80, 2002, pp. 257-269.

[53] Lee, P., *Bayesian Statistics, an Introduction*, 2004 (Oxford University Press, Inc.: New York).

[54] Liang, J., Mourelatos, Z.P., and Tu, J., A Single-Loop Method for Reliability-Based Design Optimization, *Proc. 2004 ASME Design Eng. Tech. Conf.*, Paper DETC2004/DAC-57255.

[55] Mahadevan, S. and P. Shi, Multiple Linearization Method for Nonlinear Reliability Analysis, *Journal of Engineering Mechanics, ASCE*, Vol. 127, No. 11, 2001, pp. 1165-1173.

[56] Marhadi, K., Venkataraman, S., and Pai, S., Quantifying Uncertainty in Statistical Distribution of Small Sample Data Using Bayesian Inference of Unbounded Johnson Distribution, *Proc. 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Schaumburg, IL, April 7-10, 2008.

[57] McDonald, M. and Mahadevan, S., Design Optimization with System-Level Reliability Constraints, *Journal of Mechanical Design*, Vol. 130, February, 2008.

[58] McFarland, J.M., *Uncertainty Analysis for Computer Simulations through Validation and Calibration*, Ph.D. thesis, Vanderbilt University, 2008.

[59] Meza, J.C., OPT++: An Object-Oriented Class Library for Nonlinear Optimization, Sandia Technical Report SAND94-8225, Sandia National Laboratories, Livermore, CA, March 1994.

[60] Nocedal, J. and Wright, S.J., *Numerical Optimization*, Springer, New York, 1999.

[61] Pandey, M.D., An Effective Approximation to Evaluate Multinormal Integrals, *Structural Safety*, Vol. 20, 1998, pp. 51-67.

[62] Qiu, J., and Slocum, A.H., A Curved-Beam Bistable Mechanism, *J. Microelectromechanical Sys.*, Vol. 13, No. 2, 2004, pp. 137-146.

[63] Qu, X. and Haftka, R.T., Reliability-Based Design Optimization Using Probabilistic Sufficiency Factor, paper AIAA-2003-1657 in *Proc. 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Norfolk, VA, April 2003.

[64] Rackwitz, R., Optimization and risk acceptability based on the Life Quality Index, *Structural Safety*, Vol. 24, 2002, pp. 297-331.

[65] Rackwitz, R. and Fiessler, B., Structural Reliability under Combined Random Load Sequences, *Computers and Structures*, Vol. 9, 1978, pp. 489-494.

[66] Ranjan, P., Bingham, D., and Michailidis, G., Sequential Experiment Design for Contour Estimation from Complex Computer Codes, *Technometrics*, to appear, 2007.

[67] Rosenblatt, M., Remarks on a Multivariate Transformation, *Ann. Math. Stat.*, Vol. 23, No. 3, 1952, pp. 470-472.

[68] Sacks, J., Schiller, S.B., and Welch, W., Design for Computer Experiments, *Technometrics*, Vol. 31, 1989, pp. 41-47.

[69] Sasena, M.J., *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, PhD Thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 2002.

[70] Schonlau, M. *Computer Experiments and Global Optimization*, PhD Thesis, University of Waterloo, Waterloo, Canada, 1997.

[71] Shanno, D.F., Conditioning of Quasi-Newton Methods for Function Minimization, *Mathematics of Computation*, Vol. 24, 1970, pp. 647-656.

[72] Silverman, B.W., *Density Estimation for Statistics and Data Analysis*, Chapman & Hall/CRC, New York, 1986.

[73] Simpson, T.W., Booker, A.J., Ghosh, D., Giunta, A.A., Koch, P.N., and Yang, R.-J., Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion, *Structural and Multidisciplinary Optimization*, Vol. 27, No. 5, pp. 302-313, 2004.

[74] Sirimamilla, R., Venkataraman, S., and Pai, S., Incorporating Data Uncertainty in Reliability-Based Design Optimization Using Inverse Reliability Measures, *Proc. 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Schaumburg, IL, April 7-10, 2008.

[75] Smith, N. and Mahadevan, S., Integrating System-Level and Component-Level Designs Under Uncertainty, *Journal of Spacecraft and Rockets*, Vol. 42, No. 4, 2005, pp. 752-760.

[76] Sues, R., Aminpour, M. and Shin, Y., Reliability-Based Multidisciplinary Optimization for Aerospace Systems, paper AIAA-2001-1521 in *Proc. 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Seattle, WA, April 16-19, 2001.

[77] Thacker, B.H., Riha, D.S., Millwater, H.R., and Enright, M.P., Errors and uncertainties in probabilistic engineering analysis, paper AIAA-2001-1239 in *Proc. of the 42nd Structures, Structural Dynamics, and Materials Conference*, Seattle, WA, April 16-19, 2001.

[78] Tu, J., Choi, K.K., and Park, Y.H., A New Study on Reliability-Based Design Optimization, *Journal of Mechanical Design*, Vol. 121, 1999, pp. 557-564.

[79] Tvedt, L., Distribution of Quadratic Forms in Normal Space - Application to Structural Reliability, *Journal of Engineering Mechanics*, Vol. 116, No. 6, 1990, pp. 1183-1197.

[80] Vanderplaats, G.N., *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Hill, New York, 1984.

[81] Venkataraman, S., Sirimamilla, R., Mahadevan, S., Nagpal, V., Strack, B., and Pai, S., Calculating Confidence Bounds for Reliability Index to Quantify Effect of Distribution Parameter Uncertainty, *Proc. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Waikiki, HI, April 23-26, 2007.

[82] Wang, L. and Grandhi, R.V., Efficient Safety Index Calculation for Structural Reliability Analysis, *Comput. Struct.*, Vol. 52, No. 1, 1994, pp. 103-111.

[83] Wang, L., Beeson, D., Akkaram, S., Wiggs, G., Gaussian Process Meta-models for Efficient Probabilistic Design in Complex Engineering Design Spaces, *Proc. of ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, Long Beach, CA, September 24-28, 2005.

[84] Wasserman, L., Bayesian Model Selection and Model Averaging, *Journal of Mathematical Psychology*, Vol. 44, No. 1, 2000.

[85] Wittwer, J.W., Baker, M.S., and Howell, L.L., Robust Design and Model Validation of Nonlinear Compliant Micromechanisms, *Journal of Microelectromechanical Systems*, Vol. 15, No. 1, 2006.

[86] Wojtkiewicz, S.F., Jr., Eldred, M.S., Field, R.V., Jr., Urbina, A., and Red-Horse, J.R., A Toolkit For Uncertainty Quantification In Large Computational Engineering Models, *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, paper AIAA-2001-1455, Seattle, WA, April 16-19, 2001.

[87] Wu, Y.-T., Millwater, H.R., and Cruse, T.A., Advanced Probabilistic Structural Analysis Method for Implicit Performance Functions, *AIAA J.*, Vol. 28, No. 9, 1990, pp. 1663-1669.

[88] Wu, Y.-T., Shin, Y., Sues, R., and Cesare, M., Safety-Factor Based Approach for Probability-Based Design Optimization, paper AIAA-2001-1522 in *Proceedings of*

the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Seattle, WA, April 16-19, 2001.

[89] Wu, Y.-T., An Adaptive Importance Sampling Method for Structural System Reliability Analysis, Reliability Technology 1992, In T.A. Cruse (Editor), *ASME Winter Annual Meeting*, Vol. AD-28, Anaheim, CA, pp. 217-231.

[90] Wu, Y.-T., Computational Methods for Efficient Structural Reliability and Reliability Sensitivity Analysis, *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1717-1723.

[91] Wu, Y.-T. and Wirsching, P.H., A New Algorithm for Structural Reliability Estimation, *Journal of Engineering Mechanics, ASCE*, Vol. 113, 1987, pp. 1319-1336.

[92] Xiu, D. and Karniadakis, G. M., The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations, *SIAM Journal on Scientific Computing*, Vol. 24, No. 2, 2002, pp. 619644.

[93] Xu, S., and Grandhi, R.V., Effective Two-Point Function Approximation for Design Optimization, *AIAA Journal*, Vol. 36, No. 12, 1998, pp. 2269-2275.

[94] Zou, T., Mourelatos, Z., Mahadevan, S., and Tu, J., Reliability Analysis of Automotive Body-Door Subsystem, *Reliability Engineering and System Safety*, Vol. 78, 2002, pp. 315-324.

[95] Zou, T., Mourelatos, Z., Mahadevan, S., and Tu, J., An Indicator Response Surface Method for Simulation-Based Reliability Analysis, *Journal of Mechanical Design, ASME*, Vol. 130, 2008.

[96] Zou, T., Mahadevan, S., and Rebba, R., Computational Efficiency in Reliability-Based Optimization, *Proceedings of the 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, Albuquerque, NM, July 26-28, 2004.

[97] Zou, T. and Mahadevan, S., A Direct Decoupling Approach for Efficient Reliability-Based Design Optimization, *Structural and Multidisciplinary Optimization* Vol. 31, 2006, pp. 190-200.

[98] Zou, T. and Mahadevan, S., Versatile Formulation for Multiobjective Reliability-Based Design Optimization, *Journal of Mechanical Design, ASME*, Vol. 128, 2006, pp. 1217-1226.