Fall 2011

Quantifying model uncertainty using measurement uncertainty standards

Harsheel Rajesh Shah

Department:

QUANTIFYING MODEL UNCERTAINTY USING MEASUREMENT UNCERTAINTY STANDARDS

by

HARSHEEL RAJESH SHAH

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

2011

Approved by

Dr. Xiaoping Du, Advisor
Dr. Serhat Hosder
Dr. K.Chandrashekhara

ABSTRACT

Model uncertainty quantification is mainly concerned with the problem of determining whether the observed data is consistent with the model prediction. In real world, there is always a disagreement between a simulation model prediction and the reality that the model intends to represent. Our increased dependence on computer models emphasizes on model uncertainty which is present due to uncertainties in model parameters, lack of appropriate knowledge, assumptions and simplification of processes. In addition, when models predict multi-variate data, the experimental observation and model predictions are highly correlated. Thus, quantifying the uncertainty has a basic requirement of comparison between observation and prediction. The comparison is costly on the observation side and computationally intensive on the other. The alternative approach presented in this thesis for model uncertainty quatification addresses the aforementioned problems. With the new methodology, the experiments performed according to measurement uncertainty standards will provide the experimental results in terms of expanded uncertainty. Thus, the experimental results for both model input and output will be expressed as intervals. Furthermore, interval predictions are procured from the simulation model by using the experimental results of input intervals only. The model uncertainty will then be quantified by the difference between experimental result for output interval and model prediction interval. The new methodology is easy to implement as the standards of measurement uncertainty are used which serve as a common framework for model builders and experimenters.

ACKNOWLEDGMENT

I would like to express my heartfelt appreciation for all those people who have contributed to the production of this thesis. I owe my deepest gratitude to my advisor Dr. Xiaoping Du and thank him for the opportunity to conduct this research project under his supervision and guidance. His expertise in this research topic helped me to learn better. His constant encouragement has set high research standards and guided me to achieve them successfully. His constructive criticism at all stages was welcomed and thought-provoking.

I would like to acknowledge Dr. Serhat Hosder for sharing his meticulous research insights. The valuable discussions with him improved my knowledge and work on optimization and approxiamtion techniques which were fruitful in making this research possible. Furthermore, I would like to thank Dr. K Chandrashekhara who has generously given his time and expertise pertaining not only to this research but also my future prospects.

Missouri University of Science and Technology and the Aerospace Engineering Department have played a vital role in moulding me as a competitive engineer by providing me with this opportunity to use their advanced facilities for my research. Also, I need to thank Dr. Nishant Kumar, Ben Bettis, Han Douru, Praveen Srinivas and Ravi Gowrishankar for all of their support while working together in the research laboratory. The elongated discussions with them made sure that I had no questions unanswered regarding any topic.

Finally, I want to thank my family, to whom this dissertation is dedicated, for their endless love and belief in me. Without their moral support, I would not have managed to complete my academics at a place which is a home away from home.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

NOMENCLATURE

Symbol	Description
y	True value of a model
$\tilde{y}$	System average response variable
ϵ^e	Experimental error
y^e	Experimental output
y^m	Model prediction
ϵ^m	Model error
U^m	Model uncertainty
$\vec{x}$	Input random variable vector
$\vec{w}$	Model parameter vector
h	Minimal height required (m)
r	Radius of loop trajectory (m)
g	Gravitational force (N/m^2)
v	Velocity of the body (N/m^2)
T	Temperature in the body $(^\circ C)$
i	Initial conditions (subscript)
T_i	Inital temperature $(^\circ C)$
L	Thickness of slab (m)
q	Heat flux (W/m^2)
k	Thermal conductivity $(W/m^\circ C)$
x	Desired distance from the front face of slab (m)
ρC_p	Volumetric heat capacity (kg/m^3)

1. INTRODUCTION

1.1. IMPORTANCE OF MODEL UNCERTAINTY QUANTIFICATION

Simulation models have become a vital part of many complex real world problems due to the fact that they allow for quick and inexpensive predictions. However, there is always a question regarding the accuarcy of the prediction model as to how close it gets to its intended application. Hemez et al. [4] were of the view that the accuracy must be verified due to the assumptions involved in the steps of idealization, discretization and modeling. In a broad sense, if there exists a difference between the model prediction and observations, the model is termed as *uncertain*. These observations are motivated by Popper's analysis of scientific theories [5]. Most of the methodologies used to determine the accuracy of the model or quantify *model uncertainty* are either expensive or computationally intense. The focus of the present work is to simplify the model uncertainty quantification (UQ) process and overcome its hindrances.

Recently, there has been a great deal of research activity in the field of model uncertainty where a model is considered to be an abstraction and interpretation of reality [6]. It can arise due to several reasons like lack of proper knowledge, assumtions during modeling, uncertainty in the input parametrs or the uncertainty in the model itself. Analysts have focused on model input parameters as the principal source of uncertainty, as seen in traditional reliability based design [7, 8, 9] and robust design [10, 11, 12].

Model uncertainty can be related to model validation [13, 14] which is a broad term and hence it should be defined for every application as it can have different meanings. For example, aerospace engineers consider a model valid only if it correctly and fully represents the process being modeled. While for some model builders it is fine if the model is "good enough" for its application. If the validity of the model is

questionable, it cannot be used for analysis with sufficient confidence. Hence, there is a need to detect the uncertainty present in the model and refine it [15]. As mentioned earlier, the uncertainty can be obtained by comparing the model prediction from simulation and experimentation. If they agree to a certain extent, then one should not worry about the rejection of the model. In some cases, even after the model is validated, its accuracy is desired as it helps the designers or model builders to make critical decisions [16].

1.2. OBJECTIVE

The objective of this research is to devise an alternative approach so as to quantify model uncertainty using ASME measurement uncertainty standards. The quantification of model uncertainty involves the similar procedure and information as the model validation because both need the comparison between the model prediction and observation. In many model validation literatures, it is not clear if a model builder has the capability of performing lab experiments and then producing experimental results that are needed. In reality, the model builder may not be able to control the lab work due to various reasons.

One alternative is to outsource the experimental work to professional laboratories which have the facility and infrastructure to perform these experiments. During this time, there would be many communications between the model builder and experimenter. Following the experimental standards is a common practice in professional laboratories.

As a result, the experiments would be conducted and the experimental results would be reported in accordance with certain experimental standards. Then it is up to the model builder to use the experimental results to quantify the model uncertainty. Model users can then use the model predictions, along with the quantified uncertainty, to make more reliable decisions. In this work, a situation is addressed where the

experiments are outsourced and the results are reported with ASME standards of measurement uncertainty.

1.3. LITERATURE REVIEW ON MODEL UQ

The following literature review is regarding model validation strategies and different methods used to quantify uncertainty in a particular model which can thereby be used to validate the model.

Several researchers have strived hard in order to devise a methodology to validate a model [17]. Sandia National Laboratories is trying to evolve uncertainty quantification technologies through development activities in computational engineering and its experience in model validation [18]. There have been many revelations in this field of study as they have constantly tried to reduce the complexity of the problem. Dowding et al. [1], provided a good platform in order to exercise different model validation procedures by formulating a thermal challenge problem. The thermal problem is based on one dimensional linear heat conduction in a solid slab with heat flux boundary conditions. They presented a series of experimental data to be compared to the model prediction relative to a regulatory requirement.

The regulatory requirement is in terms of probability that the surface temperature at a specific location does not exceed a specified temperature. In order to validate the model, three experimental activities are required namely, material characterization (thermal properties are characterized), ensemble validation (experiments on a range of values for deterministic variables), and accredition validation (limited number of experiments on conditions that are more representative of the original model). Regulatory compliance is then assessed based on the results of the experimental activities. Thus, the proposed method relies a lot on experimentation which sometimes creates a problem if we do not have the required facilities.

Chen et al. [19] extended the study of model validation towards the propagation of uncertainty in model predictions. According to them, as deterministic simulations for model validation do not consider uncertainty at all, there was a need to include relevant sources of uncertainties in the validation process. Many statistical inference techniques such as chi-square test on residuals were considered cumbersome due to numerous evaluations of the model and experiments. They proposed response surface methodology to create a metamodel for the original model thereby reducing the computational effort. Their example problems consisted of two finite element models for simulating sheet metal forming. Roy et al. [20] contributed to model validation by introducing statistical methods like interval based hypothesis testing explained through the example of natural frequency of a cantilever beam.

Hills and Dowding [21] presented a multivariate approach to the thermal challenge problem that accounted for model parameter uncertainty and correlation between multiple measurement differences. They proposed a method for accounting for model parameter uncertainty through first order sensitivity analysis for validation tests. Furthermore, they implemented first order sensitivity analysis and Monte Carlo analysis for the regulatory prediction. Monte Carlo analysis was found to be more accurate but computationally expensive at the same time. Experimental and predicted CDF's from the correlation structure of the model were used to validate the model and they indicated that the regulatory criterion was not met.

Brandyberry [22] proposed a surrogate model clustering technique for the thermal challenge problem that allows propagation of uncertainties in the model parameters using less number of model evaluations. Different statistical methods like t-test and hypothesis testing were applied to the experimental data to arrive at conclusions. The author assumed distributions for the random variables based on the normal probability plots and the scatter plots. In addition to this, Latin hypercube sampling was used to propagate uncertainty through the model with less number of simulations.

Ferson et al. [23] has adressed the concept of model validation in a different way. They proposed a risk analytic approach for the thermal challenge problem by assuming normal distributions for the random variables based on experimental values. Monte Carlo simulation was implemented to find the probabilty of the regulatory requirement. In further analysis, they investigated the temperature dependence of random variables which is basically the system response. A regression model using the least squares was developed for the variable found dependent on the system response. This was mainly to reduce the overall uncertainty in the model. An ad hoc model was also created based on the temperature dependence of the random variable and monte carlo simulation was revisited. In the end, authors have performed the validation assessment by comparing the prediction and empirical distributions.

Model validation technologies have also been applied to variety of uncertainty models that employ time-domain experimental data [24]. Different statistical methods such as Bayesian parameter estimation coupled with non-linear model validation and verification toolbox in Matlab have also been implemented [25]. Variuos techniques have been presented that differ from our approach, e.g. non-parametric methods (e.g. Lehmann (1975) and Friedman (1991)), robustness methods based on solving a minimax problem (e.g. Huber (1981)) etc. However, little is known about comparative merits of these strategies empirically. Rice et al. [26] presented an extensive study on uncertainty quantification in model validation by considering the hardware of a complex aerospace structure and the finite element model as uncertainty sources. He concluded that the level of uncertainty in the model affects how easily the model is validated and even the future predictions for the model.

The uncertain parameters in all the above studies have been assigned a probability distribution. However, it is not always a case where one has sufficient knowledge about the uncertain parameter. These uncertain parameters can be considered as epistemic which can be better represented as intervals. Implementation of interval

analysis for model uncertainty will simplify decision making for model builders. Moreover, our focus also lies in reducing the reliability on experimentation and thereby quantifying the uncertainty by using different probabilistic, optimization and approximation methods.

1.4. CONTRIBUTION OF THE CURRENT STUDY

As implied from the literature review, experimentation plays an important role in quantifying model uncertainty as the results need to be compared with model prediction. However, it seems unlikely that all the model builder's will be equipped with the ability of experimentation. In some cases, it may not be feasible to perform experiments due to the high cost involved, for example to quantify model uncertainty in drag prediction in airfoils. Model builder will have to rely solely on simulation models to provide accurate results. Also, in most studies on model validation, the model uncertainty is expressed in terms of probabilitic values. Probability density functions (*pdf's*) and cumulative density functions (*cdf's*) are used to convey the results based on uncertainty quantification. Interpretation of these results can be a bit difficult for the model builders.

It is important for any research project to contribute to the "state of the art" in science and engineering from a broad perspective. The current study makes three significant contributions to the topic of engineering design.

- The first contribution involves usage of measurement uncertainty standards in order to communicate the results from the experimenter to the model builder. The standardized output is easy to understand and the cost is drastically reduced as the model builder is no longer responsible for experimentation.

- Secondly, the proposed methodology provides the model uncertainty in terms of an interval which is easier to interpret or visualize. Moreover, the model

uncertainty interval has a particular confidence level which makes the decision making easier. The model builder no longer deals with probabilistic values.

- The third contribution of this study comes from the fact that the methodology is used to propagate the uncertainties through a "black-box" simulation code. Approximation methods like taylor series, second order reliability method and golden search algorithm are used to approximate the original function in case of non-linearity. A response surface is generated to estimate the true value of the model. This response surface is then utilized in place of the original function in order to quantify uncertainty.

1.5. THESIS OUTLINE

Five main sections constitute this manuscript. The first section is an introduction and literature review describing relevant work that has been accomplished in model validation and uncertainty quantification.

Furthermore, the second section will discuss the proposed methodology that can be used to appoximate highly non-linear functions consequently serving as an aide to quantify the model uncertainty. The methodology will be explained taking ASME measurement uncertainty standards into consideration which are used to bridge the gap between the model builder and the experimenter.

The third section of this manuscript consists of a detailed numerical example of a roller coaster design showcasing the application of the methodology in Section 2. Experimentation along with model prediction will be carried out in order to represent the model output in terms of an interval in both the cases. These intervals will be compared to quantify model uncertainty and model shall be updated accordingly.

In the fourth section, thermal challenge problem formulated by Sandia National Laboratory shall be analyzed using the proposed methodology. The uncertainty in the model output will be presented in the form of an interval. The results will be compared

to Monte Carlo simulation (MCS) output which will assess the confidence level of the interval derived using proposed uncertainty quantification approach. Finally, all relevant conclusions along with the future work in model uncertainty quantification will be discussed in the fifth and final section.

2. PROPOSED METHODOLOGY FOR MODEL UQ

The focus of this thesis lies in quantifying model uncertainty using ASME standards of measurement uncertainty. Efforts have been made to integrate measurement uncertainty [27] with the experiment standards, such as those from International Organization for Standardization (ISO), National Institute of Standards and Technology (NIST), and American Society of Mechanical Engineers (ASME). The ASME guidelines for measurement uncertainty are reviewed along with the methodology for uncertainty quantification in this chapter.

2.1. OVERVIEW OF PROPOSED METHODOLOGY

Overview for the process of quantification of model uncertainty can be expressed using the Figure 2.1 below. It resembles a chain of processes of which the integral components are the simulation model, experiments, results with uncertainty, simulation and comparison of simulation and experimental data.

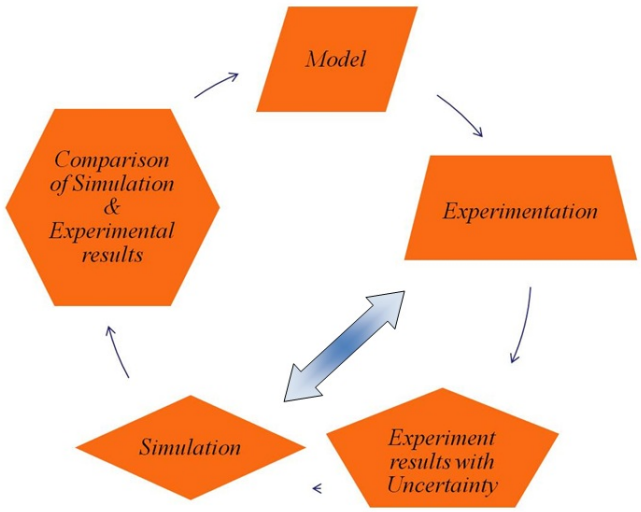

Figure 2.1. Representation of proposed methodology

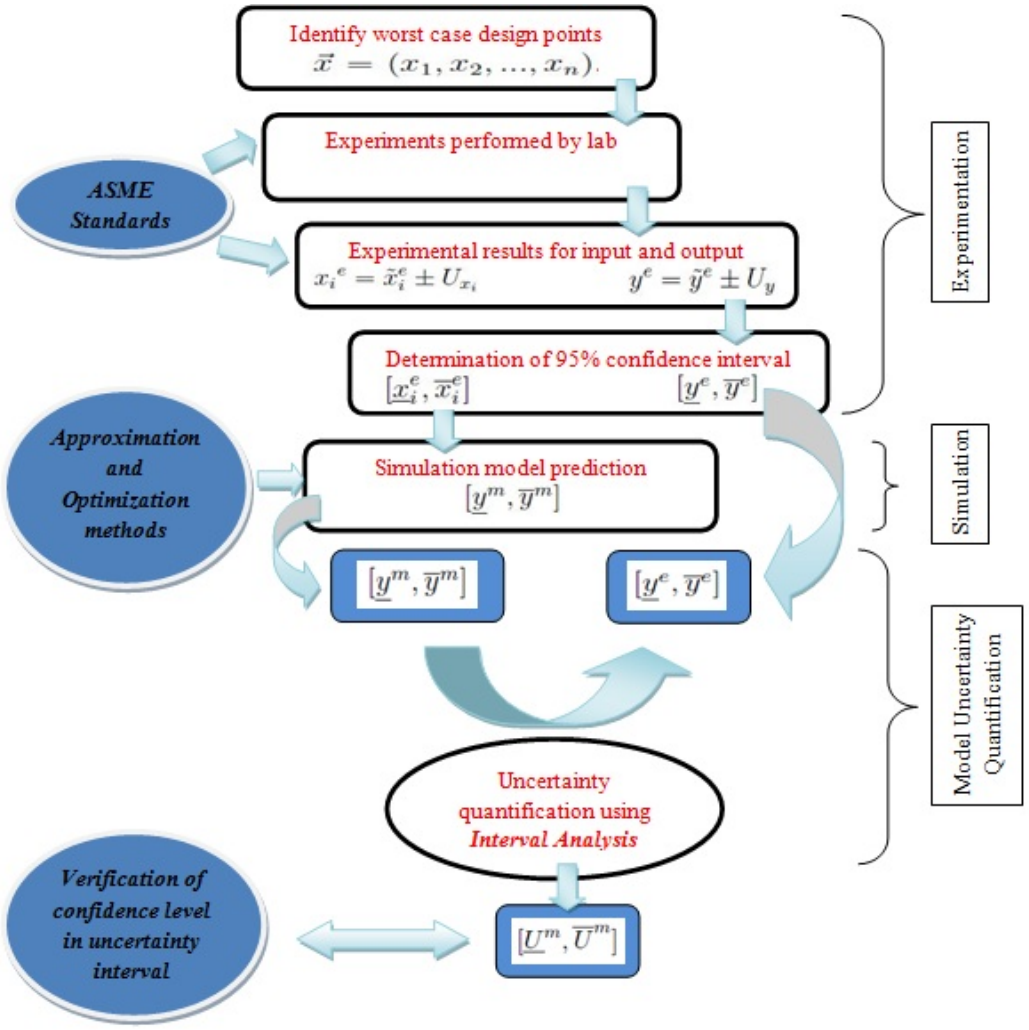

Figure 2.2. Flowchart of proposed methodology

As you can see from the flow chart in Figure 2.2, the proposed methodology is split in three main parts, namely experimentation, simulation and model uncertainty quantification. The first part consists of determining the design points at which the experiments shall be carried out by the professional laboratory for a particular model. For example, suppose the model uncertainty of $y = g(\vec{x})$ is to be quantified where y is the model output and $\vec{x}$ is a vector of all the random input variables. The model builder needs to provide the experimenter with the critical points for $\vec{x}$ at which he thinks that the design might not be safe. Furthermore, the experimenter performs

the experiment using ASME standards of measurement uncertainty as explained in the next section. Experimental results are reported to the model builder in terms of 95% confidence interval for both input and output variables.

In the next part of simulation, the interval input variables from the experiment are used to produce model predictions. Various approximation and optimization techniques are used to simulate the model. The result is again in terms of an interval as shown in the flow chart.

The third and the last part of model uncertainty quantification consists of comparing the experimental and simulation outputs using interval analysis. Moreover, to validate the results, a check for the confidence level in the derived uncertainty interval is performed. This procedure will be explained in detail in this chapter with numerical implementation on two different examples in the coming discussions.

2.2. EXPERIMENTATION USING ASME GUIDELINES

In 2007, ASME published its guidelines for evaluating dimensional measurement uncertainty [28]. Its purpose is to provide the guidelines that are less complex than the Guide to the Expression of Uncertainty in Measurement (GUM) by ISO. The ASME guidelines adopt the terminology from the ISO International Vocabulary of Basic and General Terms in Metrology [29] as follows: Uncertainty of measurement is a "parameter, associated with the result of a measurement that characterizes the dispersion of the values that could reasonably be attributed to the measurand". According to the interpretation of the ASME guidelines, "measurement uncertainty is a number that describes an interval centered about the measurement result where there is reasonable confidence that it includes the *true value* of the quantity being measured". This number is called *expanded uncertainty*, denoted by U, with a coverage factor of 2 denoted by k in this dissertation, or roughly a confidence level of 95%. The confidence level indicates the probability of including the true value of the

quantity that is being measured within a given interval. The expanded uncertainty is the end product of an uncertainty evaluation during experimentation. In other words, the measurement result is reported by

$$\tilde{y}^e \pm U_y \tag{1}$$

where $\tilde{y^e}$ is the measurement result, and U_y is the associated expanded uncertainty. Another concept is the *measurement error* ϵ^e. It is defined as the difference between the measured value $\tilde{y^e}$ and the true value y. The measurement error is never exactly known, but with the expanded uncertainty in Equation (1), it can be stated that the measurement error is $\pm U_y$ with a 95% level of confidence. The expanded uncertainty U_y can be obtained from the combined standard uncertainty u_{cy} with the following equation

$$U_y = k u_{cy} \tag{2}$$

where the coverage factor is k, which indicates a 95% level of confidence as mentioned previously. The combined standard uncertainty u_{cy} is a quantitative value that describes the magnitude of all the uncertainty sources. It can be obtained from individual standard uncertainties. A standard uncertainty, denoted by u_y from a single uncertainty source, is also a quantitative value that describes the magnitude of the uncertainty source. The standard uncertainty can be considered as *one standard deviation* of the potential variation associated with the uncertainty source. Each uncertainty source must be evaluated to produce its standard uncertainty.

Suppose there are M sources of uncertainty during the experimentation and the standard uncertainties are u_{y_i} with $i = 1, 2, \ldots, M$. Also assume that the sources of uncertainty are independent. The combined standard uncertainty u_{cy} is given by

$$u_{cy} = \sqrt{\sum_{i=1}^{M} u_{y_i}^2} \tag{3}$$

In some cases multiple sets of observations are available, but these sets of data may not have the same sizes and even same means. These sets cannot be directly combined to calculate a standard uncertainty. For these cases, to pool the data, the following equation is used to calculate the standard uncertainty for an uncertainty source

$$u_{y_i} = \sqrt{\frac{\sum_{j=1}^{N} (h_j - 1){s_j}^2}{\sum_{j=1}^{N} (h_j - N)}} \tag{4}$$

where h_j is the number of observations in the j-th data set, and s_j is the sample standard deviation of the j-th data set, and N is the number of data sets. A rough procedure can be summarized as follows: At first, indentify all the uncertainty sources; then calculate the standard uncertainty for each uncertainty source using Equation (4). After combining the standard uncertainties from all the uncertainty sources, the combined standard uncertainty can be obtained using Equation (3). The end product of the process is then expressed by an interval given in Equation (1). In summary, when the ASME guidelines are followed, the experimental results will be reported with the expanded uncertainty in the form of intervals under certain confidence level as shown in Equation (1).

2.3. EXPERIMENTATION ON MODEL

In recent years, there has been a great deal of work in developing methods for validating a model. A general methodology can be explained as simply comparing the prediction from the modeler to the new observation from the experiment. However, it is difficult to run an experiment for every single case and simulation comes up as the best option. The simulation itself will use the same model for providing an output and

if in case there is uncertainty in the model, then the output is destined to be erroneous. This becomes important for many real world complex cases. Thus, there is a need to device a method to quantify uncertainty when experimentation is not possible at the modeler's end. In such cases, the experimentation part can be outsourced to a professional laboratories which have the required facilities and infrastructure to do so. Our goal is to create a common platform for the experimenter and the model builder. This can be achieved by using the ASME standards of measurement uncertainty mentioned in the previous section.

Consider a model with system response y^m which is a function of q model parameters $\vec{w} = (w_1, w_2, ..., w_q)$ and n input variables $\vec{x} = (x_1, x_2, ..., x_n)$. Input variables are under the control of the experimenters and can be changed during experimentation whereas model parameters are deterministic and hence constant. Uncertainty in the input variables can be quantified through experimentation but it would be a good point to think about the uncertainty present in the model parameters. For example, in different hypersonic re-entry problems, recombination efficiency [30] is modeled as an epistemic uncertainty due to the fact that it arises due to lack of knowledge in a physical model [31]. Sometimes in complex cases, mixed uncertainty (aleatory and epistemic) are prevalent in the model parameters and methods like Second Order Probability Theory [32, 33] are used. The uncertainty in model parameter $\vec{w}$ is considerd as a source of uncertainty in the proposed methodology. Suppose,

$$y^m = g(\vec{x}, \vec{w}) \tag{5}$$

and the experiments are carried out by the particular lab as the model builder is considered to be incapable of carrying out experiments in this work due to the inavailability of right facility or it is time consuming to do so. According to the standards of ASME measurement uncertainty, the lab experimental results are reported

to the model builder in terms of expanded uncertainty for each input variable and the system response. If the confidence level is 95%, then there is 95% chance that the interval for y^e will cover the true value of y. The measured output for system response is reported as

$$y^e = \tilde{y}^e \pm U_y \tag{6}$$

where y^e is the experimental measurement.

Experimental bias and random error [34] are two components of experimental uncertainty. These two components are included in the expanded uncertainty concept U_y. From Equation (6), 95% confidence interval can be constructed for the output response y^e which is given by

$$y = y^e = \tilde{y}^e \pm U_y = [\underline{y}^e, \overline{y}^e] \tag{7}$$

where the lower bound is

$$\underline{y}^e = \tilde{y}^e - U_y \tag{8}$$

and the upper bound is

$$\overline{y}^e = \tilde{y}^e + U_y \tag{9}$$

According to previous model validation literatures [13], $\tilde{y}^e$ is considered as a point measurement which introduces the probability of experimental error ϵ^e to be present.

$$y^e = \tilde{y}^e \pm \epsilon^e \tag{10}$$

Along with the system response, the model builder will also need the input variables to be measured. The measurement of $\vec{x}$ will serve as the input to the model during model uncertainty quantification process, commonly known as model prediction process. The result of an input variable will be presented to the model

builder using the same procedure as for system output response

$$x_i^e = \tilde{x}_i^e \pm U_{x_i} = [\underline{x}_i^e, \overline{x}_i^e] \tag{11}$$

where $i = 1, 2, ..., n$ and U_{x_i} is the expanded uncertainty of the i^{th} input variable. The lower bound is

$$\underline{x}_i = \tilde{x}_i^e - U_{x_i} \tag{12}$$

and the upper bound is

$$\overline{x}_i = \tilde{x}_i^e + U_{x_i} \tag{13}$$

In vector form, the above equations can be represented as

$$\vec{x}^e = \vec{\tilde{x}}^e \pm \vec{U}_{\vec{x}} = [\underline{\vec{x}}, \overline{\vec{x}}] \tag{14}$$

where the lower bound is

$$\underline{\vec{x}} = \vec{\tilde{x}}^e - \vec{U}_{\vec{x}} \tag{15}$$

and the upper bound is

$$\overline{\vec{x}} = \vec{\tilde{x}}^e + \vec{U}_{\vec{x}} \tag{16}$$

where

$$\vec{\tilde{x}} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$$

$$\underline{\vec{x}} = (\underline{x}_1, \underline{x}_2, ..., \underline{x}_n)$$

$$\overline{\vec{x}} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$$

and

$$\vec{U}_{\vec{x}} = (U_{x_1}, U_{x_2}, ..., U_{x_n})$$

2.4. SIMULATION ON MODEL

ASME standards of measurement uncertainty serve as a bridge to connect both of them in a way that the data transmitted can be understood universally. Once the required data has reached the model builder, he has both, the system response interval along with the individual input variable interval. The model builder chooses the interval variable information as the input for the model prediction. Clearly, as the input variables are in the form of an interval, the end result for model output will be an interval. Mathematically it can be shown as follows

$$y^m = g(\vec{x}^e, \vec{w}) = g(\vec{\bar{x}}^e \pm \vec{U}_{\vec{x}}, \vec{w}) = g([\underline{\vec{x}}^e, \overline{\vec{x}}^e], \vec{w}) = [\underline{y}^m, \overline{y}^m] \tag{17}$$

where the lower bound for the model prediction is

$$\underline{y}^m = \min g(\vec{x}^e, \vec{w}), \vec{x}^e \in [\underline{\vec{x}}^e, \overline{\vec{x}}^e] \tag{18}$$

and the upper bound for the model prediction is

$$\overline{y}^m = \max g(\vec{x}^e, \vec{w}), \vec{x}^e \in [\underline{\vec{x}}^e, \overline{\vec{x}}^e] \tag{19}$$

Equations (18) and (19) represent the upper and lower bounds of the system response variable. In order to determine the same for highly non-linear problems, different mathematical techniques and approximation methods have been used which are listed as follows:

- Optimization
- First Order Taylor series
- Worst Case Analysis
- Second Order Reliability Method (SORM)

- Golden Search Algorithm
- Monte Carlo Analysis on SORM model
- Monte Carlo Analysis on original model

Furthermore, application of interval analysis on the results obtained quantified the model uncertainty which shall be explained in the next section.

2.4.1. Optimization. In cases where a best element is to be selected from a number of alternatives, optimization is the best method to opt for. It will find the "best available"point for an objective function given a defined domain. Optimization can be of two types, constrained and unconstrained. The former solves the objective function for the best estimate within a fixed domain subject to certain constraints. Thus, the solver maximizes or minimizes the objective function with the best possible estimate. Mathematically, it can be represented as

The model output interval is

$$y^m = [\underline{y}^m, \overline{y}^m] \tag{20}$$

where the optimization model for $\underline{y}^m$ is given by

$$\left.\begin{aligned} \underline{y}^m &= \min g(x) \\ &\text{subject to} \\ x &\in [\underline{\vec{x}}^e, \overline{\vec{x}}^e] \end{aligned}\right\} \tag{21}$$

Similarly, the optimization model for $\overline{y}^m$ is given by

$$\left.\begin{aligned} \overline{y}^m &= \max g(x) \\ &\text{subject to} \\ x &\in [\underline{\vec{x}}^e, \overline{\vec{x}}^e] \end{aligned}\right\} \tag{22}$$

The same function is used to maximize the model with the change that the objective function is negative.

2.4.2. First Order Taylor Series. There are several appoximation methods which are used to simplify computer evaluations in order to make them computationally inexpensive. These methods are devised to reduce the non-linearity of the objective functions and arrive at the best possible estimate without sacrifying much on accuracy. One of the examples is first order Taylor series [35, 36] which uses Taylor series expansion to approximate the objective function. The basic idea is to linearize the complex model problem by using first order derivatives. Furthermore, after linearizing the model, find the mean and standard deviation of the objective function. As per ASME, using a coverage factor of 2, the upper and lower bounds for the objective function are calculated. Mathematically, If y is the model output and $\vec{x}$ is the vector of input variables,

$$\tilde{y}(\vec{x}) = y(\tilde{\vec{x}}) \tag{23}$$

where $\tilde{y}$ is the mean of the objective function and $\tilde{\vec{x}}$ represents the vector of mean values of input variables.

$$\sigma_y = \sqrt{\left(\frac{\partial y}{\partial \vec{x}}\sigma_{\vec{x}}\right)^2} \tag{24}$$

where

$$\sigma_y = \text{ standard deviation of the objective function}$$

$$\sigma_{\vec{x}} = \text{ standard deviation of input variables}$$

The subsequent step after this is to calculate lower and upper bounds as per Equation (25) using ASME standards of measurement uncertainty.

$$\tilde{y} \pm 2\sigma_y \tag{25}$$

With the linear function obtained, a worst case analysis is performed in the next section which considers extreme conditions into consideration in order to evaluate the function bounds.

2.4.3. Worst Case Analysis. Worst case analysis is basically used to determine the design margin for a system response. The design margin is the measure of the margin between worst case performance of a system and the performance required by specification. Positive design margin indicates that the system will perform with margin to spare. On the other hand, negative design margin indicates that the system will not meet its specification. Hence, not performing worst case analysis can can be risky as the system will not be tested for its unidentified quantity of input variable.

At first, the system response is linearized by using first order taylor series expansion at average values of input variables $\vec{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$. First order series is given by

$$f(\vec{x}) \approx f(\overline{\vec{x}}) + \sum_{i=1}^{n} (x_i - \overline{x}_i) \tag{26}$$

The performance function for the above model is given by

$$\triangle f = f(\vec{x}) - f(\overline{\vec{x}}) = \sum_{i=1}^{n} \frac{\partial f(\vec{x})}{\partial x_i} (x_i - \overline{x}_i) \tag{27}$$

The worst case interval is then represented by $[\overline{f} - \triangle f, \overline{f} + \triangle f]$

As the worst case analysis linearizes the model, the solution will contain some error especially when the system response is non-linear. Also it is too conservative. The search for lower and upper bounds is continued for accuracy by increasing the order of approximation in the next section.

2.4.4. Second Order Taylor Series. Higher order approximations are more reliable for non linear problems as they account for the complex nature of the objective function. Second order taylor series [37] estimates the original function at the mean values of the input variables $\tilde{\vec{x}}$ using second order taylor expansion. The approximation is given by

$$y(\vec{x}) \approx y(\tilde{\vec{x}}) + \bigtriangledown(\tilde{\vec{x}})(\vec{x} - \tilde{\vec{x}})^T + \frac{1}{2}(\vec{x} - \tilde{\vec{x}})H(\tilde{\vec{x}})(\vec{x} - \tilde{\vec{x}})^T \tag{28}$$

where $H(\tilde{\vec{x}})$ is the Hessian matrix at mean values of $\vec{x}$. It is given by

$$\begin{bmatrix} \frac{\partial^2 y}{\partial {x_1}^2} & \frac{\partial^2 y}{\partial x_1 x_2} & \cdots & \frac{\partial^2 y}{\partial x_1 x_n} \\ \frac{\partial^2 y}{\partial x_2 x_1} & \frac{\partial^2 y}{\partial {x_2}^2} & \cdots & \frac{\partial^2 y}{\partial x_2 x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 y}{\partial x_n x_1} & \frac{\partial^2 y}{\partial x_n x_2} & \cdots & \frac{\partial^2 y}{\partial {x_n}^2} \end{bmatrix} \tag{29}$$

Equation (28) is utilized to obtain a response surface which was further utilized for the prediction of lower and upper bounds of the model for uncertainty interval.

2.4.5. Golden Search Algorithm (GSA). Numerical methods [38] are more than useful for approximating complex functions. As a matter of fact that while dealing with intervals, golden section algorithm being one of the most accurate methods can be used to determine the maxima or minima of the response variable within the intervals. The primary objective of golden algorithm is to reduce the interval size at each iteration of function evaluation. The reduced interval size in a particular iteration is chosen at a fixed ratio commonly known as *golden section ratio* relative to the interval size in that iteration.

Suppose for a function $F(\vec{x})$, there lies a point z in the interval $[a, b]$ such that

$$\left.\begin{array}{rrr} F(\vec{x}) & \text{decreasing on} & [a, z] \\ F(\vec{x}) & \text{increasing on} & [z, b] \end{array}\right\} \quad \text{F(z) is minimum}$$

OR

$$\left.\begin{array}{rrr} F(\vec{x}) & \text{increasing on} & [a, z] \\ F(\vec{x}) & \text{decreasing on} & [z, b] \end{array}\right\} \quad \text{F(z) is maximum}$$

Consider a case of minimizing a function. The next step is to choose points c and d suct that $c < d$ and the intervals $[a, c]$ and $[d, b]$ are symmetric. If $F(d) \geq F(c)$, then $z \in [a, d]$ becomes the new interval or else $z \in [c, b]$. The golden section ratio which is used to select the new points is given by

$$\frac{d-a}{b-d} = 1.61803$$

The derivation of the golden section ratio is shown in Gerald et al. [39]. Apart from the golden ration, the iteration process is also controlled by a convergence criteria δ which is known as the *relative tolerance*. Mathematically,

$$\frac{b_n - a_n}{b-a} = \delta$$

where $[a_n, b_n]$ is the interval at the n^{th} iteration. The iterative procedure can be summarized as follows:

- Let [a, b] be the initial interval.
- First step ($i = 0$), determine two independent points c and d using the golden section ratio.
- At the i^{th} step, if $F(c_{i-1}) \leq F(d_{i-1})$ then

  - $a_i = a_{i-1}$
  - $b_i = d_{i-1}$
  - $d_i = c_{i-1}$
  - Recalculate c_i using the new interval point d.

OR

- if $F(c_{i-1}) \geq F(d_{i-1})$ then
  - $a_i = c_{i-1}$
  - $b_i = b_{i-1}$
  - $c_i = d_{i-1}$
  - Recalculate d_i using the new interval point c.

On the contrary, for maximization of the objective function, the logic is reversed. By this method, if there are N steps to be performed, there has to be $N + 1$ function evaluations. Consider a cubic polynomial fit in the last iteration to determine the function value, thus giving a better accuracy than the second order approximation.

2.4.6. Monte Carlo Simulation (MCS) on Second Order Approximation and the Original Model. Monte Carlo analysis which relies on repeated random sampling is considered to be one of the accurate methods to estimate the true value of a function. As the number of samples increase, the estimate is more and more closer to the true mean value and thus the accuracy is improved. The basic steps followed for MCS are

- Assume a particular *distribution* for the input variables $\vec{x}$ as the experimental information is in terms of interval which is lower and upper bounds.
- Generate N samples for the input random variables using the means and standard deviations from the experimental results. As the results are provided in

terms of expanded uncertainty with a coverage factor of 2, calculate the standard deviation for each variable by dividing the uncertainty by 2. The samples are denoted by $\vec{x}_i$ where $(i = 1, 2, \ldots, N)$.

- Using these samples as input to the model, generate N output samples for model prediction y^m.

Thus, the lower and the upper bounds can now be calculated from these model predictions and re-write y^m as $[\underline{y}^m, \overline{y}^m]$.

2.5. INTERVAL ANALYSIS AND UQ

Since this study is mainly focussed on efficient methodology for quantification of model uncertainty, interval analysis [40, 41, 42] will be used to create a model uncertainty interval with the help of experimentation and model prediction outputs.

Quantifying the model uncertainty can be done by using interval analysis methods like interval arithmetic [43, 44, 45]. Two intervals for model uncertainty quantification are, (1) the experimental result $[\underline{y}^e, \overline{y}^e]$ and (2) simulation result $[\underline{y}^m, \overline{y}^m]$ respectively. To explain the methodology, let them be represented by $[a, b]$ and $[c, d]$ denoting lower and upper bounds for each interval respectively. Simple arithmetic interval operations are

$$[a, b] + [c, d] = [a + c, b + d] \tag{30}$$

$$[a, b] - [c, d] = [a - d, b - c] \tag{31}$$

$$[a, b] * [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \tag{32}$$

$$\frac{[a, b]}{[c, d]} = \left[\min\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right), \max\left(\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\right)\right] \tag{33}$$

Suppose the model error is denoted as ϵ^m and hence the true quantity is given by

$$y = y^m + \epsilon^m \tag{34}$$

Using Equations (7),(10) and (34),

$$\epsilon^m = y^e - y^m = \tilde{y}^e \pm \epsilon^e - y^m \tag{35}$$

As the experimental error ϵ^e cannot be estimated exactly, quantifying model error ϵ^m becomes difficult. Expanded uncertainty concept U_y is useful in this situation as it can replace experimental error ϵ^e thereby giving us an estimate of model error ϵ^m with a certain level of confidence.

Let the estimate of model error be defined as *model uncertainty* and denoted by U^m. Now to quantify the model uncertainty U^m, use interval arithmetic provided that model prediction y^m and the experimental observation y^e are independent. In this case, assume them to be independent and the model uncertainty is given by

$$U^m = [\underline{U}^m, \overline{U}^m] \tag{36}$$

with

$$\underline{U}^m = \underline{y}^e - \overline{y}^m \quad \text{and} \quad \overline{U}^m = \overline{y}^e - \underline{y}^m \tag{37}$$

or

$$U^m = [\underline{U}^m, \overline{U}^m] = [\underline{y}^e - \overline{y}^m, \overline{y}^e - \underline{y}^m] \tag{38}$$

Now that the result for model uncertainty [46, 47] is in the form of an interval, it is upon the model builder whether to accept the model with uncertainty or reject it. The criteria can be decided depending upon the intended application. In aerospace applications, quantifying model uncertainty becomes extremely important due to the huge cost involved. In order to quantify uncertainty for the model, various techniques were used to find the lower and upper bounds. Next important step is to check whether the derived model uncertainty interval is of the same confidence level as the experimental readings? There might be cases when the confidence level is even higher

than 95%. Verification of the confidence level in the estimated uncertainty interval is carried out in the next section.

2.6. VERIFICATION OF CONFIDENCE LEVEL IN INTERVAL

As mentioned above, the experimental results are presented to the model builder with a 95% confidence interval. However, the confidence related to the derived model uncertainty interval in Equation (38) is still not proved. In order to validate the same, it needs to be proved that the confidence level in the derived uncertainty interval is at least 95% or more. The data in hand is as follows:

- Variables of the input vector $\vec{x} = (v, r)$ are independent.
- Experimental results for input vector $\vec{x}$ and system response y^e are reported to the model builder in terms of intervals using expanded uncertainty concept with confidence level of $(1 - \alpha)$.

$$\left.\begin{aligned} Pr\{\underline{x}_i^e \leq x_i^e \leq \overline{x}_i^e\} = 1 - \alpha \\ Pr\{\underline{y}^e \leq y^e \leq \overline{y}^e\} = 1 - \alpha \end{aligned}\right\} \quad (i = 1, 2, \ldots, n) \tag{39}$$

Given this information, assume that the input variables are normally and independently distributed with $x_i \sim N(\mu_{x_i}, \sigma_{x_i})$ where μ_{x_i} is the mean and σ_{x_i} is the standard deviation of the i^{th} variable with $(i = 1, 2, \ldots, n)$. Since the confidence level of $x_i \in [\underline{x}_i, \overline{x}_i]$ is $(1 - \alpha)$,

$$Pr\{x_i < \underline{x}_i\} = 0.5\alpha \tag{40}$$

$$\therefore Pr\{x_i < \underline{x}_i\} = \Phi\left(\frac{\underline{x}_i - \mu_{x_i}}{\sigma_{x_i}}\right) = 0.5\alpha \tag{41}$$

where Φ is the cumulative distribution function (CDF) of the standard normal variable. Now, from Equation (41) it can be shown that

$$\frac{\underline{x}_i - \mu_{x_i}}{\sigma_{x_i}} = \Phi^{-1}(0.5\alpha) = -k \tag{42}$$

where $k = -\Phi^{-1}(0.5\alpha) > 0$. Using the symmetry of a normal distribution and Equation (42),

$$\underline{x}_i - \mu_{x_i} = -k\sigma_{x_i} \tag{43}$$

and

$$\overline{x}_i - \mu_{x_i} = k\sigma_{x_i} \tag{44}$$

Assume a linear approximation for the model if the uncertainty in x_i measurement is small. Suppose,

$$y^m = z_0 + \sum_{i=1}^{n} z_i x_i = z_0 + \sum_{i=1}^{n} p_i \tag{45}$$

where z_0 and z_i are constant coefficients with $(i = 1, 2, \ldots, n)$. Also, $p_i = z_i x_i$ is a linear relationship from the above equation . Thus, p_i is normally distributed with $p_i \sim N(\mu_{p_i}, \sigma_{p_i})$ where

$$\mu_{p_i} = z_i \mu_{x_i} \quad \text{and} \quad \sigma_{p_i} = z_i \sigma_{x_i} \tag{46}$$

Similarly, find the mean and standard deviation for the model prediction,

$$\mu_{y^m} = z_0 + \sum_{i=1}^{n} \mu_{p_i} \quad \text{and} \quad \sigma_{y^m} = \sqrt{\sum_{i=1}^{n} \sigma_{p_i}^2} \tag{47}$$

Knowing the distribution of the model output with its mean and standard deviation, the bounds can be determined as follows:

$$\underline{y}^m = z_0 + \sum_{i=1}^{n} \underline{p}_i \quad \text{and} \quad \overline{y}^m = z_0 + \sum_{i=1}^{n} \overline{p}_i \tag{48}$$

Also assume that the output measurement y is normally distributed with its mean μ_{y^e} and standard deviation σ_{y^e}. Referring to Equations (43) and (44),

$$\underline{y}^e - \mu_{y^e} = -k\sigma_{y^e} \tag{49}$$

and

$$\overline{p}_i - \mu_{p_i} = k\sigma_{p_i} \tag{50}$$

Now, move on to finding the bounds for model uncertainty U^m using the Equations (38).

$$\underline{U}^m = \underline{y}^e - \overline{y}^m = \underline{y}^e - \left[z_0 + \sum_{i=1}^{n} \overline{p}_i \right] \tag{51}$$

and

$$\overline{U}^m = \overline{y}^e - \underline{y}^m = \overline{y}^e - \left[z_0 + \sum_{i=1}^{n} \underline{p}_i \right] \tag{52}$$

As a result, model uncertainty represented by $y^e - y^m$ is normally distributed with mean and standard deviation as follows

$$\mu_{U^m} = \mu_{y^e} - \mu_{y^m} = \mu_{y^e} - \left(z_0 + \sum_{i=1}^{n} \mu_{p_i} \right) \tag{53}$$

and

$$\sigma_{U^m} = \sqrt{\sigma_{y^e}^2 + \sum_{i=1}^{n} \sigma_{p_i}^2} \tag{54}$$

Thus, one sided probability for model uncertainty can be represented as

$$Pr\{U^m < \underline{U}^m\} = \Phi\left(\underline{U}^m - \mu_{U^m}\right) \tag{55}$$

$$= \Phi\left(\frac{\underline{y}^e - \left(z_0 + \sum_{i=1}^{n} \overline{p}_i \right) - \mu_{y^e} + \left(z_0 + \sum_{i=1}^{n} \mu_{p_i} \right)}{\sigma_{U^m}} \right) \tag{56}$$

$$\therefore Pr\{U^m < \underline{U}^m\} = \Phi\left(\frac{\underline{y}^e - \mu_{y^e} - \sum_{i=1} n(\overline{p}_i - \mu_{p_i})}{\sqrt{\sigma_{y^e}^2 + \sum_{i=1}^{n} \sigma_{p_i}^2}} \right) \tag{57}$$

Resusbstitute the values in the numerator using Equations (49) and (50),

$$Pr\{U^m \leq \underline{U}^m\} = \Phi\left[\frac{-k\sigma_{y^e} - \sum_{i=1}^{n} k\sigma_{p_i}}{\sqrt{\sigma_{y^e}^2 + \sum_{i=1}^{n} \sigma_{p_i}^2}}\right] = \Phi\left[-k\frac{\sigma_{y^e} + \sum_{i=1}^{n} k\sigma_{p_i}}{\sqrt{\sigma_{y^e}^2 + \sum_{i=1}^{n} \sigma_{p_i}^2}}\right] \tag{58}$$

Mathematically,

$$(\sigma_{y^e} + \sum_{i=1}^{n} k\sigma_{p_i}) \geq \sqrt{\sigma_{y^e}^2 + \sum_{i=1}^{n} \sigma_{p_i}^2} \tag{59}$$

$$\therefore Pr\{U^m < \underline{U}^m\} < \Phi(-k) < 0.5\alpha \tag{60}$$

Equation (60) proves the fact that for a linear model, the confidence level of uncertainty bounds derived through the proposed methodology in Section 2, is greater than or equal to $(1-\alpha)$ which in turn is the confidence level of the exprimental results. In mathematical terms, it can be expressed as,

$$Pr\{\underline{U}^m \leq U^m \leq \overline{u}^m\} \geq 1 - 2Pr\{U^m < \underline{U}^m\} = (1-\alpha) \tag{61}$$

Thus, if the linearization of the model approximates the original model with accuracy and ASME guidelines are followed in determining the uncertainty interval, then the confidence level for the uncertainty bounds is greater than or equal to 95%. However, if the original model is highly non-linear, the confidence level achieved with the uncertainty interval cannot be guaranteed. This situation will be dealt with in the next chapter in which sampling methods are used to gauge the confidence level for a non-linear function.

3. ROLLER COASTER MODEL PROBLEM

The numerical implementation of the proposed methodology will be the main focus in this section of the thesis. Basically, the examples used here will help understand the importance of the uncertainty interval model. At first, a simple example of a roller coaster is used to demostrate the procedure to be followed for model validation. Detailed description of the model problem is given in the sections below along with the energy and work principle used to derive the model.

3.1. DESCRIPTION OF THE MODEL PROBLEM

One of the most basic requirements for a roller coaster design is that the car should always keep in touch with the track. This will ensure that the car travels around the inside loop without leaving the track. The roller coaster works on an energy conversion concept. At first, the coaster is given an initial ascent to build up the potential energy which is often referred to as energy of position. Higher the coaster goes in the air, faster is the pull from the gravity force. The built up potential energy is converted to kinetic energy as the coaster descents. This energy of motion gives the momentum to the coaster which in turn helps the coaster to traverse the loop at the end as shown in Figure 3.1.

One of the design tasks for roller coaster design includes the determination of minimal height h to give the coaster enough momentum to cross the loop without going off-track. The expression for minimal height is given by

$$h = \frac{5}{2}r - \frac{1}{2}\frac{v^2}{g} \tag{62}$$

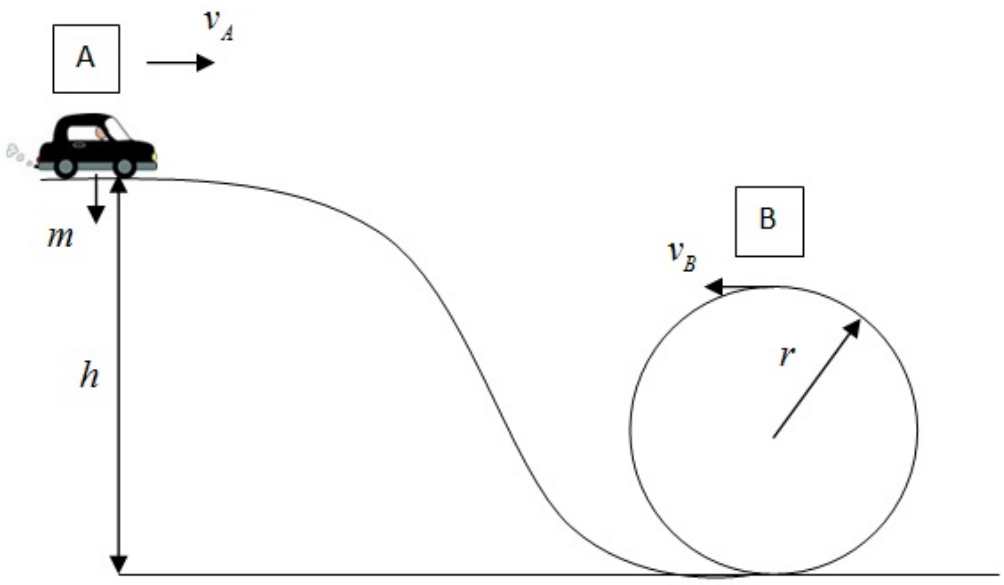

Figure 3.1. Roller coaster design

The main task is to validate this model and test its accuracy. Model validation is correctly defined as checking how accurate a model can predict the real world phenomena. However, the expression for the minimum height needs to be derived, as shown in Equation (62). The energy and work principle shall be used for the same.

3.2. MODEL PROBLEM DERIVATION

As shown in Figure 3.1, the coaster starts descenting from a height h (at point A) and gains required momentum. When the coaster is located at a height h, it has a reservoir of stored energy due to its position. This is mainly known as the *potential energy* and denoted as

$$\text{Potential Energy at A} = mgh \tag{63}$$

Let the initial speed of the coaster be v_A. Due to the velocity, stored potential energy gets converted into *kinetic energy* and can be denoted as

$$\text{Kinetic Energy at A} = \frac{1}{2}mv_A^2 \tag{64}$$

At height h, potential energy is at the maximum and kinetic energy at the minimum. Similarly, when the coaster is about to start with the loop trajectory, potential energy is at the minimum while kinetic energy is at its maximum. At point B, the height is equal to twice the loop radius r and the potential and kinetic energies is given by

$$\text{Potential energy at B} = mg2r \tag{65}$$

and

$$\text{Kinetic energy at B} = \frac{1}{2}mv_B^2 \tag{66}$$

The total energy at a point will be the summation of both potential and kinetic energies at that particualr location. For a system, total energy is always conserved. Thus, by the energy conversion principle, total energy at point A is equal to total energy at point B. Mathematically, it can be shown from the Equations (63), (64), (65) and (66) that

$$mgh + \frac{1}{2}mv_A^2 = mg2r + \frac{1}{2}mv_B^2 \tag{67}$$

At point B, the coaster will remain in contact with the track only if the normal reaction in the upward direction is equal to its weight due to the gravitational force g. Using Newton's second law of motion, the net force on a particle is equal to the time rate of change of its linear momentum. Mathematically it can be written as

$$F = ma \tag{68}$$

where F is the net force applied on a body, m is the mass of the body and a is the body's acceleration. The force F on the coaster at point B is only due to the gravitational pull and is given by Equation (65).

$$\text{Acceleration at point B} = m\frac{v_B^2}{r} \tag{69}$$

$$\therefore mg = m\frac{v_B^2}{r} \tag{70}$$

Hence, two unknowns v_B and h can be obtained from two equations. Solve them simultaneously to get the model for minimum height h as denoted in Equation (62). Relativity of terms in the Equation (62) can be explained as follows

- The required minimal height h is the model output y, $y^m = h$.
- As per the definition in Section 2, the model parameter $\vec{w}$ is the gravitational force g which cannot be changed during the experiment, $\vec{w} = g$.
- The model input variables $\vec{x}$ are the velocity v and the radius of the loop track r, $\vec{x} = (v, r)$.

The assumptions made while deriving the model are as follows

- Total energy in the system is conserved.
- The track is smooth and friction between the track and car is negligible.
- The car is actually treated as a particle and hence its motion is translational.

The data in hand with the model builder is that the highest velocity of the car is $v = 3$ m/s and the largest loop radius is $r = 10$ m. The model validation and the uncertainty quanitfication is necessary due to the reasons based on the above assumptions.

- Energy is lost to the surroundings due to friction and some intangible factors such as environmental conditions.
- The friction between the car and track exists.
- Along with translational motion, the car may have rotational motion.

Thus, the derived model is not perfectly accurate and uncertainty in the model prevails. Now start the validation process first step of which is experimentation in lab.

3.3. EXPERIMENTATION ON ROLLER MODEL PROBLEM

Basic design principle is to consider the worst case in design analysis so as to render it as a safe design. Similarly, the model builder would want to quantify model uncertainty at the worst case point and therby requests an external laboratory to perform experiments at $\vec{x} = (v, r) = (3, 10)$. In order to simulate the experiment, assume that the random input variables (v, r) are normally distributed with their means (3 m/s, 10 m) and expanded uncertainties (0.02 m/s, 0.002 m) uncer confidence level of 95%. Thus, 95% confidence interval for $\vec{x} = (v, r)$ can be expressed as follows

$$x_1^e = v^e = 3 \pm 0.02 \quad \text{m/s}$$
$$\therefore [\underline{x}_1^e, \overline{x}_1^e] = [\underline{v}^e, \overline{v}^e] = [2.98, 3.02]\,\text{m/s} \tag{71}$$
$$x_2^e = r^e = 10 \pm 0.002 \quad \text{m}$$
$$\therefore [\underline{x}_2^e, \overline{x}_2^e] = [\underline{r}^e, \overline{r}^e] = [9.998, 10.002]\,\text{m} \tag{72}$$

Generate 30 samples from the assumed distributions for input variables. Also, assume that actual energy loss for the system is between 95% and 99% with a uniform distribution. Along with the input variables, the experimenter also tests the system response which is the required height h in this case. The output will be presented to

the model builder in terms of expanded uncertainty as shown in Equation (1).

$$y^e = h^e = 25.2794 \pm 0.5864 \quad \text{m}$$
$$\therefore [\underline{y}^e, \overline{y}^e] = [\underline{h}^e, \overline{h}^e] = [24.6930, 25.8658]\,\text{m} \tag{73}$$

This provides us with a 95% confidence interval for the output variable.

3.4. SIMULATION WITH UNCERTAINTY QUANTIFICATION

Once the model builder has the experimenatl results in terms of standards of measurement uncertainty, it is not difficult to construct a confidence interval as in Equations (71) and (72). He then plugs $[\underline{x}_1^e, \overline{x}_1^e]$ and $[\underline{x}_2^e, \overline{x}_2^e]$ into the model Equation (62). As discussed earlier, the interval inputs provide us with an interval output for system response as shown below

$$h = [\underline{y}^m, \overline{y}^m] = \frac{5}{2}[9.998, 10.002] - \frac{1}{2}\frac{[2.98, 3.02]^2}{9.81} = [24.5301, 24.5524]\,\text{m} \tag{74}$$

The model for this problem is simple and hence, directly plug the experimental results for input variables and procure the system response interval. However, if the function is highly non-linear, approximation methods like Taylor expansion series, Optimization, Second order probability method, Worst case analysis etc. may be used before implementing the interval analysis. These methods will be explained in the next example which has a non-linear model. Equation (31) quantifies model uncertainty for our model problem by using the experimental and model prediction results. Thus, model uncertainty using Equation (38) is given by

$$\begin{aligned} U^m &= y^e - y^m \\ &= [\underline{y}^e, \overline{y}^e] - [\underline{y}^m, \overline{y}^m] \\ &= [24.6930, 25.8658] - [24.5301, 24.5524] = [0.1406, 1.3357]\,\text{m} \end{aligned} \tag{75}$$

Equation (75) shows that the interval for model uncertainty is not too wide and the uncertainty is relatively small. Also, the sign of model uncertainty interval is positive which concludes that the model slightly underestimates the true minimal height required for the roller coaster's ascent. This might result in a risky design if the coaster does not attain the required momentum in order to traverse the loop trajectory causing the design to fail. With this uncertainty interval, model builder is in a better position to decide whether to accept or reject the model depending on the uncertainty bounds and the intended application. The model uncertainty could also be used to predict the true minimal height interval as

$$\begin{aligned} h = y^m &= f(\vec{x}, \vec{w}) + \epsilon^m \\ &= \frac{5}{2}(10) - \frac{1}{2}\frac{3^2}{9.81} + [0.1406, 1.3357] \\ &= 24.5413 + [0.1406, 1.3357] = [24.6819, 25.877]\,\text{m} \end{aligned} \tag{76}$$

Considering model uncertainty in the original function shows that the actual minimum height falls within the above interval. In order to create a safe design, the model builder can choose the worst case value which is $h = 25.877m$. Depending upon the intended application, the choice of value will differ as aerospace applications can use mean value in order to reduce cost.

3.5. VERIFICATION OF CONFIDENCE LEVEL

This section shall implement MCS on the roller coaster model in order to verify the confidence level of the uncertainty interval obtained. The process is summarized as follows:

- Generate N_s samples for input variables $\vec{x} = (v, r)$ using their means (3 m/s, 10 m) and standard deviations (0.02 m/s, 0.002 m) from the experimental results. The samples are denoted by $\vec{x}_i = (v_i, r_i) \quad \text{where} \quad (i = 1, 2, ..., N_s)$.

- Generate N_s samples for output variable $y^e = h^e$ using its mean 25.2794 and standard deviation $\frac{0.5864}{2}$. The division factor 2 is derived from the coverage factor k which is according to ASME standards for 95% confidence. The samples are denoted by $h_i^e \quad \text{where} \quad (i = 1, 2, ..., N_s)$.

- Calculate the simulation model at each of the input variable sample which can be represented as $h_i^m = f(\vec{x}_i) = f(v_i, r_i)$, $(i = 1, 2, ..., N_s)$.

- The difference between experimental results and simulation model predictions are calculated with $U_i^m = h_i^e - h_i^m, (i = 1, 2, ..., N_s)$.

- Evalaute the probability of U_i^m lying within the calculated uncertainty interval model $[\underline{U}^m, \overline{U}^m] = [0.1406, 1.3357]$. Let the number be denoted by N_c.

- Now calculate the confidence by $\frac{N_c}{N_s}$. Using a sample size of $N_s = 10^5$, a confidence of 95.5% is obtained which proves the fact that the proposed methodology calculates the uncertainty interval with a confidence level of at least 95%, same as achieved during the experiments.

4. THERMAL CHALLENGE PROBLEM

The new model uncertainty quantification method was applied to the roller coaster problem in the previous chapter. The problem was selected due to its simplicity and ease of understanding. This section primarily focuses on applying the methodology to the thermal challenge problem devised by the Sandia National Laboratory. It involves validating a model for one-dimensional, linear heat conduction in a solid slab with heat flux boundary conditions. The thermal problem posed at the Sandia Validation Challenge workshop was purposefully designed with some model weakness in order to depict the real world problems. The uncertainty quantification was performed on the model for a selected set of experimental data from the work of Dowding et al. [1].

4.1. DESCRIPTION OF THERMAL PROBLEM

The thermal problem consists of a mathematical model of the temperature response for one-dimensional heat conduction through a slab with three sets of experimental data which differ in size ("low","medium","high") to asses the model and predict regulatory performance relative to a regulatory requirement which is in terms of probability that a surface temperature should not exceed a specified temperature at regulatory conditions.

A slab of thickness L is exposed to environment with boundary conditions flux q on $x = 0$ face and adiabatic on $x = L$ face as shown in Figure 4.1. The analytical solution for the temperature in the body (for $t > 0$) can be written as

$$T(x,t) = T_i + \frac{qL}{k}\left[\frac{(k/\rho C_p)t}{L^2} + \frac{1}{3} - \frac{x}{L} + \frac{1}{2}\left(\frac{x}{L}\right)^2 - \frac{2}{\pi^2}\sum_{n=1}^{6}\frac{1}{n^2}e^{-n^2\pi^2\frac{(k/\rho C_p)t}{L^2}}\cos\left(n\pi\frac{x}{L}\right)\right] \tag{77}$$

where T is the temperature, x is the location within the slab, t is the time elapsed,

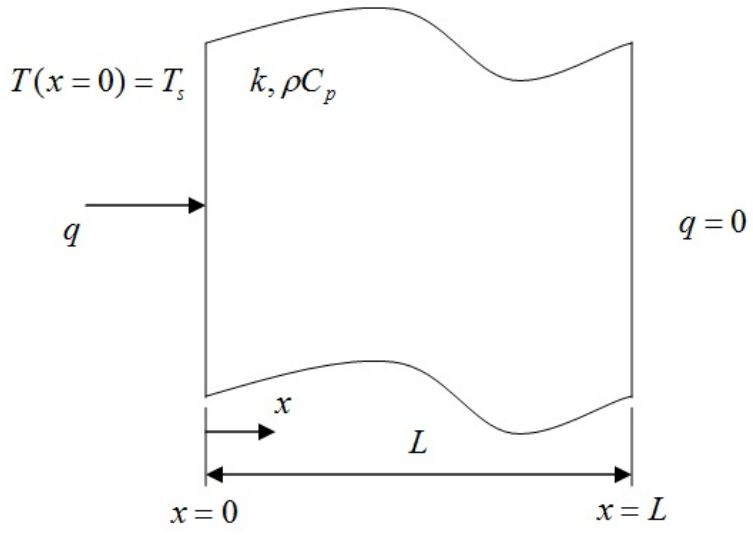

Figure 4.1. Heat conduction problem [1]

T_i is the initial temperature, q is the heat flux, L is the thickness of the material, k and ρC_p are the thermal properties. The regulatory requirement is given by

$$Pr\{T_{x=0\,cm,t=1000\,s,T_i=25°C,q=3500\,W/m^2,L=1.90\,cm} > 900°C\} < 0.01 \tag{78}$$

Relativity of the terms in Equation (77) can be expressed as follows

- The temperature T at a particular location x at a given time t is the model output y, $y^m = T$.
- As per the definition in Section 2, the model parameter $\vec{w}$ is the location within the slab x, time elapsed t, thickness of slab L and the heat flux q which were not changed during the experiment, $\vec{w} = (x, t, L, q)$.
- The model input variables $\vec{x}$ are the thermal conductivity k and the volumetric heat capacity ρC_p, $\vec{x} = (k, \rho C_p)$.

4.2. EXPERIMENTATION FOR THERMAL PROBLEM

In order to simulate an experiment and assess the model, a series of experiments ("Material characterization","Ensemble validation"and "Accreditation") were carried out by Dowding et al. [1] with different conditions and different sample sizes.

Material characterization is subjecting the specimens, which are produced with nominal dimensions representative of the intended application, over the temperature range to study the behaviour of material properties like thermal conductivity k and volumetric heat capacity ρC_p. It provides an estimate of k and ρC_p for a randomly selected specimen at a given temperature. As these experiments were carried out at different levels ("low","medium"and "high"), choose the one which provides with a good estimate of these properties. The number of readings for "low","medium"and "high"levels were 6, 20 and 30 respectively. Consequently, larger sample size provides with the best estimate and, hence select the data pertaining to "high"level of experiments.

Experimental configuration for the model parameters, $\vec{w} = (x, L, q, t)$ is explained in Table 4.1 as follows:

Table 4.1. Configuration of model parameter for experiments

Model Parameter	Value
x	$0\,cm$
L	$2.54\,cm$
q	$2000\,W/m^2$
t	$1000\,sec$

Using the experimental data in Table 4.2, calculate the mean $\tilde{k}$ and standard deviation u_k for thermal conductivity k. Choosing a coverage factor of 2 for 95% confidence, construct an interval for the random input variable k by following the

Table 4.2. Thermal Conductivity, $k(W/m^{\circ}C)$

$k(20^{\circ}C)$	$k(250^{\circ}C)$	$k(500^{\circ}C)$	$k(750^{\circ}C)$	$k(1000^{\circ}C)$
0.0496	0.0628	0.0602	0.0657	0.0631
0.0530	0.0620	0.0546	0.0713	0.0796
0.0493	0.0537	0.0638	0.0694	0.0692
0.0455	0.0561	0.0614	0.0732	0.0739
0.0483	0.0563	0.0643	0.0684	0.0806
0.0490	0.0622	0.0714	0.0662	0.0811

measurement uncertainty guidelines. The experimental output will be in the form of

$$k^e = \tilde{k} \pm 2u_k \tag{79}$$

This serves as the experimental data to simulate an experiment for random variable k. Similarly, the experimental results for volumetric heat capacity are shown in Table 4.3 which in turn gives the interval for ρC_p as shown in Equation (80) below.

Table 4.3. Volumetric Heat Capacity, $\rho C_p(J/m^{3^{\circ}}C) \times 10^5$

$k(20^{\circ}C)$	$k(250^{\circ}C)$	$k(500^{\circ}C)$	$k(750^{\circ}C)$	$k(1000^{\circ}C)$
3.76	3.87	4.52	4.68	4.19
3.38	4.69	4.10	4.24	4.38
3.50	4.19	4.02	3.72	3.45
4.13	4.28	3.94	3.46	3.95
4.02	3.37	3.73	4.07	3.78
3.53	3.77	3.69	3.99	3.77

$$\rho C_p{}^e = \rho\tilde{C}_p \pm 2u_{\rho C_p} \tag{80}$$

The calculated intervals for both the input variables along with their means and standard deviations are shown in Table 4.4.

Table 4.4. Experimental results for input variables k and ρC_p

Input Variable	Mean	Standard deviation	95% CI
k	0.0628	0.0099	[0.043, 0.0827]
ρC_p	393900	36251	[321400, 466400]

Along with the input variable experimentation, the model builder also requests to perform experiments on the output variable. The temperature model is tested for the same experimental configuration as mentioned in Table 4.1. The confidence interval for the system response T is given in Table 4.5.

Table 4.5. Experimental results for the output variable $T(^{\circ}C)$

Output Variable	Interval Bound
T	[475.6, 517.6]

Based on these experimental results, the methodoloy to quantify the uncertainty in temperature model to better assist the model builders has been put forward. Detailed information about the experiments and the procedure followed is explained by Dowding et al. [1]. Re-iterating the same here will be beyond the scope of this thesis.

4.3. SIMULATION ON THERMAL PROBLEM

Before checking whether the model in Equation (77) satisfies the regulatory requirement of Equation (78), it is necessary to quantify the model uncertainty so that the accuracy of temperature model can be gauged and a true value of temperature in the body can be estimated.

The model builder is provided with the 95% confidence interval by the experimenter as shown in Tables 4.4 and 4.5. As the function is highly non-linear, simply

plugging the lower and upper bounds into the model will result in erroneous conclusions. This was not the case for the roller coaster problem in Section 3 because the model was simple and linear in nature. The confidence level achieved in that case was more than or equal to the confidence level in experimentation. In order to quantify uncertainty for the thermal problem, the methods described in Section 2.4 are being used.

4.3.1. Optimization on Thermal Problem. The basic requirement is to find the lower and upper bounds (global maxima or minima) for the temperature model given that the input variables $\vec{x} = (k, \rho C_p)$ lie within a certain domain. Mathematically, it can be represented as

The model output interval is

$$T^m = [\underline{T}^m, \overline{T}^m] \tag{81}$$

where the optimization model for $\underline{T}^m$ is given by

$$\left.\begin{aligned} \underline{T}^m &= \min f(\vec{x}) \\ &\text{subject to} \\ \vec{x} &\in [\underline{\vec{x}}^e, \overline{\vec{x}}^e] \end{aligned}\right\} \tag{82}$$

Similarly, the optimization model for $\overline{T}^m$ is given by

$$\left.\begin{aligned} \overline{T}^m &= \max f(\vec{x}) \\ &\text{subject to} \\ \vec{x} &\in [\underline{\vec{x}}^e, \overline{\vec{x}}^e] \end{aligned}\right\} \tag{83}$$

where T^m represents the simulation output and $f(\vec{x})$ is the function in terms of input variables. A matlab code is furnished in order to solve the optimization problem. The

built-in function *fmincon* [48] is used to find the minima of the objective function. Furthermore, the same function is used to maximize the model with the change that the objective function is negative. The starting point for the input variables in minimization and maximization codes was fed as the mean values from the Table 4.4. The code is attached in Appendix B. The lower and upper bounds for the objective function through optimization are tabulated in Table 4.6.

Table 4.6. Interval bounds for temperature through Optimization

Output Variable	Interval Bound
T	[424.71341, 573.47430]

4.3.2. First Order Taylor Series on Thermal Problem. There are several appoximation methods which are used to simplify computer evaluations in order to make them computationally inexpensive. These methods are devised to reduce the non-linearity of the objective functions and arrive at the best possible estimate without sacrifying much on accuracy. One of the examples is first order Taylor series [35] which uses Taylor series expansion to approximate the objective function. The basic idea is to linearize the model by using first order derivatives. Furthermore, after linearizing the model, find the mean and standard deviation of the objective function. As per ASME, using a coverage factor of 2, find the upper and lower bounds for the objective function. Mathematically, if y is the model output and $\vec{x} = (k, \rho C_p)$ is the vector of input variables,

$$\tilde{T}(\vec{x}) = T(\tilde{\vec{x}}) \tag{84}$$

where $\tilde{T}$ is the mean of the objective function and $\tilde{\vec{x}}$ represents the vector of mean

values of input variables.

$$\sigma_T = \sqrt{\left(\frac{\partial T}{\partial k}\sigma_k\right)^2 + \left(\frac{\partial T}{\partial \rho C_p}\sigma_{\rho C_p}\right)^2} \tag{85}$$

where

$$\begin{aligned} \sigma_T &= \text{ standard deviation of the objective function } T \\ \sigma_k &= \text{ standard deviation of input variable } k \\ \sigma_{\rho C_p} &= \text{ standard deviation of input variable } \rho C_p \end{aligned}$$

Once the mean and standard deviation of the objective function are obtained, lower and upper bounds can be calculated as per Equation (86) using ASME standards of measurement uncertainty. Refer code in Appendix B.

$$\tilde{T} \pm 2\sigma_T \tag{86}$$

The results obtained as per the first order method are shown in Table 4.7.

Table 4.7. Interval bounds for temperature through First Order Taylor Series

Output Variable	Interval Bound
T	[398.23716, 561.94877]

4.3.3. Worst Case Analysis (WCA). WCA method uses the first order approximation as mentioned above in order to create a response surface for the non-linear function. The analysis results for the thermal challenge problem are shown in the Table 4.8.

Table 4.8. Interval bounds for temperature through Worst Case Analysis

Output Variable	Interval Bound
T	[367.36160, 592.82433]

However, due to the non-linearity of the objective function, linearization would not be an accurate estimate of the true value. Hence, approxiamte the function using higher order series as explained in the next section.

4.3.4. Second Order Taylor Series on Thermal Problem. The approximation for thermal challenge problem is given by

$$T(\vec{x}) \approx T(\tilde{\vec{x}}) + \bigtriangledown(\tilde{\vec{x}})(\vec{x} - \tilde{\vec{x}})^T + \frac{1}{2}(\vec{x} - \tilde{\vec{x}})H(\tilde{\vec{x}})(\vec{x} - \tilde{\vec{x}})^T \tag{87}$$

where $H(\tilde{\vec{x}})$ is the Hessian matrix at mean values of $\vec{x}$. It is given by

$$\begin{bmatrix} \frac{\partial^2 T}{\partial k^2} & \frac{\partial^2 y}{\partial k \rho C_p} \\ \frac{\partial^2 T}{\partial \rho C_p k} & \frac{\partial^2 T}{\partial \rho {C_p}^2} \end{bmatrix} \tag{88}$$

Equation (87) is utilized to obtain a response surface which was further utilized for the prediction of lower and upper bounds of the model for uncertainty interval. The Hessian matrix is of (2×2) size as there are two input variables. If there are more variables, the generalized hessian matirx can be used as shown in Equation (29). The results procured from the second order approximation model calculated in Matlab (code attached in Appendix B) are as shown in Table 4.9 below

Table 4.9. Interval bounds for temperature through Second Order Taylor Series Method

Output Variable	Interval Bound
T	[431.01541, 543.74677]

4.3.5. GSA on Thermal Problem. The algorithm explained in section 2 has been coded in Matlab and the code is attached in Appendix A. Results obtained are as shown in Table 4.10.

Table 4.10. Interval bounds for temperature through Golden Search Algorithm

Output Variable	Interval Bound
T	[390.31981, 632.85195]

4.3.6. MCS on Second Order Approximation for Temperature Model. In thermal example, 10^7 samples were generated to find the global maxima or minima for the objective function using the Matlab code attached in Appendix B. By predicting temperature at each of the input values, upper and lower bounds for the temperature in the slab at a particular location T^m as $[\underline{T}^m, \overline{T}^m]$ can be tabulated as shown in Table 4.11.

Table 4.11. MCS on Second Order Approximation Model

Output Variable	Interval Bound
T	[396.74466, 621.56256]

As can be seen from Figure 4.2, uniform distributions have been assumed for both the random variables k (left) and ρC_p (right). No matter what the distribution of the random variables is, the output variable (T) will closely represent *normal* distribution due to the large sample database. This can be clearly seen from the histogram for the output variable T in Figure 4.3. A comparison of cumulative distribution functions (*cdf*) of output variable for uniform and normal distributions of random variables is shown in the Figure 4.4 below. For both the cases, the output variable T resembles a *normal* cumulative distribution function (CDF).

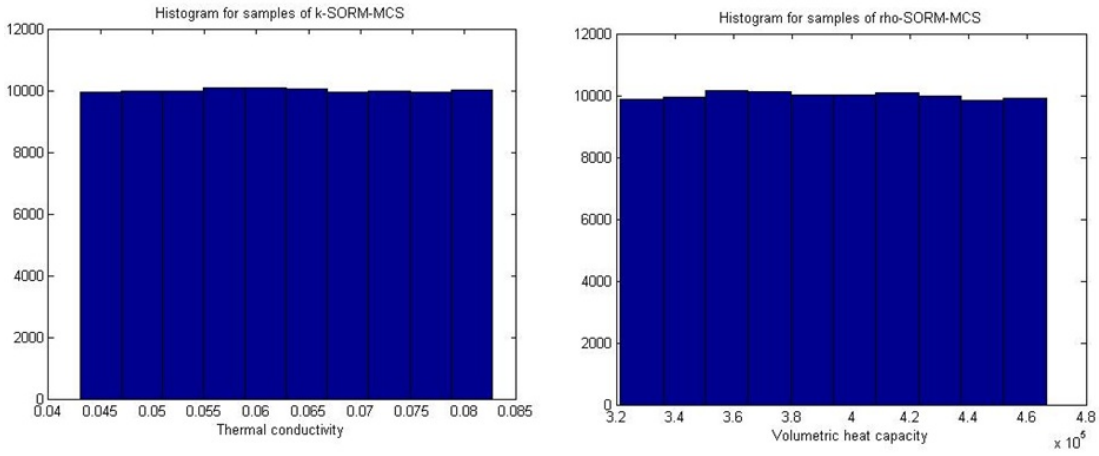

Figure 4.2. Uniform distribution of random variables for second order approximation

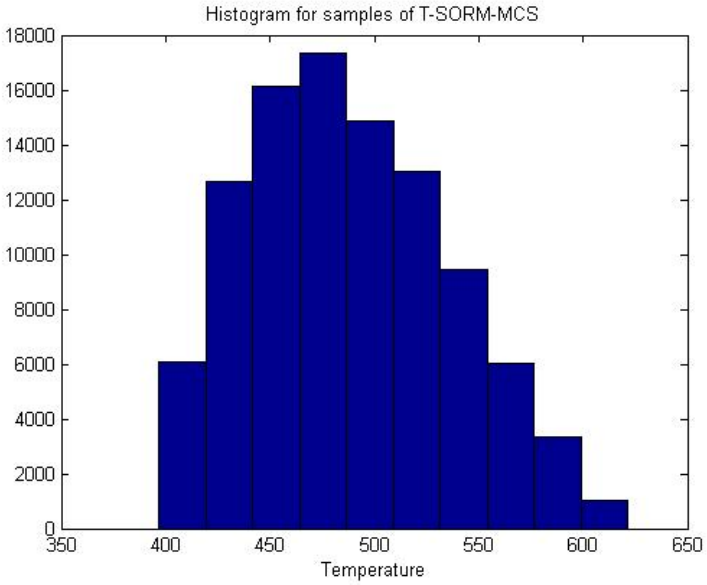

Figure 4.3. Distribution of the system response variable - Temperature $T(^{\circ}C)$

4.3.7. MCS on the Original Temperature Model. MCS is carried out on the original model in the same procedure as explained above. This will give the most accurate prediction for the system output response y as the original non-linear function is used for evaluation purpose. However, it will be computationaaly expensive due to the complexity of the problem. This exercise was basically done to evaluate the accuracy of the different methods applied to solve the problem and

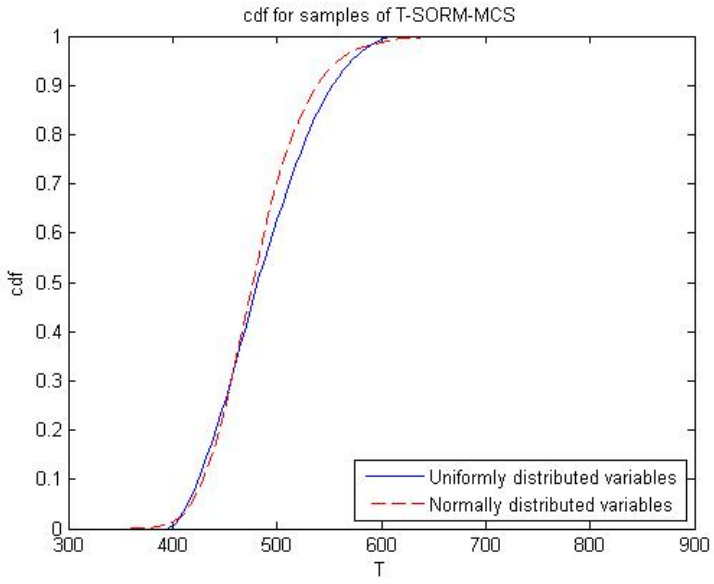

Figure 4.4. Cumulative distribution function (*cdf*) for Temperature $T(^{\circ}C)$

moreover, MCS covers the maximum range thereby increasing the probability of true model uncertainty lying between the lower and upper bounds of the model uncertainty confidence interval. A Matlab code was generated to perform the simulation on the original model and is attached in Appendix B. The results are shown below in Table 4.12

Table 4.12. Monte Carlo simulation on Original Model

Output Variable	Interval Bound
T	[390.40231, 632.70970]

A similar trend for the output variable T is observed on implementation of MCS on the original non-linear model. The data for temperature T is approxiamtely normally distributed even though the input variables are assumed to have uniform distribution as shown in Figures 4.5 and 4.6 respectively. Graphical representation of these intervals in Figure 4.7 will give a clear idea about the accuracy and the

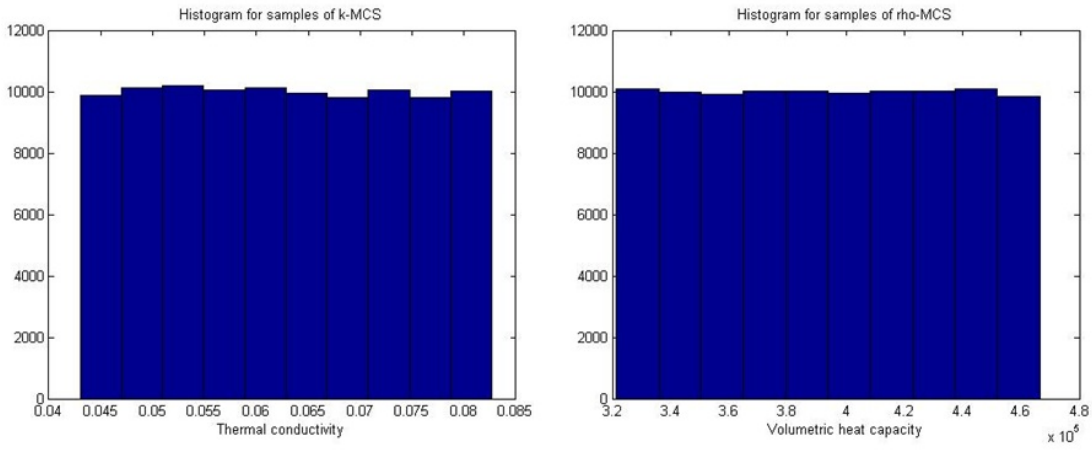

Figure 4.5. Uniform distribution of random variables k and ρC_p for original model

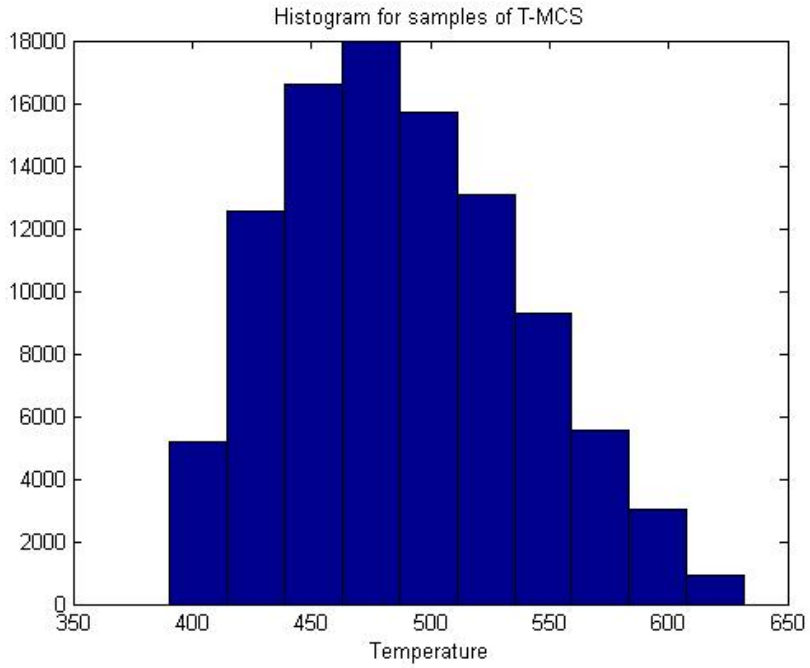

Figure 4.6. Distribution of the system response variable - Temperature $T(^{\circ}C)$

worst case scenario that can be considered for the model. The decision lies with the model builder whether to optimize or take the worst case into consideration. This will mainly depend on the intended application.

In order to summarize the prediction results by different methods and to compare them with each other, the following Table 4.13 will give a good idea.

Table 4.13. Summary of interval bounds for Temperature $T(^{\circ}C)$

Method adopted	Interval Bound for Temperature	Number of function evaluations
Optimization	[424.71341, 573.47430]	6
First Order Taylor Series	[398.23716, 561.94877]	1
Worst Case Analysis	[367.36160, 592.82433]	1
Second Order Taylor Series	[431.01541, 543.74677]	1
MCS on second order approximation	[396.74466, 621.56256]	10^7
Golden Search Algorithm	[390.31981, 632.85195]	13
MCS on original model	[390.40231, 632.70970]	10^7

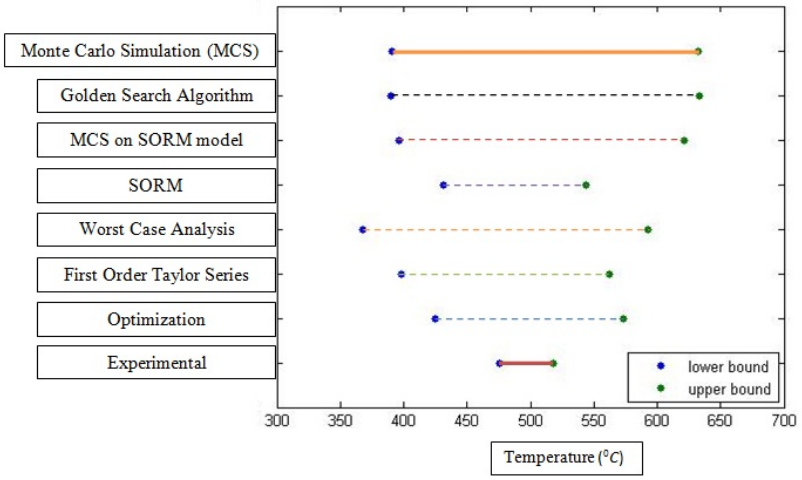

Figure 4.7. Comparison of different model prediction methods for Temperature $T(^{\circ}C)$

It is clear Figure 4.7 that second order approximates the model better than the first order series expansion but do take note of the fact that it might not give us the global minima or maxima for the obejective function. Monte Carlo simulation on original model gives the best estimate of lower and upper bounds for temperature provided that the input random variables are uncertain. To reduce the computational

effort, the MCS on second order approximation model has been carried out and its accuracy is acceptable. Also, higher order cubic polynomial used in the Golden Search Algorithm for approximating the objective function gives almost similar results to MCS on original model. Considering function evaluation as a criteria, GSA provides with the best estimate of global minimum and maximum in 13 iterations whereas MCS is computationally expensive.

4.4. MODEL UNCERTAINTY QUANTIFICATION

After going through a series of methods to find the lower and upper bounds of the model problem provided with the uncertainty in the input variables, quantify the model uncertainty using Equation (38). Interval analysis, as explained in Section 2, is implemented between the experimental and prediction results so that an interval for the uncertainty known as *model uncertainty interval* can be derived. The results obtained on interval analysis are shown in Table 4.14 below

Table 4.14. Uncertainty interval results of temperature through different methods

Method	Interval Bound for Model Uncertainty
Optimization	[-97.87430, 92.88659]
First Order Taylor Series	[-86.34877, 119.36284]
Second Order Taylor Series	[-68.14677, 86.58460]
Worst Case Analysis	[-117.22433, 150.23840]
MCS on second order approximation	[-145.96256, 120.85534]
Golden Search Algorithm	[-157.25195, 127.28019]
MCS on original model	[-157.10969, 127.19768]

4.5. VERIFICATION OF CONFIDENCE LEVEL

In the previous section, the confidence level for the evaluated model uncertainty interval for a *simpler model* is more than or equal to the confidence level achieved during experimentation. However, for the thermal challenge problem, sampling methods like Monte Carlo Simulation are implemented to check or validate the confidence

level achieved in the interval mainly due to the non-linear nature of the model. The concept of *truncated distribution* is useful which can be termed as a conditional distribution due to the fact that it results from restricting other probability distributions either above or below their threshold values. A pictorial representation in Figure 4.8 shows the interval bounds for different confidence levels.

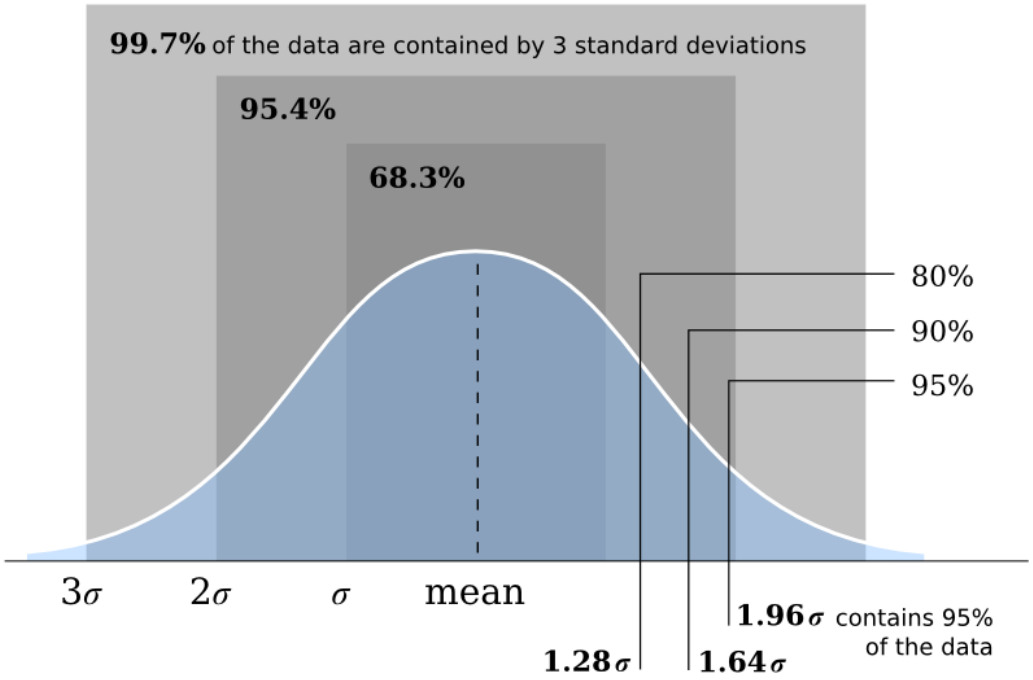

Figure 4.8. Confidence intervals for a normal distribution [2]

As the original distribution is modified, the mean μ for the new distribution will remain the same but standard deviation σ is modified. Dr. David Olive [3] from Southern Illinois University studied this concept. His work mentioned about a corollary pertaining to truncated normal distribution.

Let Y be $TN(\mu, \sigma^2, a = \mu - k\sigma, b = \mu + k\sigma)$. Then,

$$E(Y) = \mu \quad \text{and} \quad VAR(Y) = \sigma^2 \left[1 - \frac{2k\phi(k)}{2\Phi(k) - 1} \right]$$

where TN denotes truncated normal distribution and a & b are the lower and upper bounds of the truncated distribution respectively and k is the coverage factor as defined in Section 2. Using the above relation, variances for different truncated distributions and confidence levels can be computed according to the k values as tabulated in Table 4.15.

Table 4.15. Variances for different truncated normal distriutions [3]

k	$VAR(Y)$
2.0	$0.774\sigma^2$
2.5	$0.911\sigma^2$
3.0	$0.973\sigma^2$
3.5	$0.994\sigma^2$
4.0	$0.999\sigma^2$

Now, validate the confidence level and the results achieved in Table 4.14 for the model uncertainty interval U^m. A three stage Monte Carlo Simulation [49] has been used and the steps are displayed on the flow chart Figure 4.9.

In the first stage of monte carlo simulation, simulate the model in order to compute model predictions from the input variable reported by the experimenter. Next, apply the concept of truncated distribution to choose 95% confidence interval from the entire distribution. At the end of step 1, N samples of temperature in the slab denoted as $y^m = T^m$ are procured. A sample size of $N = 10^5$ is used.

In the subsequent MCS, generate $N = 10^5$ samples from the original sample database of the input variables as reported by the experimenter. Let us denote the temperature samples by $y^e = T^e$. Furthermmore, evaluate the difference between the two samples to represent the uncertainty, $U^m = y^e - y^m$. Count the number of uncertainty values lying within the uncertainty interval derived using different methods, denote it by N_{con}. Evaluate the confidence from $\frac{N_{con}}{N}$. The confidence level achieved in different mehtods is shown in Table 4.16 below A Matlab code has been generated

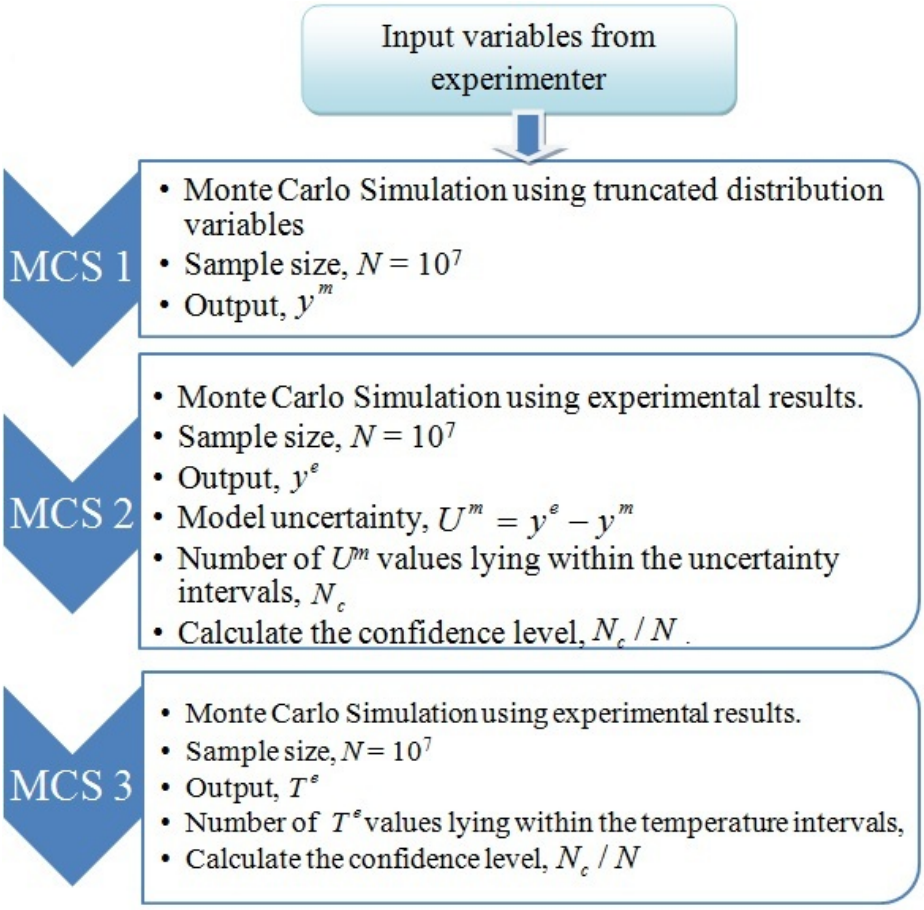

Figure 4.9. Flow chart for confidence level in uncertainty interval

Table 4.16. Confidence level achieved in uncertainty intervals

Method adopted	Confidence level (%)
Golden search algorithm	99.889
MCS	99.889
SORM (MCS)	99.762
Optimization	97.568
SORM approximation	93.975

in order to perform the monte carlo simulations and also evaluate the probability as attached in Appendix C. From Table 4.16, it is clear that Golden search algorithm and Monte Carlo simulation achieve equivalent confidence in the uncertainty interval, as high as 99.89%. Gradually, the confidence decreases through the methods of MCS on SORM, Optimization and SORM approximation as 99.76%, 97.57% and 93.98% re-

spectively. This is mainly due to the fact that the uncertainty interval range reduces for the methods in the same sequence.

Moreover, the third Monte Carlo simulation can be explained as follows

Given:

$$\left.\begin{aligned} 95\%\,[\vec{x}_l, \vec{x}_u] \\ g(\vec{x}) \end{aligned}\right\} \text{By the experimenter}$$

where

$$\vec{x} = \text{ Model input variable vector}$$

$$g(\vec{x}) = \text{ System response}$$

Analysis:

$$\left.\begin{aligned} g_l = \min g([\vec{x}_l, \vec{x}_u]) \\ g_u = \max g([\vec{x}_l, \vec{x}_u]) \end{aligned}\right\} \text{By the model builder}$$

Question:

$$\text{Is confidence level in } [g_l, g_u] \geq 95\%?$$

$$\therefore \text{Is } Pr\{g \in [g_l, g_u]\} \geq 95\%? \tag{89}$$

On implementation of the third MCS, the probability in Equation (89) was found to be 99.88%. Thus, the proposed methodology has sufficient level of confidence as proved by the justification in Equation (89). The next section will conclude the dissertation with the salient features of the model uncertainty quantification process.

5. CONCLUSIONS AND FUTURE WORK

In the introduction section, we discussed about the importance of quantifying model uncertainty in order to estimate its accuracy and agreement with its real world application. Keeping this in mind, a different methodology was devised which was easy to implement to quantify the model error for simulation models. In this final section, the thesis shall be concluded by describing the progress made towards this goal in terms of uncertainty quantification and its application on complex problems. Along with the advantages, drawbacks shall also be discussed which helped to put forward some future research directions in order to overcome them.

5.1. CONCLUSIONS

The focus of this research was to develop an efficient approach for model uncertainty quantification when the model builders are incapable of performing experiments for validation. As a result, the experimentation part is outsourced to professional laboratories. There was a need to bridge this gap between the model builders and experimenters. Also, one of the objectives of this study was to simplify decision making for the model builder by altering the uncertainty result representation from the traditional methods to a more interpretable form. For highly non-linear problems, approximation methods can be used to create a response surface so as to represent the original model.

The salient features of the proposed methodology are as follows:

- ASME measurement uncertainty standards proved to be a useful tool wherein the experiments are carried out according to the standards and the results are reported to the model builder in terms of *expanded uncertainty* as mentioned in Section 2. The experimenter carries out experiment not only on the output

variable but also on the input random variables at the worst case points requested by the model builder. Overall, this has resulted in reduced cost for the model builders as the experimental part is relatively expensive.

- Implementation of interval analysis in order to compare the observation results and model prediction.

- Other main feature of the proposed methodology is that model uncertainty is quantified in terms of interval range with at least 95% confidence level. This makes interpretation of results far more easier than the traditional probabilistic distributions.

- The proposed method is also applicable in conditions where the model builder is even not able to perform the simulation. In this case, he can decide to outsource the simulation part along with the experimental work and henceforth use the methodology.

Major findings from the study can be enumerated as follows:

1. In Section 3, the proposed methodology is implemented on a roller coaster design problem with two input random variables, velocity v and loop radius r. Since the model is simple in nature, the input variable intervals from experimentation are directly substituted into the model. On comparison of the observation results with the model prediction, the model uncertainty was found to be within $[0.1406, 1.3357]m$. Inference can be made from the positive sign of interval values that the original model underestimates the minimal height required for the car to travel the loop trajectory without going off track. The most important thing to note is that the confidence in model uncertainty interval was at least 95% according to Section 2.6. The decision of the model being rejected or accepted is with the model builder. It is easier for him to determine the risk in design if the

uncertainty is in the form of an interval. Different applications have different requirements to be met and hence the model builder can choose accordingly.

2. In Section 4, a highly non-linear model like the thermal challenge problem by the Sandia National Laboratories has been examined for the model uncertainty interval according to the proposed method. In order to reduce the computational effort, a response surface has been generated using different approximation methods like *First order taylor Series* and *Second order Reliability method*. SORM being the higher order approximation was used to run *Monte Carlo Simulation* for 10^7 random samples to evaluate the lower and upper bounds for temperature. The results were a good match with the *MCS* on the original model. However, *Golden Search Algorithm* which is an interval based method, provided with exact results so as to match with the *MCS*.

3. From the results obtained, the temperature model has an uncertainty within the interval $[-157.10969, 127.19768]$ which clears the point that it does not meet the regulatory requirement mentioned in Equation (78). The result is confirmed with the previous work done in this field by Ferson et al. [23] stating that, "the temperatures will exceed the probability specified by the regulatory requirement by a factor of 22". Most important part of this analysis is to gauge the confidence level associated with the uncertainty intervals. This is achieved by using three stage MCS, on truncated distribution of input variables as shown in Section 4.5. Observe from the result Table 4.16, that different methods provide varying confidence levels for the uncertainty interval with the GSA and MCS methods being the highest with 99.89%. The confidence level in other methods goes on decreasing gradually due to their smaller range of intervals. However, the decision lies with the model builder to choose the method best suited for the intended application. Furthermore, the confidence level of the temperature

value from model prediction lying within the 95% confidence interval for system response reported by the experimenter is 99.88%.

In overall summary, both of the examples in Section 3 and Section 4, imply that model uncertainty exists when there is uncertainty in model input variables and the proposed uncertainty quantification method is effective in determining the same. An interval output for the system response variable gives a clear picture of the variation in the model. The process is much more simpler than the traditional methods and easy to understand which makes the model builder's goal achievable even due to their inability to perform the experiment.

5.2. SUGGESTIONS FOR FUTURE WORK

As mentioned earlier, the drawbacks of this method or the concepts that were not considered during the analysis will provide us with further research directions.

One direction is to consider the correlation between the input random variables. It would be interesting to see the effect of correlation between input variables and their dependence on the output. For example, if one considers the temperature dependence of input variables k and ρC_p, model predictions are bound to be different and thereby different model uncertainty bounds. Will the uncertainty interval method be still valid for such a case? Will the confidence level achieved with the revised methodology be as high as the confidence level maintained during experimentation? Further research is needed to answer all of these questions.

Another possibility would be to integrate uncertainty interval model methodology with robust design so that even more accurate predictions can be achieved for model uncertainty bounds. As per the proposed methodology, the worst case design points are considered in order to analyze model uncertainty. In aerospace applications, where the design should be cost efficient and feasible, robust design optimizes the model uncertainty interval. Use of higher order approximation methods such as

Non-Intrusive Polynomial Chaos (NIPC) [50, 51] derived from the polynomial chaos theory based on spectral representation of uncertainty will help to improve accuracy. One more advantage of implementation of NIPC is that it can even take care of mixed (aleatory and epistemic) uncertainty present in the model.

Finally, performing the sensitivity analysis [45, 52] (both linear and non-linear) on each of the input random variables would allow us to rank their relative importance. This further leads to global sensitivity analysis to determine the correlation between input and output variables. A particular interest would be to use this methodology to study more complex models solving real world problems and improving on it successively.

APPENDIX A

Optimization through Golden Search Algorithm: MATLAB Source Code

```
%Program to optimize Temperature model by Golden search algorithm (GSA)

clear all; clc; format long;

%% Part(a) - Define variables and constants
syms rho_Cp x

r = (-1+sqrt(5))/2;      % Golden search ratio
T_ini = 25;              % Initial temperature (C)
k = 0.043;               % Thermal conductivity (W/mC)
q = 2000;                % Heat flux (W/m2)
L = 0.0254;              % Thickness (m)
t = 1000; p = 0;         % Time and location

% Temperature model
term = 0;
for n = 1:6
term = term + (1/n^2)*(exp(-n^2*pi^2*(((k/rho_Cp)*t)/L^2)))...
    *cos(n*pi*p/L);
end

T = T_ini + (q*L/k)*( (((k/rho_Cp)*t)/L^2) + (1/3) - (p/L) +...
    (0.5*(p/L^2)) - ((2/pi^2)*term) );

%% Part(b) - Golden search Algorithm (GSA)
% Starting interval, stopper and relative tolerance
a = 321400; b = 466400; flag = 0; epsilon = 0.000001;

% New interval points chosen by using the golden ratio
c = a + (1-r)*(b-a); d = a + (b-a)*r;

% Function evaluations
T_a = subs(T,rho_Cp,a); T_b = subs(T,rho_Cp,b);
T_c = subs(T,rho_Cp,c); T_d = subs(T,rho_Cp,d);

% To find number of iterations needed for GSA
N = ceil(-2.078*log(epsilon));

% Application of Golden search algorithm
for i = 1:N
    if (T_c <= T_d)
        a = c; T_a = T_c;
        c = d; T_c = T_d;
        d = a + (b-a)*r;
        flag = 1;
    else
        b = d; T_b = T_d;
        d = c; T_d = T_c;
        c = a + (1-r)*(b-a);
    end
    T_c = subs(T,rho_Cp,c);
    T_d = subs(T,rho_Cp,d);
end
```

```
if (flag == 1)
    T_d = subs(T,rho_Cp,d);
else
    T_c = subs(T,rho_Cp,c);
end

%% Part(c) - Cubic polynomial fit at last iteration of GSA
x1 = a; x2 = c; x3 = d; x4 = b;
f1 = subs(T,rho_Cp,a); f2 = subs(T,rho_Cp,c);
f3 = subs(T,rho_Cp,d); f4 = subs(T,rho_Cp,b);

q1 = x3^3*(x2 - x1) - x2^3*(x3 - x1) + x1^3*(x3 - x2);
q2 = x4^3*(x2 - x1) - x2^3*(x4 - x1) + x1^3*(x4 - x2);
q3 = (x3 - x2)*(x2 - x1)*(x3 - x1);
q4 = (x4 - x2)*(x2 - x1)*(x4 - x1);
q5 = f3*(x2 - x1) - f2*(x3 - x1) + f1*(x3 - x2);
q6 = f4*(x2 - x1) - f2*(x4 - x1) + f1*(x4 - x2);

a3 = (q3*q6 - q4*q5) / (q2*q3 - q1*q4);
a2 = (q5 - a3*q1) / q3;
a1 = ((f2 - f1)/(x2 - x1)) - (a3*((x2^3 - x1^3)/(x2 - x1))) - a2*(x1 + x2);
a0 = f1 - (a1*x1) - (a2*x1^2) - (a3*x1^3);

% Cubic polynomial derived
f = a0 + a1*x + a2*x^2 + a3*x^3;
%% Part(d) - Determine location of minimum or maximum
delta = a2^2 - 3*a1*a3;

if delta < 0
    fprintf('There are no roots to the given equation');
else
    if delta == 0
        fprintf('The function has neither a minima or maxima');
    end
end

% Determination of optimized result
Y1 = (-a2+sqrt(delta))/(3*a3);
Y2 = (-a2-sqrt(delta))/(3*a3);
f_minY1 = (Y1^4) - (Y1^3) - (sin(Y1))^2 + (cos(Y1))^2 + 2;
f_minY2 = (Y2^4) - (Y2^3) - (sin(Y2))^2 + (cos(Y2))^2 + 2;

temp = [f1, f2, f3, f4, f_minY1, f_minY2];
F_min = max(temp);

% Value of k to be changed and re-run the program
k = 0.0827;

% Temperature model
term = 0;
for n = 1:6
term = term + (1/n^2)*(exp(-n^2*pi^2*(((k/rho_Cp)*t)/L^2)))...
    *cos(n*pi*p/L);
end
```

```
T = T_ini + (q*L/k)*( (((k/rho_Cp)*t)/L^2) + (1/3) - (p/L) +...
    (0.5*(p/L^2)) - ((2/pi^2)*term) );  % Temperature Model

%% Part(b) - Golden search Algorithm (GSA)
% Starting interval, stopper and relative tolerance
a = 321400; b = 466400; flag = 0; epsilon = 0.000001;

% Choosing two points by using golden ratio
c = a + (1-r)*(b-a); d = a + (b-a)*r;

% Function evaluations
T_a = subs(T,rho_Cp,a); T_b = subs(T,rho_Cp,b);
T_c = subs(T,rho_Cp,c); T_d = subs(T,rho_Cp,d);

% Number of iterations required for GSA
N = ceil(-2.078*log(epsilon));

% Application of golden search algorithm
for i = 1:N
    if (T_c <= T_d)
        b = d; T_b = T_d;
        d = c; T_d = T_c;
        c = a + (1-r)*(b-a);
        flag = 1;
    else
        a = c; T_a = T_c;
        c = d; T_c = T_d;
        d = a + (b-a)*r;
    end
    T_c = subs(T,rho_Cp,c);
    T_d = subs(T,rho_Cp,d);
end

if (flag == 1)
    T_c = subs(T,rho_Cp,c);
else
    T_d = subs(T,rho_Cp,d);
end

%% Part(c) - Cubic polynomial fit at last iteration of GSA
x1 = a; x2 = c; x3 = d; x4 = b;
f5 = subs(T,rho_Cp,a); f6 = subs(T,rho_Cp,c);
f7 = subs(T,rho_Cp,d); f8 = subs(T,rho_Cp,b);

q1 = x3^3*(x2 - x1) - x2^3*(x3 - x1) + x1^3*(x3 - x2);
q2 = x4^3*(x2 - x1) - x2^3*(x4 - x1) + x1^3*(x4 - x2);
q3 = (x3 - x2)*(x2 - x1)*(x3 - x1);
q4 = (x4 - x2)*(x2 - x1)*(x4 - x1);
q5 = f7*(x2 - x1) - f6*(x3 - x1) + f5*(x3 - x2);
q6 = f8*(x2 - x1) - f6*(x4 - x1) + f5*(x4 - x2);

a3 = (q3*q6 - q4*q5) / (q2*q3 - q1*q4);
a2 = (q5 - a3*q1) / q3;
```

```
a1 = ((f6 - f5)/(x2 - x1)) - (a3*((x2^3 - x1^3)/(x2 - x1)))...
    - a2*(x1 + x2);
a0 = f5 - (a1*x1) - (a2*x1^2) - (a3*x1^3);

f = a0 + a1*x + a2*x^2 + a3*x^3;
%% Part(d) - Determine minima / maxima for obj function
delta = a2^2 - 3*a1*a3;

Y3 = (-a2+sqrt(delta))/(3*a3);
Y4 = (-a2-sqrt(delta))/(3*a3);
f_minY3 = (Y3^4) - (Y3^3) - (sin(Y3))^2 + (cos(Y3))^2 + 2;
f_minY4 = (Y4^4) - (Y4^3) - (sin(Y4))^2 + (cos(Y4))^2 + 2;

temp = [f5, f6, f7, f8, f_minY3, f_minY4];
F_min1 = min(temp);

[F_min1 F_min]
```

APPENDIX B

Thermal Challenge Problem: MATLAB Source Code

```
%% Program to find Error bounds for Thermal problem

            % OUTLINE OF THE PROGRAM
% Part (A): Uncertainty calculations from Experimental values
% Part (B): Minimum value for temperature
% Part (C): Maximum value for temperature
% Part (D): Program to find bounds for Thermal model using FORM
% Part (E): Worst case Analysis
% Part (F): Assuming normal distribution for k & rho_Cp
% Part (G): Approx non linear func using Second order Taylor exp (SORM)
% Part (H): Monte Carlo Method on original function (uniform distribution)
% Part (I): Monte Carlo Method on SORM function (uniform distribution)
% Part (J): Error bounds
% Part (K): Plotting the intervals

clear all; clc; close all;
format short

%% Part (A): Uncertainty calculations from Experimental values
% Experimental values for thermal conductivity
k_delta = [0.0496; 0.0530; 0.0493; 0.0455; 0.0483; 0.0490; 0.0628;...
           0.0620; 0.0537; 0.0561; 0.0563; 0.0622; 0.0602; 0.0546;...
           0.0638; 0.0614; 0.0643; 0.0714; 0.0657; 0.0713; 0.0694;...
           0.0732; 0.0684; 0.0662; 0.0631; 0.0796; 0.0692; 0.0739;...
           0.0806; 0.0811];

% Experimental values for Volumetric heat capacity
rho_Cp_delta = 1e5*[3.76; 3.38; 3.50; 4.13; 4.02; 3.53; 3.87; 4.69;...
                    4.19; 4.28; 3.37; 3.77; 4.52; 4.10; 4.02; 3.94;...
                    3.73; 3.69; 4.68; 4.24; 3.72; 3.46; 4.07; 3.99;...
                    4.19; 4.38; 3.45; 3.95; 3.78; 3.77];

% Mean and Standard deviation of random variables
k_mean = mean(k_delta); k_std = std(k_delta);
rho_Cp_mean = mean(rho_Cp_delta); rho_Cp_std = std(rho_Cp_delta);

% Intervals according to ASME std
k_int = [(k_mean-2*k_std) (k_mean+2*k_std)];
rho_Cp_int = [(rho_Cp_mean-2*rho_Cp_std) (rho_Cp_mean+2*rho_Cp_std)];

% Experimental data for output (temperature)
T_exp = [475.6  517.6];

%% Part (B): Minimum value for Temperature
% Input average values of k and rho_Cp
a0 = [0.0628,393900];

% Initialize variables with the average values
k_min = a0(1); rho_Cp_min = a0(2);

% Initialize upper & lower bounds for k and rho_Cp
lw = [min(k_int), min(rho_Cp_int)];
up = [max(k_int), max(rho_Cp_int)];
```

```
% Use the in-built function fmincon to optimize
option = optimset('display','iter');
a_min = fmincon('objfn_thermal_min_mod',a0,[],[],[],[],lw,up,[],option);

% Initialize the model parameters as per the thermal problem definition
q = 2000; L = 0.0254; T_ini = 25; x = 0; term = 0; t_min = 1000;

% Re-substitute the variables with calculated optimized values
k_min = a_min(1); rho_Cp_min = a_min(2);

% Function evaluation
for n = 1:6
term = term + (1/n^2)*(exp(-n^2*pi^2*(((k_min/rho_Cp_min)*t_min)/...
       L^2)))*cos(n*pi*x/L);
end

% Temperature model (Objective function)
objective_min = (T_ini + (q*L/k_min)*((((k_min/rho_Cp_min)*t_min)...
                /L^2) +(1/3)-(x/L)+(0.5*(x/L^2)) - ((2/pi^2)*term)));

% Display output (min value of temperature)
disp(['the Optimal values of Variables (k, rho_Cp) for T_min ='...
      ,num2str(a_min)]);
fprintf('\n\n\n');
disp(['the Minimum value of func =',num2str(objective_min)]);

%% Part (B): Maximum value for temperature
% Use in-built function fmincon to optimize
option = optimset('display','iter');
a_max = fmincon('objfn_thermal_max_mod',a0,[],[],[],[],lw,up,[],option);

% Re-substitute the variables with calculated optimized values
k_max = a_max(1); rho_Cp_max = a_max(2); term = 0; t_max = 1000;

% Function evaluation
for n = 1:6
term = term + (1/n^2)*(exp(-n^2*pi^2*(((k_max/rho_Cp_max)*t_max)/...
       L^2)))*cos(n*pi*x/L);
end

% Temperature model
objective_max = (T_ini + (q*L/k_max)*((((k_max/rho_Cp_max)*t_max)...
                /L^2) +(1/3) -(x/L)+(0.5*(x/L^2)) - ((2/pi^2)*term)));

% Display output (max value of temperature)
disp(['the Optimal values of Variables (k, rho_Cp) for T_max ='...
      ,num2str(a_max)]);
disp(['the Maximum value of func =',num2str(objective_max)]);

fprintf('\n\nInterval for T_exp = [%f %f]\n',min(T_exp), max(T_exp));

% Interval for Temperature from Optimization
fprintf('\nInterval for T_opt = [%f %f]\n',objective_min, objective_max);
T_model = [objective_min objective_max];
```

```
%% Part (D): Program to find bounds for Thermal model using Taylor series
% Define and initialize variables to be used
syms k rho_Cp; term = 0; t = 1000;

% Model in terms of random variables
for n = 1:6
term = term + (1/n^2)*(exp(-n^2*pi^2*(((k/rho_Cp)*t)/L^2)))...
       *cos(n*pi*x/L);
end

g = ( T_ini + (q*L/k)*( (((k/rho_Cp)*t)/L^2) + (1/3) - (x/L) +...
    (0.5*(x/L^2)) - ((2/pi^2)*term) ) );

% Mean of obejective function considering it to be linear
g_mean = subs(g,[k,rho_Cp],[k_mean,rho_Cp_mean]);

% Differentiating the model with respect to random variables
% Derivatives are commonly known as sensitivities.
dk = diff(g,k); dr = diff(g,rho_Cp);
dgdk = subs(dk,[k,rho_Cp],[k_mean,rho_Cp_mean]);
dgdr = subs(dr,[k,rho_Cp],[k_mean,rho_Cp_mean]);

% Standard deviation of the objective function
g_std = sqrt( (dgdk*k_std)^2 + (dgdr*rho_Cp_std)^2 );

% Display output
fprintf('Interval for T_Taylor = [%f %f]\n',(g_mean-2*g_std),...
       (g_mean+2*g_std));
T_Taylor = [(g_mean-2*g_std) (g_mean+2*g_std)];

%% Part (E): Worst case Analysis
% Determining the co-efficients of the linear model
C0 = g_mean - dgdk*k_mean - dgdr*rho_Cp_mean;

% Maximum value of the function
T_worstmax = C0 + dgdk*min(k_int) + dgdr*min(rho_Cp_int);

% Minimum value of the function
T_worstmin = C0 + dgdk*max(k_int) + dgdr*max(rho_Cp_int);

% Display output
fprintf('Interval for T_WorstCase = [%f %f]\n',T_worstmin, T_worstmax);
T_WorstCase = [T_worstmin T_worstmax];

%% Part (F): Assuming normal distribution for k and rho_Cp
% Determining the number of samples N
N = 1e5; T_sample = zeros(N,1);

% Generation of random samples for random variables
k_sample = normrnd(k_mean,k_std,N,1);
rho_Cp_sample = normrnd(rho_Cp_mean,rho_Cp_std,N,1);

% Function evaluation at each of the sample points
```

```
for i = 1:N
    term = 0;
    for n = 1:6
        term = term + (1/n^2)*( exp(-n^2*pi^2*(((k_sample(i)/...
               rho_Cp_sample(i))*t)/L^2)) )*cos(n*pi*x/L);
    end
    T_sample(i) = ( T_ini + (q*L/k_sample(i))*((((k_sample(i)/...
                  rho_Cp_sample(i))*t)/L^2) + (1/3) - (x/L) +...
                 (0.5*(x/L^2)) - ((2/pi^2)*term) ) );
end

% Plotting histograms to display distribution
hist(k_sample); xlabel('Thermal conductivity');
title('Histogram for samples of k-Norm'); figure
hist(rho_Cp_sample); xlabel('Volumetric heat capacity');
title('Histogram for samples of rho-Norm'); figure
hist(T_sample); xlabel('Temperature');
title('Histogram for samples of T-Norm'); figure

%% Part(G): Approx non linear func using Second order Taylor exp (SORM)
% Determining function in terms of random variables
term = 0;
for n = 1:6
term = term + (1/n^2)*( exp(-n^2*pi^2*(((k/rho_Cp)*t)/L^2)))...
       *cos(n*pi*x/L);
end

T = T_ini + (q*L/k)*( (((k/rho_Cp)*t)/L^2) + (1/3) - (x/L) +...
    (0.5*(x/L^2)) - ((2/pi^2)*term) );

% Function evaluation at the mean point
T_mean = subs(T,[k,rho_Cp],[k_mean,rho_Cp_mean]);

% First & second order derivatives
dtdk = diff(T,k); dtdr = diff(T,rho_Cp);
d2tdkdr = diff(dtdk,rho_Cp); d2tdrdk = diff(dtdr,k);
d2tdk2 = diff(dtdk,k); d2tdr2 = diff(dtdr,rho_Cp);

% Construction of the Hessian matrix
del = [subs(dtdk,[k,rho_Cp],[k_mean,rho_Cp_mean]) subs(dtdr,...
      [k,rho_Cp],[k_mean,rho_Cp_mean])];
H = [subs(d2tdk2,[k,rho_Cp],[k_mean,rho_Cp_mean]) subs(d2tdrdk,...
    [k,rho_Cp],[k_mean,rho_Cp_mean]);...
     subs(d2tdkdr,[k,rho_Cp],[k_mean,rho_Cp_mean]) subs(d2tdr2,...
     [k,rho_Cp],[k_mean,rho_Cp_mean])];

% SORM model
V = [k-k_mean rho_Cp-rho_Cp_mean];
func = T_mean + del*V' + (1/2)*(V*H*V');

% Maximum & minimum values for the function
T_min = double(subs(func,[k,rho_Cp],[(k_mean+k_std),...
        (rho_Cp_mean+rho_Cp_std)]));
T_max = double(subs(func,[k,rho_Cp],[(k_mean-k_std),...
```

```
        (rho_Cp_mean-rho_Cp_std)]));

% Display output
fprintf('Interval for T_SORM = [%f %f]\n',T_min, T_max);
T_SORM = [T_min, T_max];

%% Part (H): MCS on original function (Uniform distribution)
for i = 1:100
% Generating samples for Monte carlo Simulation
    k_sample = unifrnd(min(k_int),max(k_int),N,1);
    rho_Cp_sample = unifrnd(min(rho_Cp_int),max(rho_Cp_int),N,1);

% Function evaluation at these samples
for j = 1:N
term = 0;
for n = 1:6
term = term + (1/n^2)*( exp(-n^2*pi^2*(((k_sample(j)./...
       rho_Cp_sample(j)).*t)./L^2)) )*cos(n*pi*x/L);
end

T_calc1(j) = T_ini + (q*L./k_sample(j)).*( (((k_sample(j)./...
             rho_Cp_sample(j)).*t)./L^2) + (1/3) - (x/L) +...
            (0.5*(x/L^2)) - ((2/pi^2)*term) );
end
Min_T(i) = min(T_calc1); Max_T(i) = max(T_calc1);
end

% Display output
fprintf('Interval for T_MCS = [%f %f]\n',min(Min_T), max(Max_T));
T_MCS = [min(Min_T), max(Max_T)];

% Generaing points for plotting cdf and pdf for uniform distribution
n_point=50;
for i=1:n_point; step=(max(T_calc1)-min(T_calc1))/n_point;
    T_point(i)=min(T_calc1)+(i-1)*step;
end
m=hist(T_calc1,T_point); cdf(1)=m(1);
for i=2:n_point; cdf(i)=m(i)+cdf(i-1); end
cdf=cdf/N; pdf=m/N;

% Generaing points for plotting cdf and pdf for normal distribution
for i=1:n_point; step1=(max(T_sample)-min(T_sample))/n_point;
    T_point1(i)=min(T_sample)+(i-1)*step1;
end
m1=hist(T_sample,T_point1); cdf1(1)=m1(1);
for i=2:n_point; cdf1(i)=m1(i)+cdf1(i-1); end
cdf1=cdf1/N; pdf1=m/N;

% Plotting cdf and pdf using the above points
plot(T_point,cdf); hold on; plot(T_point1,cdf1,'--r');
xlabel('T'); ylabel('cdf'); title('cdf for samples of T-MCS');
legend('Uniformly distributed variables',...
       'Normally distributed variables',4); figure
plot(T_point,pdf); hold on; plot(T_point1,pdf1,'--r');
```

```
xlabel('T'); ylabel('pdf'); title('pdf for samples of T-MCS');
legend('Uniformly distributed variables',...
       'Normally distributed variables');figure

% Plotting histograms for random and output variables
hist(k_sample); xlabel('Thermal conductivity');
title('Histogram for samples of k-MCS'); figure
hist(rho_Cp_sample); xlabel('Volumetric heat capacity');
title('Histogram for samples of rho-MCS'); figure
hist(T_calc1); xlabel('Temperature');
title('Histogram for samples of T-MCS'); figure

%% Part (I): MCS on SORM function (Uniform distribution)
% Generating samples for Monte Carlo Simulation
k_sample = unifrnd(min(k_int),max(k_int),N,1);
rho_Cp_sample = unifrnd(min(rho_Cp_int),max(rho_Cp_int),N,1);

% Function evaluation at these samples
term = 0;
for n = 1:6
term = term + (1/n^2)*( exp(-n^2*pi^2*(((k/rho_Cp)*t)/L^2)) )...
       *cos(n*pi*x/L);
end

T = T_ini + (q*L/k)*( (((k/rho_Cp)*t)/L^2) + (1/3) - (x/L) +...
    (0.5*(x/L^2)) - ((2/pi^2)*term) );

% Mean and Standard deviation
k_mean = mean(k_delta); k_std = std(k_delta);
rho_Cp_mean = mean(rho_Cp_delta); rho_Cp_std = std(rho_Cp_delta);

% Function evaluation at mean point
T_mean = subs(T,[k,rho_Cp],[k_mean,rho_Cp_mean]);

% First and second order derivatives
dtdk = diff(T,k); dtdr = diff(T,rho_Cp);
d2tdkdr = diff(dtdk,rho_Cp); d2tdrdk = diff(dtdr,k);
d2tdk2 = diff(dtdk,k); d2tdr2 = diff(dtdr,rho_Cp);

% Construction of Hessian matrix
del = [subs(dtdk,[k,rho_Cp],[k_mean,rho_Cp_mean]) subs(dtdr,...
      [k,rho_Cp],[k_mean,rho_Cp_mean])];
H = [subs(d2tdk2,[k,rho_Cp],[k_mean,rho_Cp_mean]) subs(d2tdrdk,...
    [k,rho_Cp],[k_mean,rho_Cp_mean]);...
     subs(d2tdkdr,[k,rho_Cp],[k_mean,rho_Cp_mean]) subs(d2tdr2,...
     [k,rho_Cp],[k_mean,rho_Cp_mean])];

% Derived SORM model and function evaluation at sample points
V = [k-k_mean rho_Cp-rho_Cp_mean];
func = T_mean + del*V' + (1/2)*(V*H*V');
T_calc = 1/2*((rho_Cp_sample) - 393900).*(3969562796162817/...
         1152921504606846976*k_sample + 2994134114987257/...
         1208925819614629174706176*rho_Cp_sample - ...
         1125745240091807533352751/944473296573929042739200000) -...
```

```
        5496802930135779/9223372036854775808*(rho_Cp_sample) - ...
        3855333645375677/1099511627776*(k_sample) + 1/2*((k_sample)...
        - 1571/25000).*(3092920434760619/34359738368*k_sample +...
        3969562796162817/1152921504606846976*rho_Cp_sample -...
        631657864458351123937699 9/900719925474099200000) +...
        6738724829706847330610773/72057594037927936000 00;

% Display output
fprintf('Interval for T_SORM_MCS = [%f %f]\n',min(T_calc), max(T_calc))
T_SORM_MCS = [min(T_calc), max(T_calc)];

% Generating points for plotting cdf & pdf
for i=1:n_point; step2=(max(T_calc)-min(T_calc))/n_point;
      T_point2(i)=min(T_calc)+(i-1)*step2;
end
m2=hist(T_calc,T_point2); cdf2(1)=m2(1);
for i=2:n_point; cdf2(i)=m2(i)+cdf2(i-1); end
cdf2=cdf2/N; pdf2=m/N;

% Plotting cdf and pdf using above points
plot(T_point2,cdf2); hold on; plot(T_point1,cdf1,'--r');
xlabel('T'); ylabel('cdf'); title('cdf for samples of T-SORM-MCS');
legend('Uniformly distributed variables',...
    'Normally distributed variables',4);figure
plot(T_point2,pdf2); hold on; plot(T_point1,pdf1,'--r');
xlabel('T'); ylabel('pdf'); title('pdf for samples of T-SORM-MCS');
legend('Uniformly distributed variables',...
    'Normally distributed variables');figure

% Plotting histograms for the random & output variables
hist(k_sample); xlabel('Thermal conductivity');
title('Histogram for samples of k-SORM-MCS'); figure
hist(rho_Cp_sample); xlabel('Volumetric heat capacity');
title('Histogram for samples of rho-SORM-MCS'); figure
hist(T_calc); xlabel('Temperature');
title('Histogram for samples of T-SORM-MCS'); figure

%% Part (J): Model Error bounds
% Optimized values among all the methods
Maximum = [max(T_Taylor) max(T_WorstCase) max(T_SORM) max(T_MCS)...
           max(T_model) max(T_SORM_MCS)];
Minimum = [min(T_Taylor) min(T_WorstCase) min(T_SORM) min(T_MCS)...
           min(T_model) min(T_SORM_MCS)];

% Model error from extreme values
Err_low = (min(T_exp) - min(Maximum));
Err_high = (max(T_exp) - max(Minimum));
fprintf('\nInterval for Error_extreme = [%f %f]\n',Err_low, Err_high);

% Model error from Optimization values
Err_low = (min(T_exp) - max(T_model));
Err_high = (max(T_exp) - min(T_model));
fprintf('\nInterval for Error_opt = [%f %f]\n',Err_low, Err_high);
```

```
% Model error from Taylor series
Err_low = (min(T_exp) - max(T_Taylor));
Err_high = (max(T_exp) - min(T_Taylor));
fprintf('\nInterval for Error_Taylor = [%f %f]\n',Err_low, Err_high);

% Model error from SORM values
Err_low = (min(T_exp) - max(T_SORM));
Err_high = (max(T_exp) - min(T_SORM));
fprintf('\nInterval for Error_SORM = [%f %f]\n',Err_low, Err_high);

% Model error from Worst Case
Err_low = (min(T_exp) - max(T_WorstCase));
Err_high = (max(T_exp) - min(T_WorstCase));
fprintf('\nInterval for Error_WorstCase = [%f %f]\n',Err_low, Err_high);

% Model error from MCS on SORM model
Err_low = (min(T_exp) - max(T_SORM_MCS));
Err_high = (max(T_exp) - min(T_SORM_MCS));
fprintf('\nInterval for Error_SORM_MCS = [%f %f]\n',Err_low, Err_high);

% Model error from Golden Search Algorithm
T_GSA = [390.31981, 632.85195];
Err_low = (min(T_exp) - max(T_GSA));
Err_high = (max(T_exp) - min(T_GSA));
fprintf('\nInterval for Error_GSA = [%f %f]\n',Err_low, Err_high);

% Model error from MCS on original function
Err_low = (min(T_exp) - max(T_MCS));
Err_high = (max(T_exp) - min(T_MCS));
fprintf('\nInterval for Error_MCS_original = [%f %f]\n',...
    Err_low, Err_high);

%% Part (K): Plotting the intervals
plot(T_exp,'-+r','LineWidth',2); hold on; plot(T_model,'-*g');
hold on; plot(T_WorstCase,'-xb'); hold on;
plot(T_Taylor,'-sc'); hold on; plot(T_SORM,'-dm'); hold on;
plot(T_SORM_MCS,'-py'); plot(T_MCS,'--hk','LineWidth',1.5); hold on;
plot(T_GSA,'-+r'); ylabel('Temperature');
legend('Experimental','Optimization','Worst Case Analysis',...
    'Taylor series','SORM','SORM-MCS','MCS','GSA',2);

function objective_max = objfn_thermal_max_mod(a_max)
% Define constants or model parameters
q = 2000; L = 0.0254; T_ini = 25; x = 0; term = 0; t_max = 1000;

% Initialize variables
k_max = a_max(1); rho_Cp_max = a_max(2);

% Function evaluation
for n = 1:6
term = term + (1/n^2)*( exp(-n^2*pi^2*(((k_max/rho_Cp_max)*t_max)/...
        L^2)) )*cos(n*pi*x/L);
end
```

```
objective_max = -( T_ini + (q*L/k_max)*((((k_max/rho_Cp_max)*t_max)/...
            L^2) + (1/3) - (x/L) +(0.5*(x/L^2)) - ((2/pi^2)*term) ) );
end

function objective_min = objfn_thermal_min_mod(a_min)
% Define constants or model parameters
q = 2000; L = 0.0254; T_ini = 25; x = 0; term = 0; t_min = 1000;

% Initialize variables
k_min = a_min(1); rho_Cp_min = a_min(2);

% Function to be evaluated
for n = 1:6
term = term +(1/n^2)*(exp(-n^2*pi^2*(((k_min/rho_Cp_min)*t_min)/L^2)))...
    *cos(n*pi*x/L);
end

objective_min = (T_ini + (q*L/k_min)*((((k_min/rho_Cp_min)*t_min)/L^2)...
    + (1/3) - (x/L) +...
    (0.5*(x/L^2)) - ((2/pi^2)*term) ) );
end
```

APPENDIX C

Verification of Confidence Level for Model Error: MATLAB Source Code

```
% Program to determine the confidence level in model
% uncertainty interval

clear all; close all; clc;
format long;

% Define variables
q = 2000; L = 0.0254; T_ini = 25; x = 0; t = 1000; N = 1e5;

% Experimental values for thermal conductivity
k_delta = [0.0496; 0.0530; 0.0493; 0.0455; 0.0483; 0.0490; 0.0628;...
           0.0620; 0.0537; 0.0561; 0.0563; 0.0622; 0.0602; 0.0546;...
           0.0638; 0.0614; 0.0643; 0.0714; 0.0657; 0.0713; 0.0694;...
           0.0732; 0.0684; 0.0662; 0.0631; 0.0796; 0.0692; 0.0739;...
           0.0806; 0.0811];

% Experimental values for Volumetric heat capacity
rho_Cp_delta = 1e5*[3.76; 3.38; 3.50; 4.13; 4.02; 3.53; 3.87; 4.69;...
                    4.19; 4.28; 3.37; 3.77; 4.52; 4.10; 4.02; 3.94;...
                    3.73; 3.69; 4.68; 4.24; 3.72; 3.46; 4.07; 3.99;...
                    4.19; 4.38; 3.45; 3.95; 3.78; 3.77];

% Mean and Standard deviation
k_mean = mean(k_delta); k_std = std(k_delta);
rho_Cp_mean = mean(rho_Cp_delta); rho_Cp_std = std(rho_Cp_delta);

% Determination of new mean and standard deviation acc to Truncated
% distributions
k_new_mean = k_mean; k_new_std = 0.87977*(k_std);
rho_Cp_new_mean = rho_Cp_mean; rho_Cp_new_std = 0.87977*(rho_Cp_std);

% MCS on original model using truncated mean and std deviation for y_m
for i = 1:100
    k_sample = normrnd(k_new_mean,k_new_std,N,1);
    rho_Cp_sample = normrnd(rho_Cp_new_mean,rho_Cp_new_std,N,1);

    for j = 1:N
        term = 0;
        for n = 1:6
            term = term + (1/n^2)*( exp(-n^2*pi^2*(((k_sample(j)./...
                   rho_Cp_sample(j)).*t)./L^2)) )*cos(n*pi*x/L);
        end
        T_cal(j) = T_ini + (q*L./k_sample(j)).*( (((k_sample(j)./...
                   rho_Cp_sample(j)).*t)./L^2) + (1/3) - (x/L) +...
                   (0.5*(x/L^2)) - ((2/pi^2)*term) );
    end
    Min_T(i) = min(T_cal); Max_T(i) = max(T_cal);
end

% Display output
fprintf('Interval for T_proof_MCS1 = [%f %f]\n',...
    min(Min_T), max(Max_T));
T_proof_MCS1 = [min(Min_T) max(Max_T)];
```

```
% MCS for T values using exp mean and std deviation for y_e
T = [475.6 493.9 496.4 517.6];
mu_T = mean(T); std_T = std(T);
T_sample = normrnd(mu_T, std_T, N, 1);
T_sample = T_sample';

% Defining the uncertainty intervals from uncertainty analysis
U_GSA = [-157.10969, 127.19768];
U_SORMmcs = [-145.96256, 120.85534];
U_opt =  [-97.87430, 92.88659];
U_sorm = [-68.14677, 86.58460];

% Initializing counters for different methods
c_gsa = 0; c_opt = 0; c_sormmcs = 0; c_sorm = 0;

% Determining the probability of error lying in original intervals
Diff1 = T_sample - T_cal;

% Check for Golden Search Algorithm
for i=1:N
    if (Diff1(i)>min(U_GSA)) && (Diff1(i)<max(U_GSA))
        c_gsa = c_gsa+1;
    else
    end
end
p_gsa = c_gsa/N

% Check for SORM_MCS
for i=1:N
    if (Diff1(i)>min(U_SORMmcs)) && (Diff1(i)<max(U_SORMmcs))
        c_sormmcs = c_sormmcs+1;
    else
    end
end
p_sormmcs = c_sormmcs/N

% Check for Optimization
for i=1:N
    if (Diff1(i)>min(U_opt)) && (Diff1(i)<max(U_opt))
        c_opt = c_opt+1;
    else
    end
end
p_opt = c_opt/N

% Check for SORM
for i=1:N
    if (Diff1(i)>min(U_sorm)) && (Diff1(i)<max(U_sorm))
        c_sorm = c_sorm+1;
    else
    end
end
p_sorm = c_sorm/N
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MCS on original model using old mean and std deviation
% for Temperature
for i = 1:100
    k_sample = normrnd(k_mean,k_std,N,1);
    rho_Cp_sample = normrnd(rho_Cp_mean,rho_Cp_std,N,1);

    for j = 1:N
        term = 0;
        for n = 1:6
            term = term + (1/n^2)*( exp(-n^2*pi^2*(((k_sample(j)./...
                    rho_Cp_sample(j)).*t)./L^2)) )*cos(n*pi*x/L);
        end

        T_calc1(j) = T_ini + (q*L./k_sample(j)).*( (((k_sample(j)./...
                        rho_Cp_sample(j)).*t)./L^2) + (1/3) - (x/L) +...
                    (0.5*(x/L^2)) - ((2/pi^2)*term) );
    end
    Min_T_calc1(i) = min(T_calc1);
    Max_T_calc1(i) = max(T_calc1);
    means(i) = mean(T_calc1); stds(i) = std(T_calc1);
end

% Display output
fprintf('Interval for T_proof_MCS2 = [%f %f]\n',min(Min_T_calc1),...
    max(Max_T_calc1));
T_proof_MCS2 = [min(Min_T_calc1) max(Max_T_calc1)];

% Determining the probability of temperature interval
% lying in 95% interval
p = normcdf(T_proof_MCS1,mean(means),mean(stds));
Prob = p(2)-p(1)

% Defining the uncertainty intervals from uncertainty analysis
T_GSA = [390.31981, 632.85195];
T_SORMmcs = [396.74466, 621.56256];
T_opt =  [424.71341, 573.47430];
T_sorm = [431.01541, 543.74677];

% Initializing counters for different methods
t_gsa = 0; t_opt = 0; t_sormmcs = 0; t_sorm = 0;

% Check for Golden Search Algorithm
for i=1:N
    if (T_calc1(i)>min(T_GSA)) && (T_calc1(i)<max(T_GSA))
        t_gsa = t_gsa+1;
    else
    end
end
pt_gsa = t_gsa/N

% Check for SORM_MCS
for i=1:N
    if (T_calc1(i)>min(T_SORMmcs)) && (T_calc1(i)<max(T_SORMmcs))
```

```
        t_sormmcs = t_sormmcs+1;
    else
    end
end
pt_sormmcs = t_sormmcs/N

% Check for Optimization
for i=1:N
    if (T_calc1(i)>min(T_opt)) && (T_calc1(i)<max(T_opt))
        t_opt = t_opt+1;
    else
    end
end
pt_opt = t_opt/N

% Check for SORM
for i=1:N
    if (T_calc1(i)>min(T_sorm)) && (T_calc1(i)<max(T_sorm))
        t_sorm = t_sorm+1;
    else
    end
end
pt_sorm = t_sorm/N
```

BIBLIOGRAPHY

[1] K.J. Dowding, M. Pilch, and R. Hills. Formulation of the thermal problem. *Computer Methods in Applied Mechanics and Engineering*, 197(2008):2385–2389, 2007.

[2] *Hall, M.* Geoscience consulting for geoscientists. `http://www.agilegeoscience.com/journal/tag/data`, 2011 (accessed Jan 22, 2011).

[3] D. Olive. Truncated distributions. `http://www.math.siu.edu/olive/ch4.pdf`, 2011 (accessed Jan 22, 2011).

[4] F.M. Hemez and S.W. Doebling. Model validation and uncertainty quantification. In 19^{th} *International Modal Analysis Conference*, Kissimmee, Florida, February 5-8, 2001.

[5] K.R. Popper. *Conjectures and Refutations: The Growth of Scientific Knowledge*. Routledge and Paul, London, 1969.

[6] J.C. Refsgaard, J.P. Sluijs, J. Brown, and P. Keur. A framework for dealing with uncertainty due to model structure error. *Advances in Water Resources*, 29(11):1586–1597, 2006.

[7] X. Du. Reliability-based design using saddlepoint approximation. In *ASME International Design Engineering Technical Conferences*, Philadelphia, Pennsylvania, September 10-13, 2006.

[8] H. Agarwal, C.K. Mozumder, J.E. Renaud, and L.T. Watson. An inverse-measure-based unilevel architecture for reliability-based design optimization. *Structural and Multidisciplinary Optimization*, 33(3):217–227, 2007.

[9] H. Arenbeck, S. Missoum, A. Basudhar, and P. Nikravesh. Reliability-based optimal design and tolerancing for multibody systems using explicit design space decomposition. *Journal of Mechanical Design*, 132(2):021010, 2010.

[10] J.K. Allen, C. Seepersad, and H. Choi. Robust design for multiscale and multidisciplinary applications. *Journal of Mechanical Design, Transactions of the ASME*, 128(4):832–843, 2006.

[11] Z. Dai, M.J. Scott, and Z.P. Mourelatos. Robust design using preference aggregation methods. In *ASME International Design Engineering Technical Conferences*, Chicago, Illinois, September 2-6, 2003.

[12] C.D. McAllister and T.W. Simpson. Multidisciplinary robust design optimization of an internal combustion engine. *Journal of Mechanical Design, Transactions of the ASME*, 125(1):124–130, 2003.

[13] W. Chen, L. Baghdasaryan, T. Buranathiti, and J. Cao. Model validation via uncertainty propagation and data transformations. *AIAA Journal*, 42(7):1406–1415–843, 2004.

[14] T. Buranathiti, J. Cao, W. Chen, L. Baghdasaryan, and Z.C. Xia. Approaches for model validation: Methodology and illustration on a sheet metal flanging process. *ASME Journal of Manufacturing Science and Engineering*, 128(2):588–597, 2006.

[15] Y. Xiong, W. Chen, K.L. Tsui, and D.W. Apley. A better understanding of model updating strategies in validating engineering models. *Computer Methods in Applied Mechanics and Engineering*, 198(15-16):1327–1337, 2009.

[16] W.L. Oberkampf, T.G. Trucano, and C. Hirsch. Verification, validation and predictive capability in computational engineering and physics. *Applied Mechanics Reviews*, 57(1-6):345–384, 2004.

[17] R. Hill. Model validation: Model parameter and measurement uncertainty. *Journal of heat transfer, Transactions of the ASME*, 128:339–351, 2006.

[18] T.G. Trucano. Uncertainty quantification at sandia. In *SAND2000-0524C, Sandia National Laboratories*, Albuquerque, NM, 2000.

[19] L. Baghdasaryan, W. Chen, T. Buranathiti, and J. Cao. Model validation via uncertainty propagation using response surface models. In *ASME International Design Engineering Technical Conferences*, Montreal, Canada, September 29 - October 2, 2002.

[20] R. Rebba and S. Mahadevan. Statistical methods for model validation under uncertainty. In 47^{th} *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Newport, Rhode Island, May 1-4, 2006.

[21] R. Hills and K.J. Dowding. Multivariate approach to the thermal challenge problem. *Computer Methods in Applied Mechanics and Engineering*, 197(2008):2442–2456, 2007.

[22] M.D. Brandyberry. Thermal problem solution using a surrogate model clustering technique. *Computer Methods in Applied Mechanics and Engineering*, 197(2008):2390–2407, 2007.

[23] S. Ferson, W.L. Oberkampf, and L. Ginzburg. Model validation and predictive capability for the thermal challenge problem. *Computer Methods in Applied Mechanics and Engineering*, 197(29-32):2408–2430, 2008.

[24] K. Poolla and P. Khargonekar. A time-domain approach to model validation. *IEEE Transactions on Automatic Control*, 39(5):951–959, 1994.

[25] K.C. Yap, G.W. Wathugala, and T.K. Hasselman. An updated toolbox for validation and uncertainty quantification of non-linear finite element models. In 7^{th} *International LS-DYNA Users Conference*, Dearborn, Michigan, May 19-21, 2002.

[26] A.E. Rice, M. Arviso, T.L. Paez, T.G. Carne, and P.S. Hunter. Uncertainty quantification in model validation of a complex aerospace structure. In *Proceedings of the IMAC-XXVII*, Orlando, Florida, February 9-12, 2009.

[27] R.H. Dieck. Measurement uncertainty models. *ISA Transactions*, 36(1):29–35, 1997.

[28] Institute A.N.S. and Engineers A. S. O. M. Guidelines for the evaluation of dimensional measurement uncertainty. Technical report, American Society of Mechanical engineers, 2007.

[29] I. Vim. International vocabulary of basic and general terms in metrology. *International Organization for Standardization*, pages 9–14, 2004.

[30] B.R. Bettis and S. Hosder. Quantification of uncertainty in aerodynamic heating of a re-entry vehicle due to uncertain wall and freestream conditions. In 10^{th} *AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, Chicago, Illinois, 28 June-1 july, 2010.

[31] W.L. Oberkampf, J.C. Helton, and K. Sentz. Mathematical representation of uncertainty, aiaa paper 2001-1645. In 3^{rd} *Non-deterministic Approaches Forum*, Seattle, WA, April, 2001.

[32] M. Eldred and L. Swiler. Efficient algorithms for mixed aleatory-epistemic uncertainty quantifications with application to radiation hardened electronics. Technical Report Vol. SAND2009-5805, Sandia National Laboratories Report, September, 2009.

[33] L. Swiler, T. Paez, R. Mayes, and M. Eldred. Epistemic uncertainty in calculation of margins, aiaa paper 2000-2249. In 50^{th} *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, May 4-7, 2009.

[34] H.W. Coleman. *Experimentation and uncertainty Analysis for Engineers*. 0471635170. Wiley-interscience Publication, New York, 1989.

[35] T.M. Mitchell. First order taylor series approximations and cost functions. *The Scandinavian Journal of Economics*, 92(3):513–524, 1990.

[36] X. Du. Unified uncertainty analysis by the first order reliability method. *Journal of mechanical Design, Transaction of the ASME*, 130(9):0914011–09140110, 2008.

[37] M.S. Eldred. Second order reliability formulations in dakota/uq. In 47^{th} *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Newport, Rhods Island, May 1-4, 2006.

[38] S.C. Chapra. *Applied numerical methods with Matlab for engineers and scientists*. 9780072976779. McGraw Hill Higher Education, Boston, Massachusetts, 2005.

[39] C.F. Gerald and O. Patrick. *Applied numerical Analysis*. Addison-Wesley, seventh edition, 2004.

[40] R.C. Penmetsa and R.V. Grandhi. Efficient estimation of structural reliability for problems with uncertain intervals. *Computers and Structures*, 80(12):1103–1112, 2002.

[41] J. Guo and X. Du. Reliability analysis for multidisciplinary systems with random and interval variables. *AIAA Journal*, 48(1):82–91, 2010.

[42] O. Garcia, J. Vehi, E.M. Campos, A. Henriques, and R. Casas. Structural assesment under uncertain parameters via interval analysis. *Journal of Computational and Applied Mathematics*, 218(1):43–52, 2008.

[43] X. Du, P.K. Venigella, and D. Liu. Robust mechanism synthesis with random and interval variables. *Mechanism and Machine theory*, 44(7):1321–1337, 2009.

[44] X. Du, A. Sudjianto, and B. Huang. Reliability based design with a mixture of random and interval variables. *Journal of mechanical Design, Transaction of the ASME*, 127(6):1068–1076, 2005.

[45] J. Guo and X. Du. Reliability sensitivity analysis with random and interval variables. *International Journal for Numerical methods in Engineering*, 78(13):1585–1617, 2009.

[46] E.R. Hansen. Sharpness in interval computations. *Reliable Computing*, 3(1):17–29, 1997.

[47] C. Jiang, X. Han, F.J. Guan, and Y.H. Li. An uncertain structural optimization method based on non-linear interval number programming and interval analysis method. *Engineering Structures*, 29(11):3168–3177, 2007.

[48] B.R. Hunt, R.L. Lipsman, and J.M. Rosenberg. *Guide to MATLAB: For Beginners and Experienced Users*. 9780511219177. Cambridge University Press, West Nyack, NY, second edition, 2002.

[49] S.V. Crowder and R.D. Moyer. A two stage monte carlo approach to the expression of uncertainty with non-linear measurement equation and small sample size. *Institute of Physics Publishing, Metrologia*, 43(2006):34–41, 2005.

[50] S. Hosder, R.W. Walters, and M. Balch. Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables, aiaa 2007-1939. In 48^{th} *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii, April 23-26l, 2007.

[51] T.E. Lovett and A. Monti. A polynomial chaos approach to measurement uncertainty. *IEEE Transactions on Instrumentation and Measurement*, 55(3):729–736, 2006.

[52] C. Arndt. An introduction to systematic sensitivity analysis via gaussian quadrature. Technical Report 2, Global trade Analysis Project, West Lafayette, IN, July 1996.

VITA

Harsheel Rajesh Shah was born in Mumbai, Maharashtra (India) on November 04, 1985. He started studying in St. Francis D Assissi High school located in Mumbai. After gradutaing from the secondary school in 2001, he opted to excel in the field of science and hence, chose to pursue high school in Mumbai, M.V.L.U college. Successfully passing the high school with good percentage in 2003, he started his undergraduate career in Mechanical Engineering at the University of Mumbai, Vidyavardhini's College of engineering and Technology. He graduated from the university as a Bachelor of Science in Spring 2007. During his undergraduate career, he was an active member of the college core committee helping organize different cultural and academic events. He worked as a Mechanical Engineer in a multinational company Tecnimont ICB ltd. till 2009. After gaining two years of work experience, he pursued his higher studies in the Aerospace field in Missouri University of Science and Technology, USA. He served as the president of India Association for the academic year 2009-2010 and began conducting research under Dr. Xiaoping Du in the field of uncertainty. He received his Master of Science Degree in Aerospace Engineering at Missouri University of Science and Technology in December 2011.