

MAR 9 2000

SANDIA REPORT

SAND2000-0515

Unlimited Release

Printed March 2000

RECEIVED
MAR 20 2000
OSTI

Methodology for Characterizing Modeling and Discretization Uncertainties in Computational Simulation

Kenneth F. Alvin, William L. Oberkampf, Brian M. Rutherford, and Kathleen V. Diegert

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States
Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (703) 605-6000
Web site: <http://www.ntis.gov/ordering.htm>

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161



DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND2000-0515
Unlimited Release
Printed March 2000

Methodology for Characterizing Modeling and Discretization Uncertainties in Computational Simulation

Kenneth F. Alvin
Structural Dynamics Development and Smart Systems Department

William L. Oberkampf
Validation and Uncertainty Quantification Department

Brian M. Rutherford
Statistics and Human Factors Department

Kathleen V. Diegert
Reliability Assessment Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0847

Abstract

This research effort focuses on methodology for quantifying the effects of model uncertainty and discretization error on computational modeling and simulation. The work is directed towards developing methodologies which treat model form assumptions within an overall framework for uncertainty quantification, for the purpose of developing estimates of total prediction uncertainty. The present effort consists of work in three areas: framework development for sources of uncertainty and error in the modeling and simulation process which impact model structure; model uncertainty assessment and propagation through Bayesian inference methods; and discretization error estimation within the context of non-deterministic analysis.

Acknowledgment

We acknowledge the generous support of the Sandia Computational Sciences Research Foundation through the Laboratory Directed Research and Development Program, and the infinite patience of the LDRD office.

Contents

Acknowledgment	ii
Introduction.....	1
Concepts of Uncertainties and Errors	4
Introduction and Background	4
Definitions: Variability, Uncertainty and Error.....	4
Realizations of the Modeling Process.....	8
Model structure, continuous and discrete parameters.....	10
Types of uncertainties and errors in model structure.....	11
Propagation tree structures for model realizations.....	11
Project Activities and Contributions in Identification of Uncertainty and Error Sources in Computational Simulation	14
Assessment and Propagation of Model Uncertainty	15
Introduction.....	15
Statistical Inference using Prior Probabilities.....	15
Revising Prior Probabilities to Account for New Data.....	17
Propagation of Model Uncertainty: Bayesian approach	19
Combining model uncertainty with parametric variability	21
Assignment of prior subjective model probabilities	23
Selection of Models	24
Project Activities and Contributions on Assessment and Propagation of Model Uncertainty.....	25
Estimation of Discrete Numerical Solution Errors	28
Grid convergence and error estimation via Richardson extrapolation.....	30
A-posteriori error estimation in the finite element method	32
Project Activities and Contributions on Estimation of Numerical Solution Error	37
Conclusions.....	38
References.....	40
APPENDIX A–Uncertainty Quantification in Computational Structural Dynamics: A New Paradigm for Model Validation	43
APPENDIX B–A Method for Treating Discretization Error in Non-Deterministic Analysis	53
APPENDIX C–A New Methodology for the Estimation of Total Uncertainty in Computational Simulation	83
APPENDIX D–Use Of Metamodels And Qualitative Inputs For Assessment And Propagation Of Errors Due To Uncertainty In Model Form	114
APPENDIX E–Combining Metamodels with Rational Function Representations of Discretization Error for Uncertainty Quantification	163

Figures

Trade-off between Model Uncertainty and Solution Error	7
Framework for Sources of Uncertainty and Error in Computational Simulation	12
Propagation Tree of Model Structures	13
A distribution of a discrete set of deterministic models	20
A distribution of a discrete set of nondeterministic models	22
Decomposing model probabilities in a propagation tree structure	25
Proposed Phases of Modeling and Simulation	52
Sources of Uncertainty and Error in Modeling and Simulation	52
Model I: Paved/Extruded Hex Mesh w/whole detail (no fillets) n=31554	52
Model II: ProE geometry (holes and fillets) meshed w/10-node tet elements n=50565	52
Model III: Triparametric Hex Mesh w/no hole or fillet details n=53202	52
Uncertainty Analysis using Equal Assumed Probabilities	52
1-D Prismatic Bar Model with Nominal Mesh	79
Results from Nondeterministic Analysis of 1-D Bar	80
Proposed Phases for Computational Modeling and Simulation	88
Activities Conducted in the Phases of Computational Modeling and Simulation.....	90
Conceptual Modeling Activities for the Missile Flight Example	91
Mathematical Models for the Missile Flight Example	94
Tree-Structure for Models, Solutions, and Representations in the Missile Flight Example	98
Histogram from LHS for Mass Variability	100
Variability in Range due to Variability in Initial Weight	100
Frequency Data from LHS for Range Offset due to Initial Weight.....	101
Uncertainty in Range due to Thrust Uncertainty and Mass Variability for 6-DOF Model	101
Frequency Data from LHS for Range Uncertainty due to Thrust Uncertainty for 6-DOF Model.....	102
Uncertainty in Range due to Solution Error and Mass Variability for 6-DOF Model	102
Uncertainty in Range due to Solution Error and Mass Variability for 3-DOF Model	103
Uncertainty in Range due to Mass Variability, Thrust Uncertainty, and Solution Error for 6-DOF Model	104
Uncertainty in Range due to Mass Variability and Solution Error for the High Temperature Motor for 3-DOF Model.....	104
D-Optimal design parameter.	139
Selected experiment design points.	139
RMS displacement for 86 point experiment design sorted by model. - . . exact, ___ estimated	140
Peak displacement for 86 point experiment design sorted by model. - . . exact, ___ estimated	140
RMS displacement for 1,002 candidate experiments.	141
Peak displacement for 1,002 candidate experiments.	141
Peak displacement residual vs. metamodel prediction.	143
Peak displacement residual vs. forcing frequency.	143

Frequency distribution for rms displacement - Case I.	144
Frequency distribution for predicted rms displacement - Case I.	144
Frequency distribution for rms displacement - Case II.	145
Frequency distribution for predicted rms displacement - Case II.	145

Intentionally Left Blank

1. Introduction

To meet the goals of the scientifically-based stockpile stewardship program, significantly improved confidence must be established in the wide variety of methodologies and approaches associated with numerical simulations. The goal of these simulations is to accurately characterize the response of the complete system with high fidelity three-dimensional, unsteady, coupled physics, together with uncertainty bounds on the response predictions. Quantifying the uncertainties in computational physics-based simulations is also critical to the verification and validation of these simulations; without uncertainty quantification, validation is highly subjective and arbitrary. Total uncertainty in computational analysis is understood to arise from a full range of modeling and simulation activities which can be broadly classified as variabilities, errors, and uncertainties. Similarly, the computational activities which comprise uncertainty quantification can be broadly classified as nondeterministic analysis, numerical error estimation, and assessment and propagation of model uncertainty.

Nondeterministic analysis refers to methods which propagate variabilities in model parameters (usually defined in terms of a probability distribution) through one or more models with the goal of estimating some statistics of interest on the predicted quantities of the models. These methods include such well-known techniques as Monte Carlo analysis, as well as reliability methods, stochastic finite element analysis, and response surface methods. Numerical error estimation refers to methods which seek to estimate errors in a numerical solution of differential equations, usually on the basis of an underlying model for the error space. Finally, assessment and propagation of model uncertainty refers to methods which address the form and assumptions underlying the mathematical model. This is also sometimes referred to as structural uncertainty in that it concerns the structure, rather than the parameters, of the model in question [1-2].

The vast majority of the research efforts in uncertainty quantification, both at Sandia and throughout the larger computational modeling and simulation community, has been directed towards analysis methodology for non-deterministic parameters. In nondeterministic analysis, the code itself through which variabilities are propagated is treated as a given deterministic function. Nondeterministic analysis does not therefore treat the effects of uncertainties and errors inherent in the conceptual modeling, mathematical modeling, discretization, programming, and discrete numerical phases of modeling and simulation. These issues are the concern of the numerical error estimation and model uncertainty phases of uncertainty quantification, and are the focus of the present research project. While numerical error estimation has been researched extensively, model uncertainty issues have lacked significant investigation. Uncertainty quantification methods related to errors and structural uncertainties are the least mature as they address

theoretical and conceptual issues, simplifications and trade-offs which are not only mathematically difficult to define, but are also difficult to order in terms of their impact on the solution. Furthermore, the coupling between parametric sources of variability, numerical solution error, and model uncertainty has not been studied. This document details work to develop methods to quantitatively estimate the errors or uncertainties due to the mathematical form of the partial differential equations and the errors due to the discretization of the equations.

Sandia is in an excellent position to development methods for model uncertainty quantification. The DOE ASCI program is funding the development of high fidelity physics simulation codes and computational platforms for those codes. ASCI will provide the computational resources necessary for analysts to estimate numerical error through mesh convergence studies, as well as problem solving environments which should enable analysts to quickly remodel problems while varying the level of feature fidelity in the partial differential equations, and the subgrid constitutive models used in the equations. Furthermore, the ongoing development of uncertainty and reliability analysis tools for non-deterministic analysis should help analysts to efficiently estimate the contribution from parametric sources to prediction uncertainty, and enable them to spend more time focusing on model uncertainty issues.

This research effort has initiated a collaboration with Professor Daniel Kammer at the University of Wisconsin at Madison. His contributions to this work on model uncertainty has heightened interest in response surface methods for uncertainty quantification throughout the engineering analysis community at Sandia. Furthermore, his well-recognized work in D-optimality experiment design should help to advance this aspect of uncertainty quantification, which still requires additional development. The research effort has also resulted in the development of a graduate-level university course, offered by the University of New Mexico (UNM) through the departments of Mechanical Engineering and Mathematics, on uncertainty estimation in computational modeling and simulation. This course was also offered as an elective in the UNM Scientific & Engineering Computation Certification Program, which was developed with Sandia for retraining of technical staff personnel in the discipline of computational modeling and simulation.

This Lab Directed Research and Development (LDRD) effort focuses on methods for characterizing modeling uncertainties and discretization errors in modeling and simulation. This report will summarize the results of this effort. While this project was considered high risk from the beginning, some noteworthy progress has been made. Furthermore, a number of publications have resulted from this work, including one accepted peer-reviewed publication, three papers in or being prepared for peer-reviewed journal review, four conference papers, and one invited presentation at a University of Minnesota-sponsored symposium on usability of mathematical models. Finally, this research effort, as part of a multi-disciplinary uncertainty quantification program involving two other ESRF-funded LDRD projects and numerous ESRF tech-base projects, has raised the stature of Sandia in

the ASCI Verification and Validation Program, and helped lead to the development of new departments in both Engineering Sciences and Computational Sciences directed towards uncertainty quantification and validation.

2. Concepts of Uncertainties and Errors

Introduction and Background

This section provides a discussion of significant sources of uncertainty and error which are relevant to modeling and discretization. Primarily these are mathematical modeling uncertainties and acknowledged errors that occur in the mathematical modeling process and in computing the discrete numerical solution. Mathematical modeling uncertainties arise due to lack of knowledge of the appropriate mathematical representation of the conservation equations, auxiliary equations and boundary/initial conditions. These can be referred to as model "form" uncertainties, or structural uncertainties, to distinguish them from the continuous parametric uncertainties and variabilities treated via nondeterministic analysis. Acknowledged errors occur in the mathematical modeling and numerical solution activities, and to some extent in the discretization activities. This discussion will focus mainly these sources of prediction uncertainty.

The remainder of this section proceeds as follows. First, specific definitions are offered for variability, uncertainty and error, which are regarded as general categories into which specific contributors to prediction uncertainty can be grouped. Then the concepts of model form uncertainty are introduced, which deal with prediction uncertainty contributors arising from the conceptual and mathematical form of the equations used in modeling and simulation. Important issues discussed within the context of model form uncertainty include the trade-off between simplification errors, which affect the choice of continuous partial differential equations, and numerical errors, which affect the conversion of the continuous equations into solvable algebraic equations. This leads to the concept of realizations of the modeling process, from which alternative structural choices in modeling can arise. Within the context of structural realizations of the modeling process, parameters, uncertainties and errors are discussed and a decision tree-like structure for visualizing the propagation of structural errors and uncertainties through model realizations is introduced. Finally, specific project activities related to the identification of uncertainty and error sources in computational simulation are summarized.

Definitions: Variability, Uncertainty and Error

In [3] and Appendices A and C, specific definitions of variability, uncertainty and error are developed in order to understand the sources of total modeling and simulation uncertainty. In the present context of mathematical modeling and numerical solution, variability is not generally encountered since it arises from replicated random phenomena or processes and not from the modeling process itself. However, there is a need to consider

uncertainties and errors and their similarities and differences. Uncertainty is defined as a potential deficiency which is due to lack of knowledge. An example is energy dissipation in vibrating systems, such as structures and machinery. Energy dissipation is thought to arise primarily from friction and slippage at material interfaces. These phenomena are not well understood, and so there is a lack of knowledge regarding the models for predicting these effects. Error, on the other hand, is defined as a recognizable deficiency in any phase of modeling and simulation which is *not* due to lack of knowledge. Furthermore, errors can be categorized as either acknowledged or unacknowledged. Unacknowledged errors are mistakes or blunders, whereas acknowledged errors are those deviations from a more accurate modeling and simulation process which are consciously made for reasons of economy or necessity. The motivation for identifying all sources is that acknowledged errors are documented so that their presence in the analysis can be reviewed and considered by peers, and possibly treated via quantification methods. Acknowledged errors which are not typically documented are almost as troublesome as unacknowledged errors in terms of their effect on the modeling and simulation process.

Genuine model form uncertainty arises when there is lack of knowledge about the appropriate form of the model equations. This could occur when competing theories exist for the mathematical description of some phenomenon, such as in the example of energy dissipation in vibrating systems. A more common example is when an empirical submodel is used within a simulation model. Then there may be uncertainty on the form of the submodel equations because several plausible equations are available to fit the empirical data. An example of this would be constitutive modeling for many materials, particularly when describing material behavior outside the linear elastic range. Most of the issues which arise in what is often termed "model form uncertainty," however, are not genuine uncertainties at all but instead are related to acknowledged errors, and in particular, simplification errors.

One example of acknowledged error in simulations which involve the numerical solution of partial differential equations is discretization error. This error is the result of using a discretization method with a finite number of degrees of freedom to solve the set of differential equations. These errors lead to a bias in the computed solution with respect to the true solution of the continuous differential equations. Discretization errors are a special class of acknowledged errors because they can be understood and ordered by use of an error model, which describes the simulation error in terms of the parameter which accounts for the error. For discretization error in the finite element method, where the elemental interpolation functions are polynomials, the appropriate error model is one where the error is proportional to the characteristic mesh spacing taken to some power. The value of that exponent is related to the interpolation or shape functions used for the elements, and the relationship between the predicted quantity and the field variable which is being discretized.

Simplification errors are the most common form of acknowledged error and are distinct from discretization errors because they cannot generally be ordered and modeled. One example of simplification error is any sort of geometrical simplification incorporated into a mathematical model, such as ignoring a small geometric feature of a body. It is clearly an error as a model which does not incorporate this simplification is more accurate, at least *ideally*, than the chosen approach, independent of the magnitude and direction of other errors. But what happens when two or more models differ by a variety of simplification errors, such as might occur when different analysts build models of the same system according to their own judgements? If one model is clearly inferior its results can be discarded or down-weighted for the final analyses. But when multiple models have both advantages and disadvantages in treating the conceptual model, it is impossible to distinguish which is more accurate without either empirical data or quantification of the effects of each modeling error. In this case there are a number of competing mathematical models which are in error with respect to the conceptual model, but whose accuracy relative to one another is uncertain. This is a situation in which uncertainty and error can appear very similar.

It might seem that the proper approach to simplification error is to simply eliminate it by modeling in more and more detail. But modeling is always an abstraction and there are always additional levels of simplification and complexity. Since resources are finite, at some point there must be an acceptance of the remaining simplifications and an attempt to understand their gross effects on prediction. Furthermore, it usually is the case that analysis capabilities are "playing catch-up" with systems design. As soon as predictive modeling and simulation is a reality for a certain class of systems (by an understanding of the appropriate level of simplification in modeling), the systems of interest to designers have grown more complex and have introduced new material and physics uncertainties. For example, Sandia is already at the forefront of the development of microelectronic-mechanical systems (MEMS) and micromachines, to be inevitably followed by nanotechnology. Finally, the progressive removal of simplifications inevitably leads to the coupling of micro and macro-scale effects, which significantly complicates the numerics and may often lead to the emergence of localized chaotic behavior. Again, the example of damping in vibration comes to mind. It is thought that the primary mechanisms at play in this case are friction between surfaces, changes in contact surfaces, dynamic impacts, and micro-slip. Understanding the slip and friction phenomena alone from "first principles" would require modeling the surface profile, which is a stochastic function.

Ultimately, the key relationship between model form uncertainty and error is the trade-off between them which occurs when determining the level of complexity in the model. This is illustrated somewhat crudely in Figure 1, which illustrates the level of model uncertainty and solution error inherent in five models which vary in complexity. This is a notional illustration and is not based upon actual results. Here the models are compared under the assumption that the same fixed level of computational effort is devoted to all the models. Technically, it could be said that what is being traded off is simplification error and

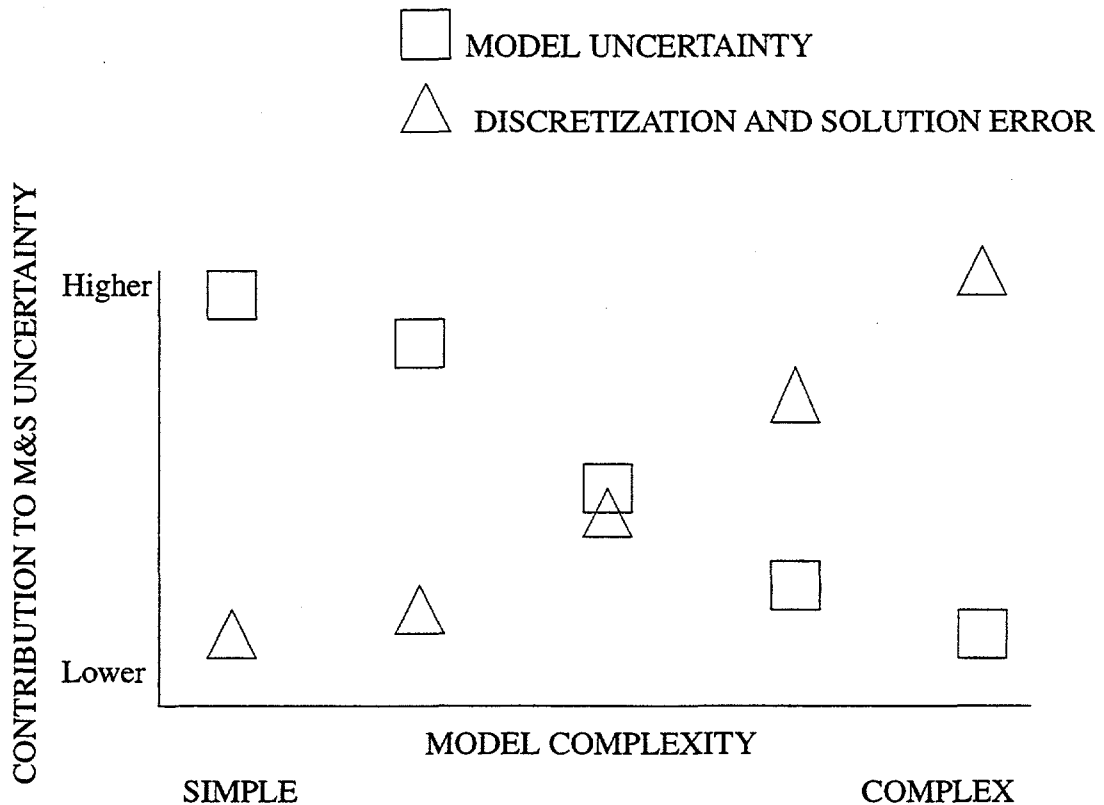


Figure 1: Trade-off between Model Uncertainty and Solution Error

solution error, but the practical impact of simplification error is a relative level of model uncertainty as was discussed above. The higher the level of simplification, the higher the resultant uncertainty in the applicability of the model and the higher the number of plausible alternative assumptions. But removing simplification errors does not necessarily improve predictive accuracy, since reducing the simplification of the model inevitably leads to higher numerical solution error (from discretization, for example) when resources are limited. Thus, at each level of modeling complexity, as well as across a range of complexity levels, it is possible to develop a suite of models whose ordered accuracy cannot be established. This relative model uncertainty arising from the lack of ordered accuracy amongst a suite of simplified models is practically no different than the case of "genuine" model uncertainty arising from a lack of physics knowledge.

Notwithstanding these analogies between model simplification and model uncertainty, simplifications will continue to be referred to as errors because it is important to the interpretation of an uncertainty analysis, regardless of the methodology used for quantifying their contribution to uncertainty. To clarify, suppose an analyst treats a set of competing models in a quasi-probabilistic fashion by assigning probabilities to each model and combining the results of the set of models according to those probabilities. If the underlying source of multiple models is true uncertainty (lack of knowledge of the

physics), it might be reasonable to treat the distribution of model set results as a best estimate of the real world, conditioned on the phenomenological experiments available. Furthermore, the set of models can only be expected to be refined and improved by collecting additional information; i.e. by performing some additional experiments which directly address the lack of knowledge. On the other hand, if the set of models differ by a set of acknowledged errors from the conceptual model, then it should be acknowledged that the distribution of the model set is not necessarily the best estimate given the knowledge base, but simply the best available given a constraint on resources. Furthermore, the model set will not necessarily bound in any sense the real world and it could have a bias with respect to more accurate models. Improving the results of this analysis in this case would then entail devoting more resources to the computational analysis effort.

In summary, true model uncertainties involve a lack of knowledge of the appropriate physics. In practice, however, many sources of so-called model uncertainty result from acknowledged errors and in particular simplifying assumptions. The existence of plausible alternative assumptions, as well as other levels of modeling complexity with varying levels of solution error, result in a potentially large set of plausible models whose relative accuracy cannot be modeled or even ordered. Therefore, there is a blurring between error sources and uncertainty sources. And while methodologies for dealing with genuine uncertainty and relative uncertainty (due to simplification errors) may be similar, it is important to segregate these sources as much as possible so that the analyst and policy maker can understand the nature of the uncertainty and how best to reduce its effects.

Realizations of the Modeling Process

It is critical to view any mathematical model in a broad context, as a particular intersection of assumptions in a larger pool of potential models. Only then can the analyst step back and consider that other models probably exist with equal (or nearly equal) validity but perhaps different behaviors. Unfortunately, the current state of algorithms and software for defining mathematical models and their discretization is an impediment toward understanding and assessing model uncertainty. Simply defining, discretizing and programming a mathematical model can be such a labor intensive task that often the end product of that effort is thought of as THE model, right down to the nodes and elements of the mesh. But even if the modeling task required six months of effort, it is still only one realization of the modeling process.

The term realization is borrowed from systems theory, where Krippendorff [4] defines it as:

“The production of an object or system according to specific instructions, plans or programs, e.g., building a house according to blueprints, replicating a scientific experiment published previously, cooking by recipe, realizing an artistic idea.”

In the context of computational physics simulations, a mathematical model is a realization

of the mathematical modeling process. A model is never unique unless the process used to derive, estimate, or develop the model constrains it to take on only one form. Although the modeling process can have a infinite number of realizations, it is not necessarily true that those models predict different behaviors for a pre-selected set of inputs and outputs. In fact, equivalent realizations are a set of realizations for which the input-output mapping functions are identical, despite differences in the model parameters. It is also possible for realizations to have different orders of dynamic complexity and yet be equivalent.

Just as understanding of uncertainty in physical processes is ultimately grounded in an understanding of the variability in the environment, an understanding of model uncertainty begins with the reality that modeling is approximation, and that the process used to arrive at a model is itself not deterministic. At all levels of modeling complexity, simplifying assumptions exist which are the result of trade-offs between accuracy and efficiency. The function used to measure and optimize this trade-off is defined by the analyst, usually in an intuitive and non-explicit fashion. A natural question arises: what kinds of models are important to the understanding, and assessment, of model uncertainty? The brief answer is that one should seek models which are plausible from the standpoint of the system description and a knowledge of physics, and are diverse in their underlying assumptions. The desire for model diversity arises from the fact that it is the difference between the input-output behaviors over the entire response space of the considered models which is evidence of model uncertainty. That is, in the absence of model uncertainty, all plausible models would predict the same system response for the same model parameter set. Therefore, if an analysis goal is to estimate the magnitude of model uncertainty from the least number of models, it is important to consider plausible models whose responses deviate the greatest from one another. The measure of model diversity, however, must be based upon the modeling assumptions since one does not know a priori the responses of potential model choices. Ultimately, plausibility and diversity must be traded off in arriving at a set of considered models.

The proposed approach for dealing with model uncertainty is to consider multiple realizations of the modeling process which produce different forms of the model. The analyst can then perform simulations with these different models and compare their outputs. Thus, it is similar to nondeterministic analysis, where simulations are performed for multiple values of a variable parameter. In nondeterministic analysis, however, enormous efficiencies are gained by using the same programmed discrete model for all values of a variable parameter. Essentially the costs of nearly the entire modeling and discretization process are leveraged in most nondeterministic analyses. In assessing model uncertainty this is usually not the case. The discretization and programming of a model can be a user-intensive and time-consuming task which is highly dependent on the mathematical model form. Each model form considered requires a new set of definition, discretization and programming tasks. Hence, the cost of developing, discretizing and programming multiple models is a significant barrier to treatment of model uncertainty. Because of limited resources, the analyst must be judicious in his or her choice of models.

Model structure, continuous and discrete parameters

To best proceed with this treatment of model uncertainty, the following definitions are offered. A model M_i is a mapping which describes how input variables x and output variables y are related. M_i consists of two parts: a *structure* S_i which is founded on modeling assumptions, and *parameters* $\{\theta\}_i$ which have meaning based on a given structural choice S_i . The set of all model structures relevant to the problem is termed the *model space* S . The model space is effectively infinite with each discrete point in the space representing a structural choice with some level of plausibility. The set of models which result from an understanding of the modeling and simulation process are necessarily a subset of all possible model structures. The more thorough and comprehensive the understanding of the modeling and simulation process is, the more representative the model realizations will be of the total model space. Without such an understanding, an assessment of "total" uncertainty could be potentially misleading. A taxonomy of uncertainty and error sources is a necessary guide to understand the limits of a numerical assessment of total prediction uncertainty.

Model parameters have meaning within the context of a given model structure. Many parameters of interest, however, will transcend a specific model structure. Normally, many of the model parameters in first principles-based engineering simulation will be parameters of the conceptual model, derived directly from properties of the physical system being studied. Therefore, these parameters will probably have the same meaning under all model structures being considered and can be termed *global parameters* for the present discussion. Other parameters will be dependent on the structure of auxiliary submodels, such as constitutive parameters or fluid dynamics turbulence model coefficients. These shall be referred to as *submodel parameters*. Finally, some parameters will be defined as part of a numerical solution method, such as a grid clustering parameter. These shall be termed *numerical parameters*.

Global parameters will typically have variability arising from manufacturing processes, environment, and other uncontrolled external influences to the system. Submodel parameters are somewhat different in that their variability, in addition to being driven by the same factors as global parameters, are also a function of the quality of their empirical estimation. That is, submodel parameters are usually estimated from experimental data within the context of the submodel form. For example, a submodel parameter might be the exponent in a material model of a braze alloy. Then load-strain and strain rate data would be used to estimate the exponent, whose estimated value would have variability arising from variability in physical specimen behavior, uncertainty in the measured data, and differences between the data and the best model fit. Numerical parameters will not have variability but are subject to uncertainty, although that uncertainty is rarely modeled.

There can be a certain lack of distinction between model structure and parameters. Some might include within structural assumptions such issues as a linear relationship versus a cubic nonlinear relationship. Others might simply generalize the problem to include the exponent in question as a continuous parameter, thereby replacing a structural uncertainty with a parametric uncertainty. A more useful distinction can be drawn by considering as parameters those elements of the model which are continuous. That is, parameters which can take on a continuum of values. Then model structures are defined by discrete structural assumptions. Therefore, model structures are not related to one another in a continuous sense; that is, by infinitesimal perturbations.

Types of uncertainties and errors in model structure

A diagram of the framework (presented in Appendix A) for the sources of uncertainty and error in simulation is shown in Figure 2. Before proceeding, it is helpful to consider the types of uncertainties and errors which bear on model structure or arise from the modeling process. As mentioned previously, this research is focused mainly on the mathematical modeling, discretization and numerical solution phases. This taxonomy of uncertainty and error sources drives the assessment of model uncertainty since it defines the scope or dimension of the model space.

Mathematical Modeling

- Conservation equations - (2D vs. 3D), competing plausible conservation laws
- Submodels - form of law/equation, linear vs. nonlinear, etc.
- Loads, Boundary conditions, Initial conditions - infinite vs. finite boundary

Discretization

- Spatial - element selection, mesh topology
- Temporal - integration rule
- Loading - lumped vs. consistently integrated
- Boundary conditions - infinite element formulation
- Coupled fields - fully coupled vs. partitioned, stabilization

Numerical Solution

- It could be argued that parameters such as mesh size are discrete because they cannot take on a continuum of values except in the limit as the mesh size approaches zero. The same could be said of the time step if an integral number of time steps are required between two fixed times.

Propagation tree structures for model realizations

It is helpful to visualize the impact of model uncertainty, as well as other error sources, in terms of a propagation tree. Here it is suggested that each "leaf" of the tree includes a deterministic model structure in which some continuous parameter variability may be implicit. Therefore, each leaf implies a nondeterministic analysis with respect to

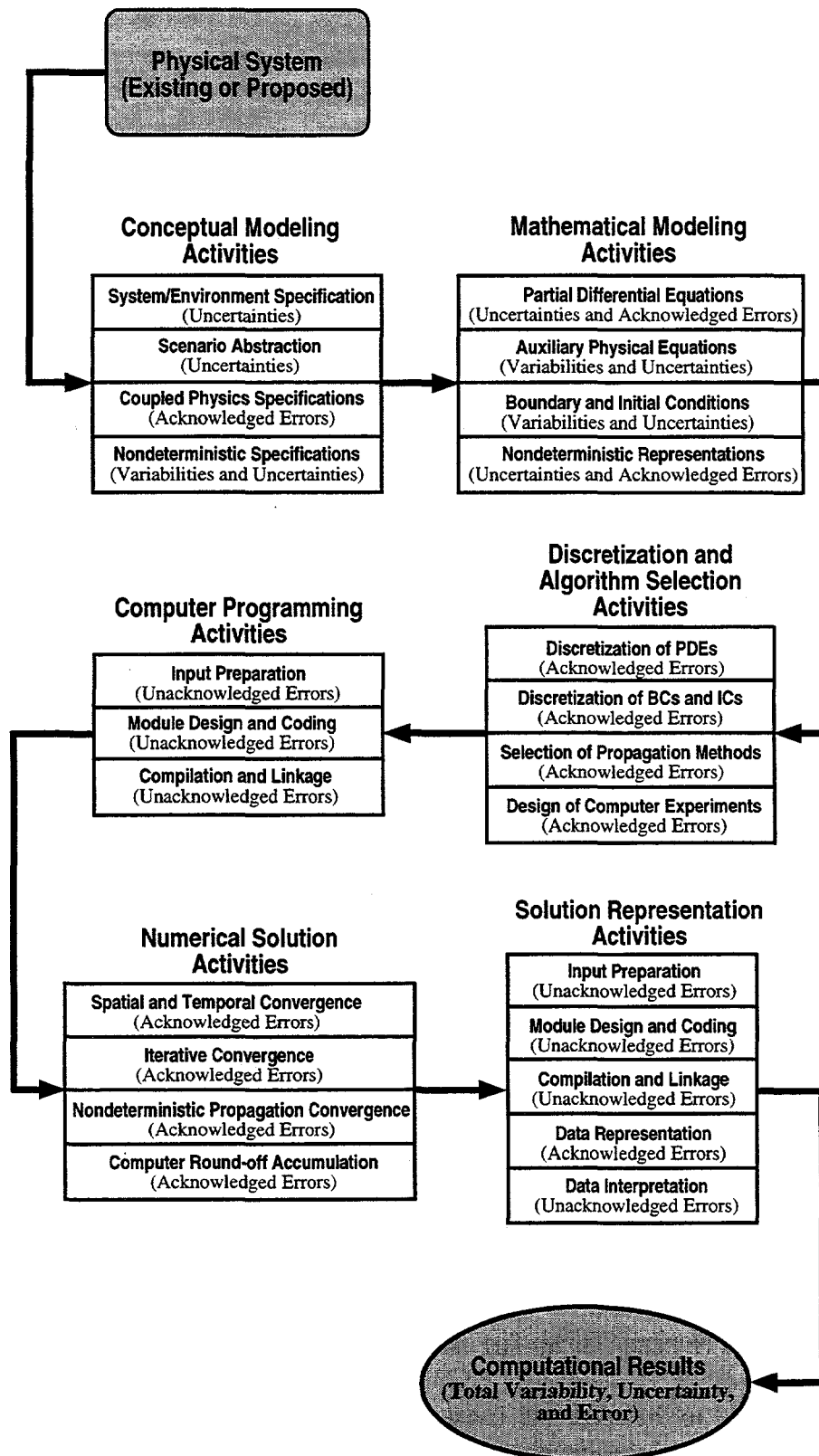


Figure 2: Framework for Sources of Uncertainty and Error in Computational Simulation

some set of continuous model parameters, and the methodologies developed for non-deterministic systems, such as sampling and reliability methods, would be applicable to each of these systems. Figure 3 illustrates a section of this tree, in which at each phase of the modeling and simulation process only one configuration of assumptions is expanded upon. Note that each level of the tree corresponds to a phase of the modeling and simulation process, illustrated in Figure 2. Note also that the errors due to spatial and temporal mesh size are not treated as leading to different model structures because of the specific error models available for these sources. The need to quantify these errors may lead to a specific number of analyses performed on different meshes, but these analyses can be regarded as similar to the parameter sampling used to address continuous parametric variabilities via Monte Carlo analysis or in building response surfaces.

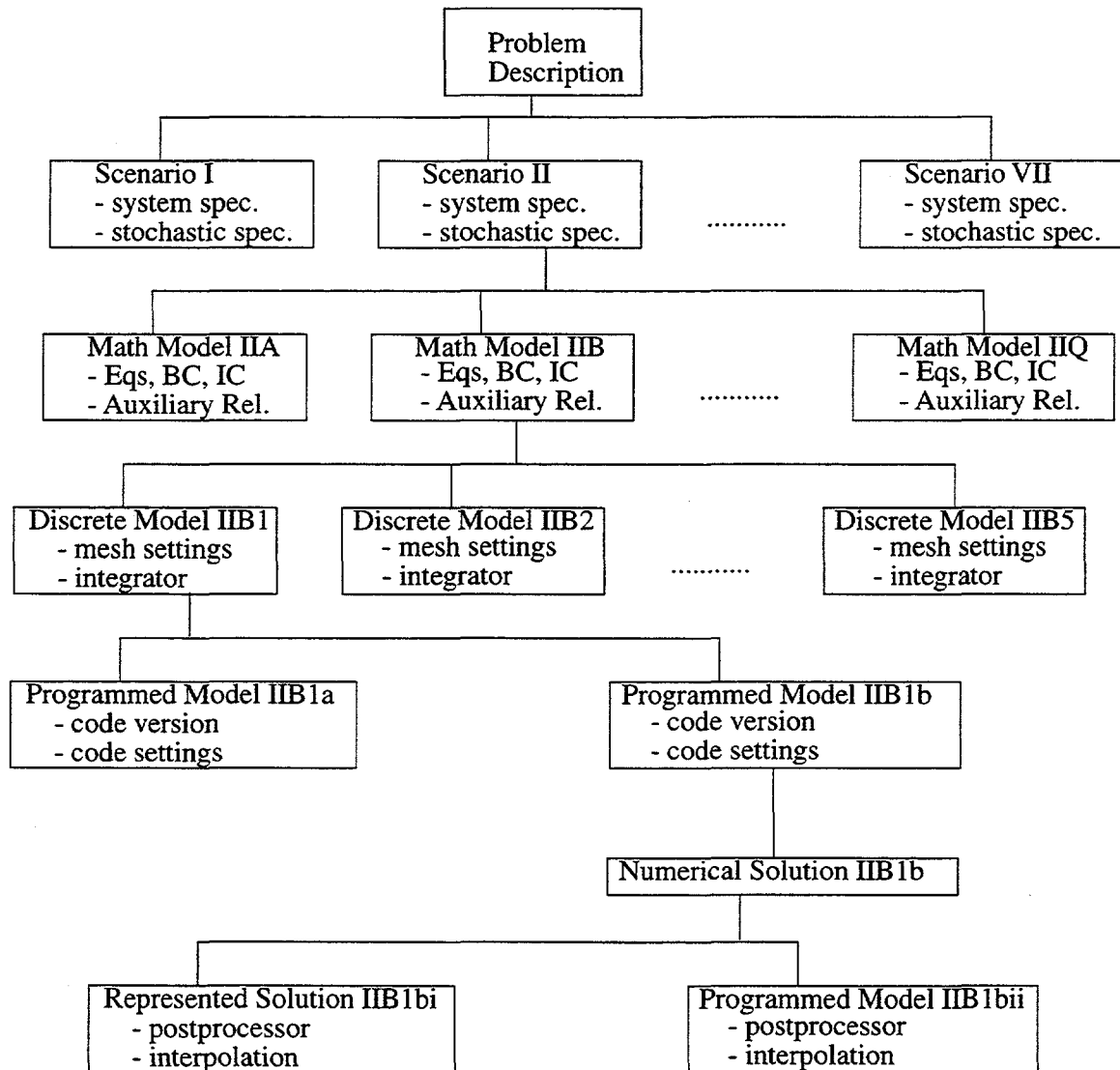


Figure 3: Propagation Tree of Model Structures

Project Activities and Contributions in Identification of Uncertainty and Error Sources in Computational Simulation

The present LDRD project, leveraged with an Engineering Sciences Tech Base project on framework definition for uncertainty quantification, performed literature searches and brainstorming on such problems as the prediction of outcome for a weapon in an aircraft crash and fire. The result of this work is a framework for uncertainty and error sources detailed in Appendices A and C and used in the previous discussion on the concepts of uncertainty and error in model structure. Appendix C provides the most thorough and detailed discussion on this work.

3. Assessment and Propagation of Model Uncertainty

Introduction

In Section 2 and Appendices A and C, the problems associated with selection of mathematical model form and the errors and uncertainties which arise from it were discussed. Furthermore, the different forms of acknowledged errors and how simplification errors can result in effective model uncertainty were also addressed. Finally, an approach for treating or assessing model uncertainty which involves the consideration of different model forms arising as different realizations of the modeling process was proposed. In order to proceed, it is now necessary to develop a mathematical framework for the consideration of multiple models. This mathematical framework is based on probability and statistics, specifically Bayesian statistical inference [1,5,6,7,8,9,10,11]. The development here follows from work first presented by Draper [1]. This section begins by studying the use of statistical inference for propagation of uncertainty in a single model with continuous parameters. This result is then extended to the treatment to a continuous space of models, and from there to a discrete set of models. In Bayesian inference, prior probabilities are utilized to make inferential predictions regarding future events for which there is no data. These prior probabilities are often based on experimental data but may be subjective, or a combination of subjective judgement and statistical results. Furthermore, by employing Bayes Theorem, it is possible to revise the probabilities for uncertain elements of the problem when new data, such as that obtained by phenomenological and validation testing, becomes available. The revised or posterior probability distributions generally improve the predictive accuracy of the inference, assuming that the uncertainties in the data are handled properly.

Statistical Inference using Prior Probabilities

Consider a simple, linear, single output-single parameter model, viz.

$$y = y_0 + M\theta \quad \theta \sim N(\mu_\theta, \sigma_\theta^2) \quad (1)$$

where M can be thought of as a sensitivity arising from the functional form for the model, perhaps from an integral over a physical domain. θ is a parameter whose variation is described by a normal distribution with mean μ_θ and variance σ_θ^2 , y is the system output which the model predicts, and y_0 is a constant. Statistical inference rests on determining the probability density function (p.d.f.) of the predicted quantity y , $p(y)$. Determining $p(y)$, which characterizes the prediction uncertainty, requires the conditional probability of y given θ (which is called the likelihood function), the p.d.f. of θ , and an integration

over the domain of θ :

$$p(y) = \int p(y|\theta)p(\theta)d\theta \quad (2)$$

The likelihood function $p(y|\theta)$ is given as

$$p(y|\theta) = \begin{cases} 1 & \text{if } y = y_0 + M\theta \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The (prior) probability for θ is

$$p(\theta) = \frac{1}{\sqrt{2\pi\sigma_\theta^2}} e^{-\frac{1}{2}\left[\frac{(\theta - \mu_\theta)^2}{\sigma_\theta^2}\right]} \quad (4)$$

Then $p(y)$ is given by

$$p(y) = \frac{1}{\sqrt{2\pi(M^2\sigma_\theta^2)}} e^{-\frac{1}{2}\left[\frac{(y - (y_0 + M\mu_\theta))^2}{M^2\sigma_\theta^2}\right]} \quad (5)$$

which is just a normal distribution with mean $(y_0 + M\mu_\theta)$ and variance $M^2\sigma_\theta^2$. It can be proved, in fact, that linear operations on Gaussian random vectors produce Gaussian random vectors. Therefore, it is also possible to solve the above problem by deriving the mean and variance of y :

$$\begin{aligned} E[y] &= E[y_0 + M\theta] & V[y] &= E[(y - E[y])^2] \\ &= E[y_0] + E[M\theta] & &= E[M^2(\theta - \mu_\theta)^2] \\ &= y_0 + ME[\theta] & &= M^2E[(\theta - \mu_\theta)^2] \\ &= y_0 + M\mu_\theta & &= M^2\sigma_\theta^2 \end{aligned} \quad (6)$$

Before proceeding, note that the previous analysis is nothing more than a closed form solution to a nondeterministic analysis problem. These results could have equivalently been obtained using numerical techniques such as Monte Carlo sampling, response surface methodology, or analytical/reliability methods. This example is presented in some detail so that meanings of the probability density functions would be clear, and so that the reader can begin to develop a working understanding for these types of problems.

Revising Prior Probabilities to Account for New Data

Now suppose that a single test is performed yielding a value x which can be predicted using a version of the model modified to account for the variance in the test. Therefore

$$x = y_0 + M\theta + \varepsilon \quad (7)$$

where the test variance is modeled as $\varepsilon \sim N(0, \sigma_x^2)$, a normal distribution with zero mean and variance σ_x^2 . The objective is to compute a new distribution for y which accounts for the additional data x ; that is, the conditional probability density function $p(y|x)$ is desired:

$$p(y|x) = \int p(y|\theta, x)p(\theta|x)d\theta \quad (8)$$

This however requires the so-called posterior distribution for θ , $p(\theta|x)$, which is the probability distribution for θ conditioned on the new data x . This is determined by employing Bayes rule:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \quad (9)$$

These elements are given by

$$\begin{aligned} p(x|\theta) &\sim N(y_0 + M\theta, \sigma_x^2) \\ p(\theta) &\sim N(\mu_\theta, \sigma_\theta^2) \\ p(x) &\sim N(y_0 + M\mu_\theta, \sigma_x^2 + M^2\sigma_\theta^2) \end{aligned} \quad (10)$$

Plugging Eq. (10) into Bayes rule and simplifying, the posterior distribution is given as¹²

$$p(\theta|x) = N((\mu_\theta)^*, (\sigma_\theta^2)^*) \quad (11)$$

where

$$(\mu_\theta)^* = (\sigma_\theta^2)^* \left[\frac{\hat{\theta}}{\sigma_\theta^2} + \frac{M(x - y_0)}{\sigma_x^2} \right] \quad (\sigma_\theta^2)^* = \left[\frac{1}{\sigma_\theta^2} + \frac{M^2}{\sigma_x^2} \right]^{-1} \quad (12)$$

Finally, for the prediction y ,

$$p(y|x) \sim N(y_0 + M(\mu_\theta)^*, M^2(\sigma_\theta^2)^*) \quad (13)$$

Before proceeding, consider what this result represents. The mean of the revised distribution for y is the previous mean plus a correction $M((\mu_\theta)^* - \mu_\theta)$ due to the change in the mean of the distribution of θ . Also examine the revised variance for the distribution of θ . The inverse of the revised variance $(\sigma_\theta^2)^*$, which can be thought of as an absolute confidence parameter, is equal to the previous confidence plus a measure of the increase in information provided by the new data. The information in the new data is the square of the sensitivity of x to θ , times the confidence in the data, given as the inverse of the variance in the data. Thus, the incorporation of additional data adds information (if the confidence in the data is relatively high) to the knowledge of the parameter θ , which is reflected by a narrowing of the distribution and possibly a shift in the mean.

A number of questions can be posed: is it reasonable to update the distribution of θ in light of the new data? On what basis is this new data judged relative to the bulk of previous data or experience inherent in the prior distribution? And what about the values of M and y_0 ? Is there enough certainty in those values that all the "blame" in the predictive error should be assigned to the distribution on θ ? Now there is new data which provides an updated mean and variance of θ which might be significantly different than the prior distribution. What does this imply about the true distribution of θ and about the model?

To address these questions, first examine the assumptions in the previous analysis. Basically, the assumptions are the distributions on the model parameters and the data, and the form of the model which is implied in M . The origin of the distribution for θ was not stated; ideally it was determined from an extensive quantity of data which reflects all of the physical sources of variability in the parameter, and is free of bias in the measurement of the data. It is very possible, however, that the mean of the distribution is a "nominal" value taken from a handbook of physical or material parameters, or from a design document for the system. The variance of the distribution might be based on a dimensional tolerance or simply on an estimate by the analyst. Suppose that the values for the prior mean and variance of θ came from a reliable database of experiments designed to measure θ and account for experimental uncertainty and bias.

The other assumed distributions are for the data and the parameter y_0 . The distribution in the data should be viewed as the distribution of many tests for which the value of θ is known but the noise and all other unmodeled variabilities take on values in accordance to their distribution functions. If these other sources of variabilities (and errors) are not either modeled and estimated, or if an insufficient number of actual tests are performed to experimentally quantify σ_ϵ , it is very possible to overestimate the confidence in the data and thus to "overfit" the model. Even if the data are accompanied by a reliable expert opinion about the variance of the data, there could be significant uncertainty about

the mean. After all, if only one data point is known, it might be two or more standard deviations away from the actual mean.

Finally there is the parameter y_0 and the model which determines M . The distribution for the parameter y_0 is in fact not distributed at all but simply a delta function; the potential for error and overconfidence given this assumption is probably obvious. Then there is the assumption of model form, or in this simplistic case the value of S . It could potentially take on a different value if different modeling assumptions were used. The fact is that uncertainty in the model is the most likely cause of a posterior distribution for θ which is significantly different than a prior distribution based on a reliable database. So, if there is a lack of confidence in the form of the model, how should this uncertainty analysis proceed? Draper's approach is to treat the model structure itself as an additional source of uncertainty (modeled in a probabilistic fashion), and to perform inference in the same Bayesian fashion as presented herein. This is detailed in the next section.

Propagation of Model Uncertainty: Bayesian approach

Draper [1] demonstrated that the previous Bayesian treatment of parametric uncertainty can be extended to the problem of model uncertainty. To begin, simply reverse the roles of M and θ in the previous discussion. If M can take on a continuous range of values based upon changes in the model structural assumptions, then there is a need only to assign or determine a prior probability density function for M , $p(M)$, and proceed using exactly the same methodology. Therefore

$$p(y) = \int p(y|M)p(M)dM \quad (14)$$

Note that since in the previous discussion M was assumed to be deterministic, and the roles of M and θ are now reversed, it is now assumed that the parameter θ is deterministic. Shortly the treatment will be generalized such that both M and θ are treated as non-deterministic quantities. First, however, it is necessary to adjust the prior analysis for the case where there is a discrete and finite set of model structures.

For a finite set $S = \{S_1, \dots, S_m\}$ of model structures with only deterministic (i.e. not variable) parameters

$$p(y|S) = \sum_{i=1}^m p(y|S_i, S)p(S_i|S) \quad (15)$$

where it is noted that the prediction y is conditioned on the finite set S of model structures considered. Here we have dropped the parameter M , which is simply the sensitivity of y with respect to the model structure S_i and is assumed to be deterministic. The resultant

distribution for the predicted quantity y might look like that in Figure 4. Here there are 5 model structures considered with probabilities (0.30, 0.25, 0.20, 0.15, 0.05, 0.05). As required, the sum of the model probabilities is unity. What about other possible model

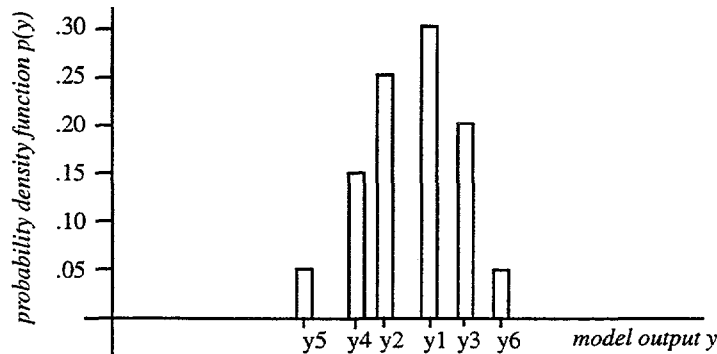


Figure 4: A distribution of a discrete set of deterministic models

structures beyond the 6 considered in this example? The probabilistic interpretation is that all other possible model structures have been assigned a probability of zero.

When the inference is also conditioned on some new data x

$$p(y|x, S) = \sum_{i=1}^m p(y|x, S_i, S)p(S_i|x, S) \quad (16)$$

The determination of posterior probabilities $p(S_i|x, S)$ (i.e. the updated probabilities conditioned on the new data) is again from Bayes Rule:

$$p(S_i|x, S) = \frac{p(x|S_i, S)p(S_i|S)}{p(x|S)} \quad (17)$$

where $p(x|S) = \sum p(x|S_i, S)p(S_i|S)$. Note that if a model is assigned a prior probability of zero, its posterior probability will also be zero. Thus, no matter how well a model structure might accurately fit some new data, if its prior probability is zero (possibly because the model structure is completely implausible from what is known of the physics), it cannot gather support from the new data. This can be a good feature of the Bayesian treatment, because it precludes giving weight to empirical data models simply because they optimally fit a limited data set. On the other hand, model structures with small but nonzero prior probabilities might gain significant support from new validation testing, but if those models are not treated (effectively setting their probabilities to zero) this change in support for the models cannot be determined. This is why the probability density functions in the preceding equations are dependent on the assumed discrete model space S .

To summarize, the Bayesian treatment of model uncertainty involves

- the treatment of the mathematical model structure or form as a nondeterministic quantity
- the selection of a discrete set of models to consider
- the modification of statistical inference mathematics from continuous variables and distributions to discrete variables and probabilities

Clearly, the value of this formulation and the results it provides rests on the model probabilities $p(S_i|S)$ assumed for the space of models. The problem of assigning model probabilities will be discussed in more detail near the end of this section. First, this treatment must be generalized to include both parametric and structural uncertainties. This is the subject of the next subsection.

Combining model uncertainty with parametric variability

The most general case is where both structural uncertainty and parametric uncertainty are present. Parametric uncertainty is often conditioned on the model structure. To propagate these uncertainties, it is necessary to integrate over the parameter variations and then sum over the model structures. This suggests a nested uncertainty structure where the model uncertainty generates a higher level of uncertainty analysis into which the nondeterministic analysis of each model provides distributions:

$$\begin{aligned}
 p(y|x, S) &= \sum_{i=1}^m \left\{ \int_{\theta} p(y|x, S_i, \theta, S) p(\theta|x, S_i, S) d\theta \right\} p(S_i|x, S) \\
 &= \sum_{i=1}^m p(y|x, S_i, S) p(S_i|x, S)
 \end{aligned} \tag{18}$$

where

$$p(y|x, S_i, S) = \int p(y|x, S_i, \theta, S) p(\theta|x, S_i, S) d\theta \tag{19}$$

Compare Eq. (19) to Eq. (8). This is exactly the same treatment of parametric uncertainty as was developed previously except that now all quantities are explicitly conditioned on the structural choice S_i and the model space S . Furthermore, by assigning relative probabilities to the S_i and summing the results of all models in the model space, a general treatment for both model structure and parameter uncertainty is obtained. This formulation is sometimes referred to as Bayesian Model Averaging (BMA) [9]. The inferential distribution with model uncertainty is given by a weighted average of the inferential distributions of all the considered models, where the weights are the prior or posterior model structure probabilities $p(S_i|S)$. The resultant distribution is shown in Figure 5. Here each of the

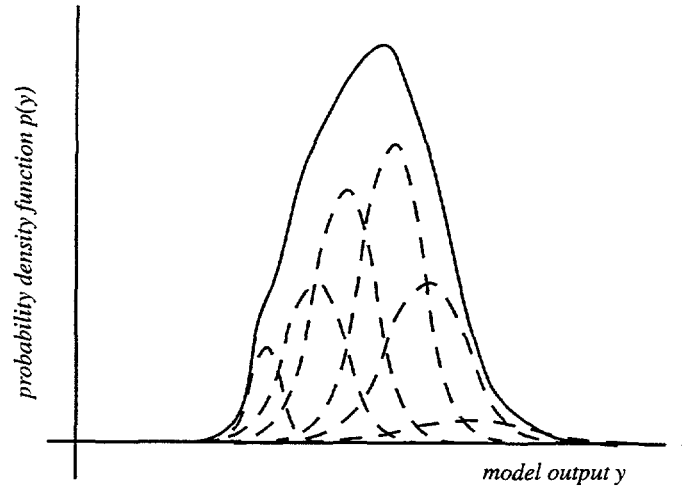


Figure 5: A distribution of a discrete set of nondeterministic models

models provides a distribution for the predicted output, weighted by the models' probability weights. These are shown in the dashed curves. The composite distribution of the model set is the sum of these and shown in the solid curve. Compare the effective uncertainty in the output (the spread of the distributions) of the composite distribution to the uncertainties of the individual models. Generally, the total uncertainty, including model uncertainty, is much greater than the uncertainty predicted by any single model. This is not guaranteed to be true, however. In the model averaging approach, it is possible to have a considered model which predicts a higher variance than the composite distribution. This only occurs, however, when that model has a very small structural probability. Given this fact, that model would not be the chosen model if only a single deterministic model form were chosen.

Finally, for updating prior probabilities, Eq. (17) is used with the generalization

$$p(x|S_i, S) = \int p(x|S_i, \theta_i, S) p(\theta_i|S_i, S) d\theta_i \quad (20)$$

The formula for updating the prior model probability Eq. (17) can also be written as

$$p(S_i|x, S) = \frac{p(S_i|S)}{\sum_j \left[\frac{p(x|S_j, S)}{p(x|S_i, S)} \right] p(S_j|S)} = \frac{p(S_i|S)}{\sum_j p_{ij} p(S_j|S)} \quad (21)$$

where

$$\rho_{ij} = \frac{p(x|S_j, S)}{p(x|S_i, S)} \quad (22)$$

is the *Bayes factor* for comparing structure S_j against structure S_i . Although the Bayesian model averaging approach is a relatively new development in applied statistics, Bayes factors have been used for a much longer time as a basis for selecting competing models or theories. Hence, some literature exists on efficient computation or approximation of Bayes factors for the purposes of model selection. The applicability of these techniques for the present purposes is not yet clear. What is clear is that, given sufficient resources, the analyst can apply the same nondeterministic analysis techniques for single model forms within the framework of Bayesian model uncertainty analysis.

Once model probabilities are determined, or assumed, it is straightforward to apply them to the assessment of model uncertainty. The resultant distribution for the predicted quantity y is simply a weighted average of the distributions resulting from each of the model structures considered. The weightings for the space of considered model structures always sum to unity. Note that while the distributions from each model may be normal or at least unimodal (having only one maximum), the superposition of a number of distributions is likely to appear non-normal as in Figure 5 and possibly multimodal. Some simple statistics can be computed, however. For example, the mean and variance of the prediction y is given as¹

$$E[y|x, S] = \sum_{i=1}^m p(S_i|x, S)E[y|x, S_i, S] \quad (23)$$

$$V[y|x, S] = \sum_{i=1}^m p(S_i|x, S)V[y|x, S_i, S] + \sum_{i=1}^m p(S_i|x, S)(E[y|x, S_i, S] - E[y|x, S])^2$$

The uncertainty in the prediction y , as measured by its variance, is decomposed into the sum of two terms. The first is the weighted average of the variances predicted by each considered model (the expected internal model variability). The second term is more or less a direct measure of the model uncertainty itself, as it is a measure of the spread in the estimated means across the space of considered models (the model-to-model variability).

Assignment of prior subjective model probabilities

To make the preceding treatment of model uncertainty effective, there is a need to develop ways of determining prior model probabilities. After all, these are fundamental assumptions which underlie the Bayesian methodology for model uncertainty. In fact, since the probabilistic interpretation of multiple models may be viewed somewhat critically, its more helpful to view this methodology as model averaging, and to view the model

probabilities as weights which reflect the relative plausibility of each model. These plausibility weightings are highly subjective, even when conditioned on some body of data. Model plausibility is a function of accuracy, consistency, and completeness. Model accuracy is a judgement of how the sum total of all sources of error affect the predictive quality of the model. Discretization error may or may not play a role in this judgement, depending on whether it is treated by other error quantification methodologies. Model consistency refers to whether the model form is consistent with what is known about the governing physics in the system response. It is possible for an empirical model to possess apparently high accuracy with respect to known data but to have a form which is completely unrelated to the physics. Extrapolation, and even interpolation, with this model might lead to significant errors. Therefore, given two models of apparently equal accuracy with respect to known data, one should prefer for predictive purposes the model which is the most consistent with the mathematical form of the physics (subject to the trade-off between simplification error and numerical solution error mentioned in Section 2). Finally, model completeness refers to a model's ability to treat the important sources of variability in a system. There might, for example, be a closed form solution for a system of interest which is highly accurate and consistent. The model's drawback, however, may be that its form does not admit certain physical variations in a problem. In a sense, both consistency and completeness are issues of accuracy, but they are stressed as separate issues because it is easy to misinterpret a model's accuracy by neglecting to consider all potential sources of error when extrapolating from a region of quantified accuracy.

Another consideration in the assignment of model probabilities is ensuring that they are decomposed consistently at each level in a propagation tree. In fact, the propagation tree itself can help guide the assignment of model probabilities because of this constraint. To illustrate, consider Figure 6, in which a single scenario is considered, with alternate models arising at the mathematical modeling and discretization phases. It can be seen that the probabilities of each model structure must sum to the probability of the higher level model it supports. Alternate mathematical models would be assigned probabilities in accordance to the issues of accuracy, consistency and completeness described previously. Then if those mathematical models are further decomposed into competing discretizations, those structures "inherit" the probability of their parent model.

Selection of Models

Selecting models to consider in a model uncertainty assessment is not unlike selecting input locations in an experiment design. If the problem is viewed from a Monte Carlo-like standpoint, models should be selected purely on the basis of relative plausibility. As with Monte Carlo, however, it may take a high number of model "samples" or "trials" to begin to arrive at a useful assessment of model uncertainty. For this reason, the aspects of diversity and efficiency are also important. Think of diversity as a subjective judgement of how different a model is from other models on the basis of its structural assumptions. Efficiency simply refers to the amount of effort required to perform uncertainty analysis on

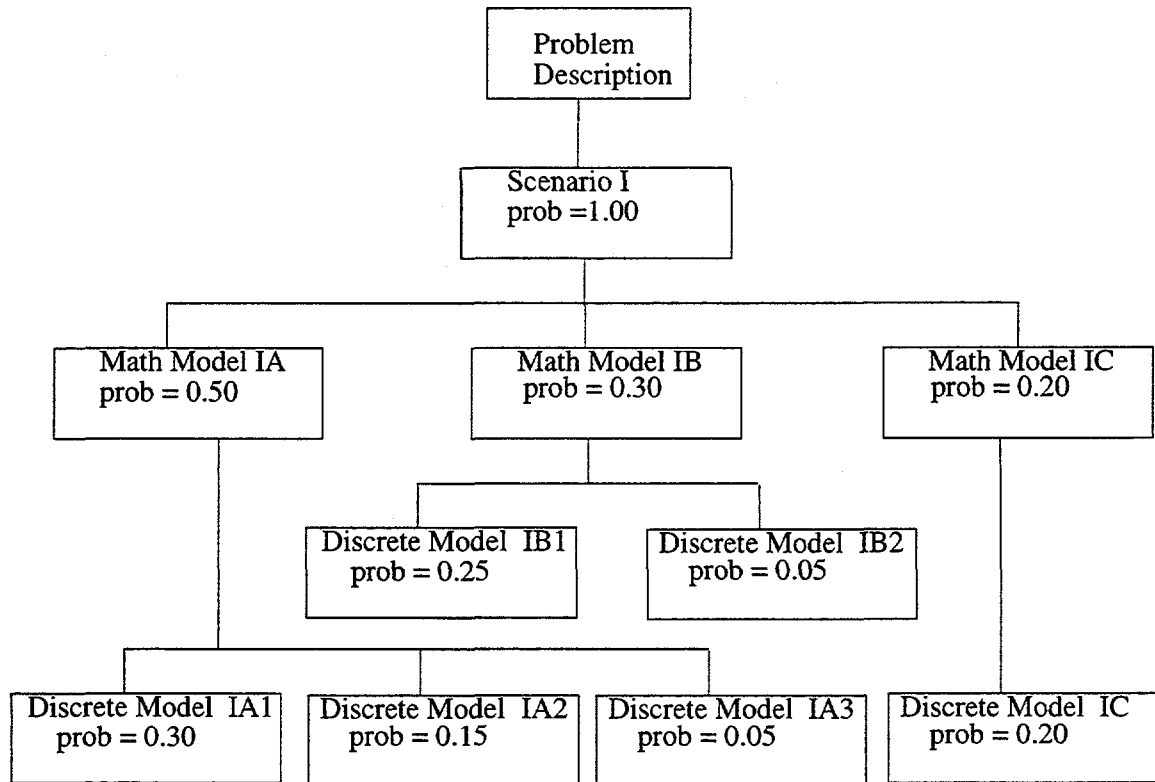


Figure 6: Decomposing model probabilities in a propagation tree structure

a particular model space. When all other considerations are equal, models should be chosen which can be assessed with the least expenditure of resources.

Given limited resources, it is inefficient to evaluate models with probabilities below some user-defined threshold. For those models above the critical threshold, models could be further prioritized and the accuracy level of the nondeterministic analyses adjusted in accordance to the model probabilities. This should be done, however, while keeping in mind the relative diversity and efficiency of each model, especially if some models are predicting responses close to a failure region for the system.

Project Activities and Contributions on Assessment and Propagation of Model Uncertainty

The present LDRD project investigated the use of Draper's Bayesian approach to assessment and propagation of model uncertainty by application to a modal analysis of a neutron generator mounting bracket. This analysis included both forward propagation and inverse analysis (through the determination of the posterior model probability distribution)

and is detailed in Appendix A. In addition, under project funding, an investigation was performed on the use of metamodels for the assessment of model uncertainty. In this work, discrete model form parameters were introduced to the response surface methodology of uncertainty quantification, and then used to propagate model uncertainty. An interesting aspect of this work was the nesting of continuous metamodel parameters to reflect their model form-dependent definition. This work is detailed in Appendix D. Finally, a detailed uncertainty analysis of a missile in flight was performed and is detailed in Appendix C. Part of this analysis considered issues of model form arising from differing propulsion system models, as well as comparing 3 degree-of-freedom and 6 degree-of-freedom models of rigid body flight response with associated continuous parameter variabilities and numerical solution error. This analysis did not consider the use of model probabilities, but rather a small range of model forms individually.

Three additional aspects of model uncertainty were considered but have not yet been investigated. One aspect is how differing model forms can be treated within the Bayesian framework when the models treat different and incomplete regions of the total variability space; that is, the models have different parameter sets. The proposed approach is to define a master response surface which characterizes the entire variability space of the problem. The coefficients of this master response surface would be determined via analysis of each considered model form. In the regions where multiple models are defined, the coefficients of the master metamodel would have two components of uncertainty. One component would be due to the fitting of the coefficient to the response space. The other component would be due to model uncertainty and would express the difference between the value of the coefficient as determined by the multiple models. A variation of this approach is currently being investigated in a small project at the University of Wisconsin funded by this LDRD. In this approach, an alternate response surface methodology is proposed in which multiple metamodels are synthesized into a master response surface. The goal is to integrate multiple models which fit different regions of the variability space into a global model for uncertainty propagation. While this approach is more oriented towards improvement of response surface methodology, the implementation can equally apply to the aforementioned problem of integrating multiple model forms with different parameter sets.

Another important aspect of model uncertainty which remains to be investigated is the issue of model space diversity and completeness. As noted in this section, alternative model structures are assigned a probability of zero in the Bayesian model uncertainty propagation approach if the models are not being considered. In order to converge most quickly to an estimate of model uncertainty, it is important to stake out the extremes of the model space. There is, however, no metric to measure the adequacy of the model space considered. There is a need to determine metrics for the model space based on the relative correlations between alternate model predictions, as well as the posterior likelihood of relevant physical data given the predictive distribution provided by the model space.

Finally, non-probabilistic approaches to modeling structural uncertainty have been identified but not yet investigated. Foremost amongst these are convex nested sets of models (as proposed by Ben-Haim [13,14,15]), in which uncertainty could be defined as a region of prediction space relative to the deterministic model. Thus, the model prediction is defined as any point within the convex set, which is itself parameterized by a model uncertainty coefficient of undetermined size. The model uncertainty coefficient can either be estimated in an inverse sense as the degree of model uncertainty for which inferences are “robust”, or in a forward sense as the norm of prediction error indicated by validation experiments.

Additional approaches which are attempting to span the chasm between probability theory and set theory are generally grouped into evidence theory [16]. This encompassing theory, which is still under major development, was originally discovered by Dempster [17] and Shafer [18]. Coming from the probabilistic root of evidence theory is Bayesian inference and imprecise probability theory [19]. Coming from the set theory root of evidence theory is possibility theory [20] and fuzzy set theory [21]. Of all the various theories under the umbrella of evidence theory, only Bayesian inference is mature and in widespread use. We simply call the reader’s attention to other approaches to representing model uncertainty. The present research effort, however, did not pursue these other approaches in any depth.

4. Estimation of Discrete Numerical Solution Errors

The problem of interest is the solution of partial differential equations (PDEs) over continuous spatial and temporal domains. An analytical solution to these equations and the associated initial and boundary conditions is generally not known, so the solution must be approximated by some numerical approach. The basis for the finite difference methods is the construction of a discrete grid and the replacement of the continuous derivatives in the governing PDEs with equivalent finite difference expressions. For example, the governing equation for unsteady heat conduction in a rod is [22]

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0 \quad (24)$$

This can be solved by replacing the time derivative with a forward difference approximation and the spatial derivative with a centered difference approximation. This is then the classic forward time, centered space (FTCS) scheme at spatial grid location x_j and time step t_n is given as

$$T_j^{n+1} = sT_{j-1}^n + (1 - 2s)T_j^n + sT_{j+1}^n \quad (25)$$

where $s = (\alpha\Delta t)/(\Delta x^2)$, and the spatial and temporal grid spacings are Δx and Δt , respectively. To determine the truncation error for this approximation, each term of the FTCS approximation is expanded by a Taylor series:

$$\begin{aligned} T_j^{n+1} &= T_j^n + \left[\frac{\partial T}{\partial t} \right]_j^n \Delta t + \frac{1}{2} \left[\frac{\partial^2 T}{\partial t^2} \right]_j^n (\Delta t)^2 + O(\Delta t^3) \\ T_{j-1}^n &= T_j^n - \left[\frac{\partial T}{\partial x} \right]_j^n \Delta x + \frac{1}{2} \left[\frac{\partial^2 T}{\partial x^2} \right]_j^n (\Delta x)^2 - \frac{1}{6} \left[\frac{\partial^3 T}{\partial x^3} \right]_j^n (\Delta x)^3 + \frac{1}{24} \left[\frac{\partial^4 T}{\partial x^4} \right]_j^n (\Delta x)^4 + O(\Delta x^5) \\ T_{j+1}^n &= T_j^n + \left[\frac{\partial T}{\partial x} \right]_j^n \Delta x + \frac{1}{2} \left[\frac{\partial^2 T}{\partial x^2} \right]_j^n (\Delta x)^2 + \frac{1}{6} \left[\frac{\partial^3 T}{\partial x^3} \right]_j^n (\Delta x)^3 + \frac{1}{24} \left[\frac{\partial^4 T}{\partial x^4} \right]_j^n (\Delta x)^4 + O(\Delta x^5) \end{aligned} \quad (26)$$

Simplifying leads to

$$\left[\frac{\partial T}{\partial t}\right]_j^n \Delta t - s \left[\frac{\partial^2 T}{\partial x^2}\right]_j^n (\Delta x)^2 + \frac{1}{2} \left[\frac{\partial^2 T}{\partial t^2}\right]_j^n (\Delta t)^2 - \frac{s}{12} \left[\frac{\partial^4 T}{\partial x^4}\right]_j^n (\Delta x)^4 + O(\Delta t^3, \Delta x^6) = 0 \quad (27)$$

or equivalently

$$\left[\frac{\partial T}{\partial t}\right]_j^n - \alpha \left[\frac{\partial^2 T}{\partial x^2}\right]_j^n + \left[\frac{1}{2} \left[\frac{\partial^2 T}{\partial t^2}\right]_j^n \Delta t - \frac{\alpha}{12} \left[\frac{\partial^4 T}{\partial x^4}\right]_j^n (\Delta x)^2 + O(\Delta t^2, \Delta x^4)\right] = 0 \quad (28)$$

Comparing this result to the original PDE, the truncation error for the FTCS scheme has leading terms proportional to Δt and $(\Delta x)^2$. Thus, the truncation error is said to be $O(\Delta t, \Delta x^2)$. This error will be small if $\frac{\partial^2 T}{\partial t^2}$ and $\frac{\partial^4 T}{\partial x^4}$ are bounded and if Δt and $(\Delta x)^2$ are sufficiently small.

The finite element method is a different numerical approach to the spatial discretization of PDEs. In the finite element method, the spatial domain is partitioned into elemental domains and within each domain the solution is interpolated according to the element shape functions. The mathematical basis for the finite element method is as a weighted residual solution to a variational form of the PDEs. For time dependent problems, application of the finite element method results in a system of ordinary differential equations in time, which can then be solved by a number of time integration methods. Error estimates in the finite element method are usually written in terms of error norms. For example, in steady state (i.e. static) linear elasticity problems, in the absence of singularities the solution error is bounded as [23]

$$\|u - u^h\| = \left[\int_{\Omega} (u - u^h)(u - u^h) d\Omega \right]^{1/2} \leq ch^{k+1} \|u\| \quad (29)$$

where u is the exact solution, u^h is the finite element approximation to u , Ω is the spatial domain over which the PDEs are defined, h is the characteristic element size, k is the order of the polynomial for the element shape functions, and c is a problem dependent constant which is independent of h and u . Generally, the order of spatial convergence depends on not only the polynomial order of the shape functions, but also the highest order of derivatives which appear in the variational form of the problem. The convergence with respect to time integration depends on the ordinary differential equation solver; i.e. the time integration method chosen.

Grid convergence and error estimation via Richardson extrapolation

Richardson [24], in 1910, described a method for increasing the rate of convergence of a numerical solution to a partial differential equation. This method is commonly referred to as Richardson's extrapolation. The basic principle was to compute the solution on two different grids and then to use those solutions to estimate an improved solution. The relationship of the discrete numerical solution to the exact solution is described as

$$u_{exact} = u_D(h) + \alpha h^p + O(h^{p+1}) \quad (30)$$

where u_D is the discrete numerical solution with mesh spacing h , p is the formal convergence rate for the method, and α is a parameter which is independent of h . Then two discrete numerical solutions on meshes with spacing h_1 and h_2 yield the following equations:

$$\begin{aligned} u_{exact} &= u_D(h_1) + \alpha h_1^p + O(h_1^{p+1}) \\ u_{exact} &= u_D(h_2) + \alpha h_2^p + O(h_2^{p+1}) \end{aligned} \quad (31)$$

Thus

$$\begin{aligned} \alpha &= \frac{u_D(h_1) - u_D(h_2)}{h_2^p - h_1^p} + O(h^{p+1}) \\ u_{exact} &= \left[\frac{h_2^p u_D(h_1) - h_1^p u_D(h_2)}{h_2^p - h_1^p} \right] + O(h^{p+1}) \\ &= u_{ED}(h_1, h_2) + O(h^{p+1}) \end{aligned} \quad (32)$$

where $u_{ED}(h_1, h_2)$ is the extrapolated discrete numerical solution based on mesh spacings h_1 and h_2 . Note that this solution is extrapolated towards the exact solution but is not equivalent to the exact solution. The formal order of the extrapolation-based method is equal to the order of the next leading error term in error model of the original discrete numerical solution.

Richardson extrapolation carries with it three basic assumptions. First, the exact solution must be sufficiently smooth so that the Taylor series expansion for the error is justified. Second, the accuracy of the method, p , is known. Third, it is critical that both mesh spacings are sufficiently small such that the computed solutions are in the formal asymptotic range. This is equivalent to stating that the leading error terms truly dominate the total discretization error. Given these assumptions, Richardson extrapolation is a powerful technique for both finite difference and finite element solutions. It is sufficiently

general that it can be applied to virtually any chosen output of the computational model, and in addition can be applied to functionals of the solution vector. Unfortunately, in the presence of singularities, the formal convergence order might not be achievable, and the actual convergence order may not be clear a priori. The singularities at issue here are not just mathematical-physical oddities; they can occur throughout problems by such modeling simplifications as sharp corners (neglecting small radii such as fillets). Thus, the presence of some singularities are likely to be the rule rather than the exception, and estimating their influence on convergence can make Richardson extrapolation a far less accurate and efficient approach.

Richardson extrapolation has traditionally been used for improving convergence order on time-independent problems, but can be extended to time-dependent problems also. Richardson extrapolation is more commonly used as an error estimator of the fine mesh solution. Roache [25] defines a *grid convergence index* (GCI), which is based on Richardson extrapolation but accounts for uncertainty in the error estimate due to uncertainty in various factors such as the effective convergence rate. The error estimates for the grid with mesh spacing h is defined as

$$\begin{aligned}
 E(h) &= u_D(h) - u_{exact} \\
 E(h) &= -\alpha h^p + O(h^{p+1}) \\
 &= \left[\frac{u_D(h_2) - u_D(h_1)}{h_2^p - h_1^p} \right] h^p + O(h^{p+1})
 \end{aligned} \tag{33}$$

but is approximated as

$$E(h) \approx \left[\frac{u_D(h_2) - u_D(h_1)}{h_2^p - h_1^p} \right] h^p \tag{34}$$

This approximation neglects the higher order terms. Therefore, for this to be a reliable estimate of the error, the mesh must be sufficiently fine so that the error is truly dominated by the leading term [26]. Furthermore, the convergence order p must also be known. Roache then defines the GCI for the mesh with spacing h as

$$GCI(h) = F_s |E(h)| \tag{35}$$

where $F_s > 1$ is used as a safety factor. Roache suggests that $F_s = 1$ is analogous to a 50% error band on data, and so recommends $F_s = 3$ as a more conservative bound on the discretization error.

In summary, Richardson extrapolation is a classical yet powerful technique for improving convergence rate and estimating errors in discrete numerical solutions of partial differential equations. It has been used more typically in conjunction with the finite difference method, but it is equally applicable to finite element methods. The major problem with Richardson extrapolation is that, in order for the method to yield reliable results, it must be demonstrated that the discretizations are in the asymptotic region. This is true in all cases and can be particularly troublesome in nonlinear problems or linear problems with sharp solution gradients due to the influence of singularities.

Primarily for this reason, a number of researchers have developed an alternative approach to error estimation: *a-posteriori error estimation*. This approach has its own weaknesses, but is quite valuable because it gives an elemental decomposition of the error. This decomposition provides a basis for adaptive mesh refinement, in which spatial element sizes are adjusted in accordance to the local error. This allows errors to be minimized for a given discrete problem size and for that error to be quantified (at least in a global sense). Finally, numerical results suggest that mesh refinement in an optimally adaptive sense can restore the formal convergence order of the numerical method in cases of singularities. Since adaptive meshes are typically irregular, *a-posteriori* estimators and adaptive mesh refinement are usually found within the context of the finite element method.

A-posteriori error estimation in the finite element method

A-posteriori error estimation refers generally to the use of “post-processing” to estimate numerical solution errors. By post-processing, it is meant that error estimates will be computed based on some limited processing of the solution to a given mesh, rather than by defining a new mesh spacing and computing a new solution for comparison. That is, in Richardson extrapolation, a second (or third) global solution is obtained on a different mesh spacing completely independent of other mesh solutions. In *a-posteriori* error estimation, local “fine grid” solutions are estimated *based on the existing solution* through smoothing or other techniques. While there is nothing conceptually which limits *a-posteriori* error estimators to the finite element method, practically that is where they have had the greatest impact.

The Zienkiewicz-Zhu Error Estimator [27]

For a general linear equation of the form

$$L\{u\} + p = 0 \quad (36)$$

an energy norm of the solution error can be defined as

$$\|e\| = \left(\int_{\Omega} e^T L e d\Omega \right)^{1/2} \equiv \left[\int_{\Omega} (u_{exact} - u_D)^T L (u_{exact} - u_D) d\Omega \right]^{1/2} \quad (37)$$

where the error e is defined as $e = u_{exact} - u_D$. For elasticity problems, this is equivalent to

$$\|e\| = \left[\int_{\Omega} e^T (B^T C B) e d\Omega \right]^{1/2} \quad (38)$$

where B is the “strain-displacement” matrix or more generally a differentiation operator, and C is a constitutive matrix which in elasticity problems is the stress-strain operator. Finally, using the definitions of stress σ and strain ε

$$\varepsilon = B u \quad \sigma = C \varepsilon = C B u \quad (39)$$

the L_2 error norm can be written as

$$\|e\| = \left[\int_{\Omega} (\sigma - \hat{\sigma})^T C^{-1} (\sigma - \hat{\sigma}) d\Omega \right]^{1/2} \quad (40)$$

where σ is the exact stress solution and $\hat{\sigma}$ is the discrete numerical solution. Other error norms can also be defined, such as the L_2 norms

$$\begin{aligned} \|e_u\|_{L_2} &= \left[\int_{\Omega} (u_{exact} - u_D)^T (u_{exact} - u_D) d\Omega \right]^{1/2} \\ \|e_{\sigma}\|_{L_2} &= \left[\int_{\Omega} (\sigma - \hat{\sigma})^T (\sigma - \hat{\sigma}) d\Omega \right]^{1/2} \end{aligned} \quad (41)$$

From this type of norm, root mean square (RMS) values for error can be estimated. For example, the RMS error for displacement, $|\Delta u|$, is given as

$$|\Delta u| = \left(\frac{\|e_u\|_{L_2}^2}{\int_{\Omega} d\Omega} \right)^{1/2} \quad (42)$$

Finally, a relative energy norm error is computed as

$$\eta = \frac{\|e\|}{\|u\|} \times 100\% \quad (43)$$

where

$$\|u\| = \left[\int_{\Omega} (Bu)^T C (Bu) d\Omega \right]^{1/2} = \left[\int_{\Omega} \sigma^T \varepsilon d\Omega \right]^{1/2} \quad (44)$$

is the strain energy of the displacement field. η is a global error measure for the discrete numerical solution, suitably normalized for comparison to other meshes.

At this point it is necessary to have an estimate of the exact displacement or stress to compute the error estimate. The basic idea behind the Zienkiewicz-Zhu (ZZ) estimator is to obtain an optimal improved estimate for the discrete numerical stress solution $\hat{\sigma}$ and then to assume this solution is sufficiently close to the exact solution to yield an acceptable error estimate. That is, σ^* is estimated and then it is *assumed* that

$$\|\sigma - \hat{\sigma}\| \approx \|\sigma^* - \hat{\sigma}\| \quad (45)$$

In elasticity, the stress function σ is approximately proportional to the strain function ε , which is the spatial derivative of the displacement u . Therefore, the convergence order and smoothness of the discrete numerical solution for stress is one order of magnitude less than that of displacement. For example, if linear shape functions are used, the displacement solution is second-order accurate and C^0 continuous, while the stress is first-order accurate and discontinuous (i.e. the stresses at a node point are not equal among the elements sharing that node).

Now note that the C^0 displacement function \hat{u} is interpolated from the nodal displacements \bar{u} by the shape function matrix N , i.e.

$$\hat{u} = N\bar{u} \quad (46)$$

The same shape functions can be used for the improved stress function, viz.

$$\sigma^* = N\bar{\sigma}^* \quad (47)$$

where $\bar{\sigma}^*$ is a vector of nodal stresses to be determined. Finally, the discontinuous stress field $\hat{\sigma}$ and the smoothed estimate σ^* are equated in a weighted integral sense:

$$\int_{\Omega} N^T (\sigma^* - \hat{\sigma}) d\Omega = 0 \quad (48)$$

This gives a solution for the smoothed nodal stresses:

$$\bar{\sigma}^* = \left[\int_{\Omega} N^T N d\Omega \right]^{-1} \left(\int_{\Omega} N^T \hat{\sigma} d\Omega \right) \quad (49)$$

with $\hat{\sigma} = CB\bar{u}$. Thus, from Eq. (47) through Eq. (49), the stress error is estimated as

$$\begin{aligned} e_{\sigma} &= \sigma^* - \hat{\sigma} \\ &= N\bar{\sigma}^* - \hat{\sigma} \\ &= N \left[\int_{\Omega} N^T N d\Omega \right]^{-1} \left(\int_{\Omega} N^T \hat{\sigma} d\Omega \right) - \hat{\sigma} \\ &= N \left[\int_{\Omega} N^T N d\Omega \right]^{-1} \left(\int_{\Omega} N^T CB\bar{u} d\Omega \right) - CB\bar{u} \\ &= \left\{ N \left[\int_{\Omega} N^T N d\Omega \right]^{-1} \left(\int_{\Omega} N^T CB d\Omega \right) - CB \right\} \bar{u} \end{aligned} \quad (50)$$

Note the use of the computed discrete numerical solution for the nodal displacements in the error estimate. The operator preceding the nodal displacement vector is roughly a projection matrix times the stress-strain and strain-displacement operators. The projection acts to filter out the continuous component of the stress field, leaving the purely discontinuous component as indicative of the stress error.

Using the error estimate e_{σ} , one can compute either the energy error norm given by Eq. (40) or the L_2 stress norm Eq. (41). Furthermore, note that any of these norms can be expressed as a sum of elemental errors, viz.

$$\begin{aligned} \|e\|^2 &= \sum_{i=1}^{N^E} \left\{ \int_{\Omega^{(i)}} e_{\sigma}^T C^{-1} e_{\sigma} d\Omega^{(i)} \right\} \\ \|e_{\sigma}\|_{L_2}^2 &= \sum_{i=1}^{N^E} \left\{ \int_{\Omega^{(i)}} e_{\sigma}^T e_{\sigma} d\Omega^{(i)} \right\} \end{aligned} \quad (51)$$

This is the elemental decomposition of error mentioned earlier which is used as the basis for adaptive mesh refinement.

The accuracy of these or other a-posteriori error estimators is usually defined by the effectivity index, given as

$$\psi = \frac{\|e\|_{estimated}}{\|e\|_{exact}} \quad (52)$$

An error estimator is said to be reliable if ψ is close to one as the finite element solution converges to the exact solution. This is effectively equivalent to assuming the mesh spacing is fine enough so that the solution is in the asymptotic range. Furthermore, the estimator is said to be asymptotically exact if it converges to one in the limit as the finite element solution converges to the exact solution.

The ZZ estimator just presented is an example of a *recovery* technique, where first an improved solution is obtained by some type of projection or smoothing. This is then used in place of the exact solution to compute the error estimate. This class of methods can vary depending on the method of recovery. Usually the recovery takes the form of a local problem, such that the computations involved in the post-processing/recovery phase are roughly negligible compared to the finite element solution itself. In the ZZ estimator, the operator

$$\int_{\Omega} N^T N d\Omega \quad (53)$$

which is inverted in the computation of $\bar{\sigma}^*$ (see Eq. (49)) is usually approximated as diagonal (by lumping off-diagonal terms), and so the computations involving the inverse of the operator are local and negligible. A more sophisticated recovery is the Zienkiewicz-Zhu superconvergent patch recovery (SPR), which is important in part because the theory of superconvergence allows for a proof of convergence for the error estimator. Other types of estimators exist, which Zhu [28] refers to as *residual* type error estimators. There are some provable relationships between the recovery and residual types of methods. The recovery methods have an additional attraction in that they may exploit existing post-processing techniques for improving the numerical solution; the stress smoothing reviewed above is often implemented already in linear elasticity finite element codes for reasons other than error estimation.

Although a-posteriori error indicators have some attractive features, they are not necessarily the clear choice for quantifying discretization error. First and foremost, they can provide unreliable error estimates when there is insufficient grid resolution. The estimators are at best demonstrated to be reliable only as the finite element solution converges to the exact solution. This is a consequence of treating the locally smoothed solution as the exact solution. Another major drawback is what has been referred to as *pollution error* [29]. It is, most simply, the discretization error at some location which propagates to other regions of the mesh. The local nature of a-posteriori error computations implies that pollution errors are not estimated; i.e. they are not present in the estimated error measure. While a-posteriori error indicators can be used to estimate and minimize the local error in a particular region of the mesh, there is no measure of how errors throughout the mesh might influence the global solution. The typical corrective measure taken is to adaptively refine the mesh such that errors are uniformly small throughout the domain. This can be a costly exercise, however. The presence of pollution error implies that, for many

problems, it is difficult to develop a reliable a-posteriori error estimate for a specific output quantity of the model as was done with Richardson extrapolation. For example, the local error in displacement at the end of a cantilevered beam may be nearly zero, but if the mesh at the base of the beam is poorly resolved, the resulting error in the slope can be geometrically amplified over the length of the beam. This results in a propagated error, or pollution error, which is not included in the local error captured by the a-posteriori estimate. Finally, other drawbacks of a-posteriori error estimators are uncertainty about their performance on nonlinear and transient dynamics simulations.

Researchers are currently investigating the direct application of a-posteriori error estimates to quantitative error assessment, but in the meantime these estimators can be used in a number of valuable ways. First, a-posteriori error estimates can be used as a basis for optimal adaptive mesh refinement, so as to minimize discretization error for a given discrete problem size. It has been shown that, when optimal meshes are then uniformly refined, it is possible to retain the formal convergence rate of the numerical method even in the presence of singularities. Furthermore, the total error norm can be used in determining plausibility measures for competing models in a model uncertainty assessment. Finally, the error norms can be used to estimate or verify formal convergence rates of methods. Thus, one might use multiple meshes, each with computed a-posteriori error norms, to determine convergence rates and then apply a Richardson extrapolation-like method to compute an output quantity-specific error estimate.

Project Activities and Contributions on Estimation of Numerical Solution Error

The present LDRD project developed a methodology for estimation of numerical solution error within the context of non-deterministic analysis. The method integrates response surface methodologies with Richardson extrapolation to provide estimates of numerical solution errors or biases on statistics of the solution space. The first version of this approach uses polynomial error models and is detailed in Appendix B. A second version of the approach considers the use of rational function models for discretization error in order to expand the sphere of convergence of the error estimate. This work is detailed in Appendix E. In both investigations, it was found that relatively small bias error in computational simulation, resulting from either discretization or modeling assumptions, can have significant effects on system reliability estimates. The relative effects are most important for systems with low inherent physical variability and small failure probabilities. The integrated methodology is an effective approach to integrating different levels of computational fidelity within the context of uncertainty quantification.

5. Conclusions

This effort has investigated a variety of computational approaches to quantifying the effects of modeling and discretization uncertainty in computational simulation and uncertainty analysis. The primary methodology has been to separate ordered error sources from uncertainties and non-ordered errors and quantify their effects independently. A key computational strategy has been the use of response surfaces (or metamodels) for developing algorithms to treat both modeling and discretization effects.

Ordered error sources, such as discretization error, typically produce a systematic bias in the computational solution (relative to the solution of the underlying mathematical model) which can be estimated via extrapolation with an ordered error model. The classical technique of Richardson extrapolation was extended to non-deterministic analysis via response surface methodology, using both polynomial and rational function error models. The integrated approach allows for effective estimation of bias errors on probabilistic measures such as reliability using design of experiments over a continuous parameter space and a converging sequence of computational meshes.

Uncertainty in the structural assumptions of mathematical models, together with non-ordered modeling errors, have been addressed through a Bayesian model averaging approach. This methodology requires the definition of a space of model forms, with associated prior model probabilities which reflect the relative plausibility of each model. The Bayesian framework is easily generalized to a discrete space of model alternatives, and propagation reduces to a weighted averaging of predictions from each model, with uncertainties arising from both continuous model parameters and model-to-model variation. Work was performed to incorporate discrete model form parameters into metamodels, as well as to integrate multiple models via master response surfaces.

The following appendices, arranged in chronological order, include detailed reports on the various tasks performed within the project. The work reported herein has raised new questions and stimulated new ideas which should be pursued in subsequent uncertainty research. First, the error estimation methods using extrapolation with rational functions should be developed more completely and incorporated into an uncertainty quantification code. These methods should also be extended to use a-posteriori error measures as much as possible. Second, there should be an investigation of the use of convex sets and evidence theory for modeling structural uncertainties in complex problems. Third, the Bayesian model averaging approach should be applied to problems in constitutive modeling for structural, thermal, and fluid mechanics. Fourth, the same model averaging approach has potential for improving the robustness of linear and nonlinear model reduction methods, as

well as response surface methods, and its application to these problems should be investigated.

Finally, modeling and discretization uncertainty quantification, together with code verification and validation methodologies, needs to be examined within the context of system design qualification and certification activities. This is important because of the potential costs involved in the development of model uncertainty quantification methods and codes, as well as their use in qualification activities. Such an examination should study the use of model uncertainty techniques in validation, and determine the proper mix of validation activities and model uncertainty quantification activities to support weapon system certification.

6. References

1. Draper, D. (1995), "Assessment and Propagation of Model Uncertainty (with discussion)," *Journal of the Royal Statistical Society Series B*, Vol. 57, No. 1, pp. 45-97.
2. McKay, M. D. (1995), "Evaluating Prediction Uncertainty," Los Alamos National Laboratory, LA-12915-MS.
3. Oberkampf, W. L., Diegert, K. V., Alvin, K. F., and Rutherford, B. M. (1998), "Uncertainty and Error in Computational Simulation," *Proceedings of the 1998 AIAA Fluids and Thermodynamics Conference*, Albuquerque, NM, June, 1998.
4. Krippendorff, K. (1986), *A Dictionary of Cybernetics*, Unpublished Report, 1986; accessed through *Web Dictionary of Cybernetics and Systems* (<http://pespmc1.vub.ac.be/ASC/indexASC.html>) of the PRINCIPIA CYBERNETICA WEB (<http://pespmc1.vub.ac.be/>).
5. Madigan, D. and Raftery, A. E. (1994), "Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window," *Journal of the American Statistical Association*, Vol. 89, No. 428, pp. 1535-1546.
6. Kass, R. E. and Raftery, A. E. (1995), "Bayes Factors," *Journal of the American Statistical Association*, Vol. 90, No. 430, pp. 773-795.
7. Siu, N. and Apostolakis, G. (1985), "On the Quantification of Modeling Uncertainties," *Transactions of the 8th International Conference on Structural Mechanics in Reactor Technology*, Vol. M2 1/5, pp. 375-378.
8. Ahn, K.-I. and Jin, Y.-H. (1996), "A Formal Approach for Quantitative Treatment of Modeling Uncertainties in Safety Analysis," *Nuclear Technology*, Vol. 116, pp. 146-159.
9. Laskey, K. B. (1996), "Model Uncertainty: Theory and Practical Implications," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, Vol. 26, No. 3, pp. 340-348.
10. Raftery, A. E., Madigan, D., and Volinsky, C. T. (1996), "Accounting for Model Uncertainty in Survival Analysis Improves Predictive Performance," *Bayesian Statistics V*, to appear.
11. Madigan, D., Raftery, A. E., Volinsky, C. T., and Hoeting, J. A. (1996), "Bayesian Model Averaging: A Review" *Proceedings of the 1996 AAAI Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, American Association for Artificial Intelligence, pp. 77-83.
12. Jazwinski, A. H. (1970), *Stochastic Processes and Filtering Theory*, Academic Press, San Diego, CA, pp. 195-197.

13. Ben-Haim, Y. (1997), "Robust Reliability of Structures," *Advances in Applied Mechanics*, Vol. 33, pp. 1-41.
14. Ben-Haim, Y., Cogan, S., Sanseigne, L. (1998), "Usability of Mathematical Models in Mechanical Decision Processes," *Mechanical Systems and Signal Processing*, Vol. 12, No. 1, pp. 121-134.
15. Ben-Haim, Y. (1996), *Robust Reliability in the Mechanical Sciences*, Springer-Verlag, Berlin, 1996.
16. Guan, J., and Bell, D. A. (1991), *Evidence Theory and Its Applications, Volume 1*, Elsevier Science, North-Holland - Amsterdam.
17. Dempster, A. P. (1967), "Upper and Lower Probabilities Induced by a Multivalued Mapping," *Annals of Mathematical Statistics*, Vol. 38, pp. 325-339.
18. Shafer, G. (1976), *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ.
19. Walley, P. (1991), *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, London.
20. Dubois, D., and Prade, H. (1988), *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York.
21. Klir, G. J., and Folger, T. (1988), *Fuzzy Sets, Uncertainty, and Information*, Prentice Hill, Englewood Cliffs, NJ.
22. Fletcher, C. A. (1988), *Computational Techniques for Fluid Dynamics, Vol. I: Fundamental and General Techniques*, Springer Ser. Comput. Phys. (Springer, New York, Berlin, Heidelberg)
23. Hughes, T. J. R. (1987), *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. (Prentice Hall, Englewood Cliffs, N. J.)
24. Richardson, L. F. (1910), "The approximate arithmetical solution by finite differences of physical problems involving differential equations with an application to the stresses in a masonry dam," *Trans. R. Soc., Lond., Ser. A*, Vol. 210.
25. Roache, P. J. (1994), "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, Vol. 116, pp. 405-413.
26. Oberkampf, W. L., and Blottner, F. G. (1998), "Issues in Computational Fluid Dynamics Code Verification and Validation," *AIAA Journal*, Vol. 36, No. 5, pp. 687-695.
27. Zienkiewicz, O. C., and Taylor, R. L. (1989), *The Finite Element Method, 4th Ed., Vol. 1: Basic Formulation and Linear Problems*. (McGraw-Hill, London)
28. Zhu, J. Z. (1997), "A-posteriori error estimation - the relationship between different procedures," *Computer Methods in Applied Mechanics and Engineering*, Vol. 150, pp. 411-422.
29. Babuska, I., Strouboulis, T., Mathur, A., and Upadhyay, C. S. (1994), "Pollution Error in the h-Version of the Finite Element Method and the Local Quality of A-Posteriori Error Estimators", *Finite Elements in Analysis and Design*, Vol. 17, pp. 273-321.

Intentionally Left Blank

APPENDIX A:

Uncertainty Quantification in Computational Structural Dynamics: A New Paradigm for Model Validation

**presented at the 1998 International Modal Analysis
Conference, Santa Barbara, CA, February 1998.**

Intentionally Left Blank

UNCERTAINTY QUANTIFICATION IN COMPUTATIONAL STRUCTURAL DYNAMICS: A NEW PARADIGM FOR MODEL VALIDATION

K. F. Alvin*, W. L. Oberkampf†,
K. V. Diegert‡ and B. M. Rutherford**

P.O. Box 5800, Mail Stop 0439
Sandia National Laboratories
Albuquerque, NM 87185

ABSTRACT. We present an overview of new research efforts underway at Sandia National Laboratories to understand the sources of uncertainty and error in computational structural dynamics and other physics simulations, and to quantify their effects on predictive accuracy. In order to establish confidence in computational simulations as these simulations move further from the established experimental database, a new approach to modeling and simulation validation is needed. In particular, when simulations are used to qualify the safety and reliability of systems, we believe that validation should be based upon a comprehensive quantification of uncertainties and errors from all phases of the modeling and simulation process. Uncertainty and error quantification is a two-step process, the first step being the identification of all uncertainty and error sources in each phase of modeling and simulation. The second step is the assessment and propagation of the most significant uncertainties and errors through the phases of the modeling and simulation process to the predicted response quantities. This paper outlines the phases of modeling and simulation, the distinction between uncertainty and error, and a categorization of uncertainty and error sources in each phase of modeling and simulation. We also address the question of how uncertainties in the form or structure of the model might be assessed using multiple models. Examples from linear structural dynamics are given to illustrate these concepts.

1. Introduction

Model validation in structural dynamics is a well established discipline which bridges analysis and experiment. This field has traditionally focused on reconciliation of test-to-analysis results with the

goal of improving the predictive accuracy of computational model-based analysis. Model validation and "virtual testing" have played key roles in the engineering development of advanced aerospace and flight systems. This is because some qualification tests simply cannot be performed (or it is absurd to do so); i.e. we don't launch a satellite to see whether it will survive launch. Other tests can be performed but are not because a completely test-based approach would be prohibitively expensive. Therefore, to some extent, nearly all advanced engineering systems rely on computational simulations to not only improve designs but also to qualify, i. e. ensure the satisfactory performance of, the system hardware and design. For this reason, validation of the computational simulations is key to ensuring the performance, safety and reliability of these systems.

In structural dynamics, the criteria typically set for validation of analysis by experiment have been geared towards deterministic analysis. These validation criteria, such as the correlation of numerical model predictions to the frequencies and mode shapes estimated from modal testing, are both qualitative and subjective. The error threshold levels are intrinsically tied to what types of qualification testing will be performed, the degree of extrapolation in the critical analyses, and the notion of a factor of safety on all safety margin stress calculations. This is not to discount the value of these thresholds: they have evolved over time and express a considerable degree of expert knowledge and experience. For any particular modeling and simulation analysis, however, this type of validation does not enhance our understanding of the predictive accuracy of the analysis as we use the model to extrapolate from our measured database.

Some researchers in computational structural dynamics have used statistical techniques in parameter estimation to begin to address uncertainties [1], and have also considered the use of both uncertain parameters and historical test-analysis correlation measures to determine predictive accuracy intervals for linear and nonlinear structural dynamics modeling and simulation [2]. The primary motivations for

*Structural Dynamics Dept., Org. 9234

†Aerosciences Dept., Org. 9115

‡Statistics and Human Factors Dept., Org. 12323

**Statistics and Human Factors Dept., Org. 12323

addressing uncertainties are a.) the stochastic or uncertain nature of key elements in engineering design and reliability problems, and b.) the need to establish the credibility of computational simulations.

A convergence of trends are pushing the current "way of doing things" and lead us to reconsider how computational simulation is to be used in system qualification in the future, and how those analyses are to be validated. "Smaller, cheaper, faster" implies less testing on systems which are less easily analyzed than in the past. Computational technology provides tools for improving analysis predictions by reducing discretization errors and reducing the degree of analysis simplification by enabling more detail and better handling of coupled multiphysics. Unfortunately, systematic improvements in analysis predictions have not been seen as computational technologies have improved. We believe a key reason is that total errors are being driven more by modeling simplification than by solution errors. It is also possible that significant discretization and solution errors are still present, but that they are not being detected because of a lack of attention to mesh convergence issues. Finally, it may be that increased grid resolution has simply replaced modeling and solution errors at the macro scale with lack of knowledge (e.g. unknown joint physics) at the micro level.

In order to establish confidence in computational simulations as these simulations move further from the established experimental database, a new approach to modeling and simulation validation is needed. In particular, when simulations are used to qualify the safety and reliability of systems, we believe that validation should be based upon a comprehensive quantification of uncertainties and errors from all phases of the modeling and simulation process. This quantification would consider uncertainties arising from both continuous parametric uncertainties, such as variability in a geometric dimension or material property, and discrete modeling uncertainties, such as uncertainty about the physics equations governing joint compliance. In addition, quantification of errors would include numerical solution errors, such as those arising from grid resolution, as well as modeling errors, such as simplified modeling assumptions.

We believe there are two stages to uncertainty quantification: first, identification of all error and uncertainty sources; and second, assessment and propagation of uncertainty and error effects through the simulation to output quantities. This paper will present a general classification of all sources of uncertainty and error necessary to determine global estimates of uncertainty for predicted quantities of interest. This framework will then be applied to a structural dynamics problem: the prediction of peak acceleration on a circular plate due to random force excitation. Also, the sensitivity of the predicted quantity to some significant and nontraditional uncertainty and error sources, such as the partial differential equations and spatial discretization, will be examined via multiple numerical simulations.

2. Sources of Uncertainty and Error in Computational Mechanics Simulations

In this section we discuss the sources of uncertainty and error in computational simulations and propose an overall framework which categorizes these sources and their interactions. This framework is presented in greater detail in [3]. We begin by developing a new structure of the general phases of modeling and simulation. This new view is built upon combining modeling and simulation phases

recognized in the disciplines of operations research and the numerical solution of partial differential equations. Within this structure, we believe that a clear distinction should be made between uncertainty and error and we propose comprehensive definitions for these terms. Specific classes of uncertainty and error sources are then defined that can occur in each phase of modeling and simulation.

2.1 Phases of Modeling and Simulation

We will use the definition of *model* given by Neelamkavil [4]: "A model is a simplified representation of a system (or process or theory) intended to enhance our ability to understand, predict, and possibly control the behavior of the system." By *modeling* we mean the construction or improvement of a model. We also use Neelamkavil's definition of *simulation*: "A simulation is the process of imitating (appearance, effect) important aspects of the behavior of the system." In other words, simulation is the exercise of the model. Here we are specifically interested in the exercise of computer models, i.e., computer codes based on mathematical models.

After reviewing existing literature in both the operations research and computational mechanics fields, we have developed a representation of the phases of modeling and simulation appropriate to systems or processes analyzed by the numerical solution of partial differential equations (PDE's) as shown in Figure 1. These phases are preceded by an initial phase which includes the definition of the physical system and specification of the requirements or objectives of the modeling and simulation. Following this initial phase are six distinct phases which are described as follows.

Conceptual Modeling The conceptual modeling phase determines what physical events, or sequence of events, will be considered and what types of coupling of different physical processes will be considered. During this phase, no mathematical equations are written, but the fundamental assumptions of the events and physics are made. Only conceptual issues are considered, with an emphasis on determining all possible factors that could affect the requirements set for the modeling and simulation. It is important in this phase that all possible physics-couplings are listed that may influence the results even if they may not be considered later on in the analysis. This is critical because if events or couplings are not considered in this phase, they cannot be resurrected later in the process. This is similar to the fault-tree structure in probabilistic risk assessment of high consequence systems, such as in nuclear reactor safety analyses. Even if a certain sequence of events is considered extremely remote, it should still be considered as a possible event sequence in the fault-tree. Whether the event sequence will eventually be analyzed is not a factor in including it in the conceptual modeling phase.

Mathematical Modeling During the mathematical modeling phase, the precise mathematical, i.e., analytical, statement of the problem, or series of event-tree-driven problems, to be solved is developed. Any complex mathematical model of the problem, or physical system, is likely to be composed of many submodels. The complexity of the models depends on the physical complexity of each phenomenon being considered, the number of physical phenomena considered, and the level of coupling of different types of physics. The mathematical model formulated in this phase is considered to be the complete specification of all of the PDE's for all components of the system. Along with the PDE statement of the mathematical model,

all of the appropriate initial and boundary values, and the required auxiliary models must be specified for the physics considered.

Discretization of the Model The next phase is the conversion of the PDE form of the mathematical model into a discrete numerical model. This phase takes into account the conversion of the mathematics from a calculus problem to an arithmetic problem. In the discretization phase, all of the spatial and temporal differencing methods, discretization of the boundary conditions, discretization of the geometric boundaries, and grid generation methods are specified in analytical form. In other words, algorithms and methods are prescribed in mathematically discrete form, but the spatial and temporal step sizes are not specified. This step focuses on the conversion from continuum mechanics to discrete mathematics, not on numerical solution issues. We believe that the continuum model and the discrete model should be separately represented in the phases of modeling and simulation. This phase deals with questions such as consistency of the discrete equations with the PDE's, and conversion of mathematical singularities in the continuum into discrete representations.

Programming of the Discrete Model The next phase, which is common to all computer modeling and simulation, is the computer programming phase. This phase converts the algorithms and solution procedures defined in the previous phase into a computer program. This phase has probably achieved the highest level of maturity because of many years of programming development and software quality assurance efforts. These efforts have made a significant impact in areas such as commercial graphics, mathematics, and accounting software, telephone circuit switching software, and flight control systems. Little impact, however, has been made in corporate and university developed software developed for research applications in computational mechanics.

Numerical Solution of the Programmed Discrete Model The next phase, individual numerical solutions are obtained. This phase is the most specific of all phases of modeling and simulation. At the conclusion of this phase there are no quantities left arithmetically undefined or continuous. For example, grid spacing is specified, parameters such as material constants and damping coefficients are specified, and time and space exist only at points. If uncertainty in some inputs or physical parameters of the numerical model are passed through to the numerical solution phase, as is commonly the case in nondeterministic analysis, then multiple computational solutions would be required. Consider, for example, a shock response analysis where the material elastic modulus is specified by some probability distribution. Then thousands of Monte Carlo solutions may be required to address the problem definition.

Interpretation of Results The final phase concerns the interpretation of computational results. This phase involves determining the methods for presentation of computed results into a usable form. This phase can also be described as the construction of continuous functions based on the discrete solutions obtained in the previous phase. Here the continuum mathematics formulated in the mathematical modeling phase is approximately reconstructed. This phase is specifically called out because of the sophistication of the software that is being developed to comprehend modern computational simulations. This area includes graphical visualization of results, animation, and perhaps use of sound or virtual reality. Some may argue that this phase is simply "post-processing" of the computational da-

ta. This description does not do justice, however, to the rapidly growing importance of this area and its capability for introducing unique forms of errors.

2.2 Sources of Uncertainty and Error

We now discuss the sources of uncertainties and errors that are associated with each phase of modeling and simulation, as illustrated in Figure 2. Essentially all of the individual sources of uncertainty and error described below have been pointed out by researchers in the past. Some, like computer round-off, are very well understood, even to the point that most computational analysts do not make note of it. Others are poorly understood or characterized, and it may be unclear whether they should be treated as an uncertainty or an error. For this we must first develop comprehensive definitions for uncertainty and error that are appropriate for modeling and simulation.

Definitions of Uncertainty and Error The most developed definition or understanding of uncertainty is in regard to experimental measurements. Although this is helpful, we require definitions that apply to the much broader topic of modeling and simulation. We define *uncertainty* as a *potential* deficiency in any phase or activity of the modeling process that is due to *lack of knowledge*. The first feature which this definition stresses is "potential", meaning that the deficiency may or may not occur. In other words, there may be no deficiency, say in the prediction of some event, even though there is a lack of knowledge. Whether the deficiency occurs is most commonly represented by some type probability distribution of occurrence. The second key feature of uncertainty is that its fundamental cause is incomplete information. Since the cause of uncertainty is lack of knowledge, increasing the knowledge base can reduce uncertainty.

We define *error* as a *recognizable* deficiency in any phase or activity of modeling and simulation that is *not* due to lack of knowledge. This definition stresses the feature that the deficiency is identifiable or knowable upon examination, that is, the deficiency is not determined by lack of knowledge. By this we mean that there is an agreed-upon approach which is considered to be more accurate. If divergence from the correct or more accurate approach is pointed out, the divergence is either corrected or allowed to remain. This implies a segregation of error types: error can be either *acknowledged* or *unacknowledged*. Examples of acknowledged errors are: finite precision arithmetic in a computer; physical approximations made to simplify the modeling of a physical process; a specified level of iterative convergence of a numerical scheme; conversion of the governing PDE's into discrete equations. When the analyst introduces these acknowledged errors in the modeling or simulation process, there is typically some idea of the magnitude of the error introduced. Unacknowledged errors are blunders, or mistakes. That is, the analyst intended to do one thing in the modeling and simulation, but, for example, due to human error, did another. There are no straightforward means to estimate or bound the contribution of unacknowledged errors, although steps, such as independent checks and reviews, can reduce their frequency of occurrence.

We will now detail the sources of errors and uncertainties in four of the phases discussed previously. A discussion of error sources in the programming and results interpretation can be found in [3].

Conceptual Modeling Uncertainties The dominant deficiency in the

conceptual modeling phase is uncertainty, as opposed to error. Conceptual modeling uncertainties arise in the formulation of the analysis of the event, and in the lack of knowledge of the event. Figure 2 shows the two types of uncertainties associated with conceptual modeling: scenario abstraction and lack of system knowledge. By scenario abstraction we mean the determination of all possible physical events, or event sequences, that may affect the goals of the analysis. For relatively simple systems, such as low level vibration of a thin circular plate in a vacuum, scenario abstraction can be straight forward. For complex engineering systems exposed to a variety of interacting factors, scenario abstraction is a mammoth undertaking.

The second class of uncertainty listed, lack of system knowledge, refers to uncertainties that are primarily due to limited information about the system. This class clearly affects and interacts with scenario abstraction, but here we stress lack of knowledge for a specific scenario, rather than the possible existence of the scenario. Two important examples for this class of uncertainty should be mentioned. First is the lack of knowledge of the initial state of key elements of the system. For complex engineered systems, knowledge of the factors, such as the following, becomes important: was the system correctly manufactured and assembled, how well was the system maintained, and what level of uncertainty exists for the properties of the components which are important in the analysis (such as dimensions, densities, elastic moduli, etc.). The second example is lack of knowledge of future conditions affecting the system. Examples of these are environmental conditions and human interaction with the system during the event. These are examples where it is not possible to significantly reduce lack of knowledge, and reduce the uncertainty, by improved sampling of past events.

Mathematical Modeling Uncertainties and Errors Mathematical modeling contains both uncertainties and errors. Uncertainties and errors that occur in this phase arise from three mathematical sources (Figure 2): the continuum equations for conservation equations of mass, momentum, and energy; all of the auxiliary equations which supplement the conservation equations; and all of the initial and boundary conditions required to solve the PDE's. The primary uncertainties that occur in mathematical modeling are two fold. First is inadequate knowledge of parameters in known physics. Parameter uncertainty is by far the most commonly analyzed in uncertainty analyses. The second type of uncertainty is that due to limited, or inadequate, knowledge of the physics involved. For example, not knowing the PDE's which govern friction in a mechanical joint. Errors in the mathematical modeling phase can be equally significant. The primary errors are those due to mathematically representing the physics in a more simplified form than is known to be appropriate for the results required from the modeling and simulation. The mathematical modeling uncertainties and errors together are sometimes referred to "model form errors" or model structural errors".

A primary example of uncertainty that occurs in the conservation equations for structural dynamics is the localized nonlinear physics of friction, contact, and impact in bolted joints. Auxiliary physical equations in the mathematical model are equations such as the material constitutive models and failure models. Examples of uncertainties in initial and boundary conditions are: inaccurately known initial velocity of a body, and imprecisely known geometry of materials because of manufacturing and assembly variances. Errors in mathematical modeling can also exist. Some examples of acknowl-

edged errors are: assumption that a plate can be modeled using thin shell theory when three dimensional effects are important, assumption of a constant beam cross-section when the section is actually not constant, and assumption of material and geometric linearity when stresses and displacements are not small. All of these examples are of the character that physical modeling approximations were made to simplify the mathematical model and the subsequent solution.

Discretization Errors The discretization phase converts the continuum model of the physics into a discrete mathematics problem. Since this is fundamentally a mathematics approximations topic, errors and not uncertainties are the dominant issue in this phase. Some may question why this conversion process should be separated from the solution process, where the characteristic mesh size and time integration step sizes are set. We argue that this conversion process is the root cause of more difficulties in the numerical solution of PDE's than is generally realized. This is particularly true in cases of nonlinear phenomena such as fracture dynamics and frictional contact in mechanical joints. It can also be evident in linear structural mechanics and dynamics, where the presence of singularities in the continuum model creates solution error which does to disappear as the grid size approaches zero. It is becoming increasing clear that the mathematical features of strongly nonlinear and chaotic systems can be fundamentally different between the continuous and discrete form, regardless of grid size [5,6].

As shown in Figure 2, we identify three sources of discretization error; discretization of the conservation laws, the boundary conditions, and the initial conditions. The types of errors we are pointing out here are typically very difficult to isolate. In finite differencing, one method of identifying these type errors is to analytically prove whether the method is consistent; that is, do the finite difference equations approach the continuum equations as the step size approaches zero. Related issues dealt with in this phase: are the conservation laws satisfied for finite grid sizes, does the numerical damping approach zero as the mesh size approaches zero, and do aliasing errors exist for zero mesh size. Discretization of PDE's are also involved in the conversion of von Neumann and Robin's, i. e., derivative, boundary conditions to difference conditions. We include the conversion of continuum initial conditions to discrete initial conditions, not because there are derivatives involved, but because spatial singularities may be part of the initial conditions. Some may argue that these discontinuities and boundary singularities do not actually occur in nature, so the issue of accuracy of representation of these is superfluous. This misses the point, however. If these features exist in the mathematical model of the physics, the issue is whether the discrete model represents them accurately; not whether they exist in nature. This is an issue of verification (solving the problem right) rather than validation (solving the right problem).

Numerical Solution Errors Numerical solution errors have been investigated longer and in more depth, than any of the errors associated with the numerical solution of PDE's. Indeed, they have been investigated since the beginning of numerical solutions; Richardson in 1910 [7]. These deficiencies in the solution of the discrete equations are properly called errors because they are approximations to the solutions of the original PDE's. As shown in Figure 2, we categorize these errors into four categories: spatial grid convergence, time step convergence, iterative convergence, and computer round-off. Of these, perhaps the only one that needs explanation is iterative

convergence. By this we mean the finite accuracy to which algebraic discrete equations are solved. In linear structural dynamics, iterative convergence errors can occur when iterative methods, e. g. conjugate gradients, are used to solve the large matrix equation within a time step. Iterative errors can also occur in most algorithms, such as Lanczos, used to iteratively solve the generalized symmetric eigenvalue problem. In fact, since large matrix algebra problems are typically posed at each iteration of an iterative eigenvalue method, we can encounter both inner and outer loop iterative convergence errors if iterative methods are used to solve the matrix equation. The use of iterative techniques becomes even more necessary as nonlinear phenomena are introduced into the physics; in that case many levels of iterative convergence errors may be encountered.

Although we categorize four sources of solution error, it should be noted that they are of two types. The first is due to the finite discretized solution of the PDE's; spatial grid convergence and time step size convergence are of this type. The second type is due to the approximate solution of the discrete equations, that is, what errors are made in the solution of the resulting discrete equations. Iterative convergence and round-off error are of this type and they account for the difference between the exact solution of the discrete equations and the computer solution obtained.

3. Assessment and Propagation of Model Uncertainty

In the previous section, we presented an overall framework for the phases of modeling and simulation, and within that framework categorized sources of uncertainties and errors which are important in modeling and simulation. Understanding the sources of uncertainty or error in a particular analysis problem, however, is only the first step in quantifying the total combined uncertainty and error in the simulation. The second step is to propagate those uncertainties and errors from their origin and through the subsequent phases of the simulation to determine their impact on the output of the simulation. While a great deal of attention has been given to ways of estimating probability distributions of simulation outputs given the distributions of continuous, nondeterministic inputs, relatively little attention has been paid to the model itself. By the form of the model we mean those attributes, such as simplifying assumptions, choice of PDE's, computational mesh topology, and other non-parameterized features which determine the form of the PDE's and ultimately the order of the algebraic problem. We believe that, in order to quantify the major sources of uncertainty and error in modeling and simulation, we must develop procedures to quantify the effects of these modeling errors and uncertainties.

One approach to the assessment of mathematical model uncertainty, as distinct from parameter uncertainty, is suggested by Draper [8]. This approach has been termed by Draper and others [9] as Bayesian Model Averaging, because the approach uses the Bayesian concept of prior probability densities which are then revised to incorporate new data. In statistical parameter estimation, Bayesian methods are similar to maximum likelihood estimates in that they consider the relative uncertainty or reliability of the relevant data. They are different, however, in that they also consider prior beliefs or qualitative information on the parameter being estimated, which serve to regularize the estimation.

While probability concepts are appropriate for addressing uncertain-

ties, it is not yet clear how useful they may be in assessing the effects of modeling errors, such as acknowledged simplifying assumptions and numerical errors caused by finite spatial discretization. As discussed in the preceding section, uncertainties and errors are not equivalent. They are, however, similar in that we are interested in the sensitivity of the simulation output to both. In the present discussion and subsequent example, we will deal with multiple competing models which differ in their simplifying assumptions, element topology, and model size. While these model attributes are a combination of errors and uncertainties, we intend to apply the Bayesian model averaging approach in order to both illustrate the methodology and to begin assessing the effects of these errors and uncertainties on the simulation output.

In the problem of assessing mathematical modeling uncertainty, we must address aspects of the model which are not parameterized, at least not in continuous terms. Draper proposes that, analogous to a continuous expansion of models in a space measured by a vector of continuous parameters, we instead consider a discrete expansion of possible model structures. It is then required that the probabilities of all the models considered together sum to unity. We know that we necessarily cannot consider the potential infinity of model structures which could potentially be applied, correctly or incorrectly, to a given problem. Therefore, we seek models which are supported by expert knowledge, are supported by available diagnostic experimentation, and differ markedly in their predictions. This is because, if a model has no prior support of expert opinion, it cannot gain any support from new data. That is, the model's probability will always be zero. Similarly, if a particular model has some prior support, but no support from new data, it will eventually lose influence in competition with other models that correlate better with the data. Finally, if a number of models with nontrivial probabilities all predict the problem in the same way, they are not really contributing to the quantified uncertainty, although their relative structural diversity would enhance confidence in the uncertainty quantification itself by their representation of a large model space.

After selection of the models to be considered in such an analysis, we must then assign probabilities based on our relative beliefs in the different competing models. These probabilities might be purely subjective and qualitative, or they may begin with assumed subjective probabilities which are then updated quantitatively using relevant existing data. Then we determine prediction statistics for each model and use this information to determine prediction statistics for the space of models. That is, if y is a response quantity of interest given system input x , with a finite set $S = \{S_1, \dots, S_m\}$ of structural model alternatives, we have

$$\begin{aligned}
 p(y|x, S) &= \sum_{i=1}^m \int p(y|x, S_i, \theta_i) p(S_i, \theta_i | x) d\theta_i \\
 &= \sum_{i=1}^m p(S_i | x) p(y|x, S_i)
 \end{aligned} \tag{1}$$

where θ_i are parameters of model i . This result consists of two key quantities: $p(y|x, S_i)$ is the probability density function for the response given the known data from a particular single model S_i ; $p(S_i | x)$ is the probability of that model given the data. The result obtained is that the expected value for a response quantity of interest is weighted average of model means, and the variance is a weighted

average of individual model variances plus the variance in the means of the models across the entire model space.

The use of multiple models to determine measures of model uncertainty is strongly dependent on the relative probabilities for the competing models $p(S_i | x)$. These probabilities can be based on purely subjective knowledge - assigning weights based on qualitative judgements. That is,

$$p(S_i | x) = p(S_i) \quad i = 1, \dots, m \quad (2)$$

where $p(S_i)$ is based on qualitative judgements, with the constraint

$$\sum_{i=1}^m p(S_i) = 1 \quad (3)$$

Another approach is to use relevant historical database, i.e. compute probabilities for competing models/assumptions using known data as well as qualitative assessments. In this case, the desired probabilities are given by

$$p(S_i | x) = \frac{p(x|S_i)p(S_i)}{\sum_{j=1}^m p(x|S_j)p(S_j)} \quad (4)$$

Here again we must have some prior model probabilities $p(S_i)$ which are assumed independent of the known data x and are subject to the constraint given in Eqn. 3. Then these probabilities are updated via Eqn. 4, which accounts for how well each of the models correlate with the data.

In conclusion, we note that the predictive distribution for a particular model i is given by

$$p(y|x, S_i) = \int p(y|x, S_i, \theta_i)p(\theta_i | S_i, x)d\theta_i \quad (5)$$

This is what most of nondeterministic analysis methods such as Monte Carlo and Latin Hypercube sampling, fast probability integration, stochastic finite elements, and reliability techniques attempt to address. That is, they determine prediction statistics for model output given uncertainty in the parameters of the model θ_i , conditional on the data x and a particular model structural choice S_i . The present Bayesian model averaging approach extends this to a space of models S using the structural probability weights $p(S_i | x)$.

4. Examples

In order to illustrate the ideas presented herein, we examine two problems. The objective of the first problem is to identify the sources of uncertainty and error in the simulation of a simple structural dynamics system. In the second problem, our objective is to illustrate the concept of modeling uncertainty assessment using a structural dynamics simulation with multiple models.

4.1 Uncertainty and Error Sources: Response of a Circular Plate

Our task in this problem is to predict the peak acceleration response of a free-free thin aluminum (6061-T6 alloy) circular plate at $r=R/2$ to applied vertical force at the center of the plate. Our objective, as stated before, is to identify all sources of errors and uncertainties

likely to be encountered in the modeling and simulation of this system to meet the requirement of the analysis. The following list, although not complete, contains what we believe to be the primary sources which will influence prediction of the peak response:

Conceptual modeling uncertainties

- condition of the plate
- unknown variability in dimensions and material properties
- uncertainty in random force statistics
- environmental and system boundary conditions

Mathematical modeling uncertainties and errors

- Use of linear methods
- choice of PDE (2D or 3D)
- validity of shell theory for 2D elements
- Extension/shear/bending coupling effects
- nonsymmetry in problem if using axisymmetric elements
- mass modeling (consistent versus lumped)
- micromechanics of applied force
- variability in dimensions and material properties
- Errors in statistical methods for estimating peak response

Discretization Errors

- element selection
- force interpolation
- damping discretization
- time discretization (modal superposition vs. time integration)
- substructure reduction effects

Programming and documentation errors

- Errors in closed-form solutions
- binary format conversions
- equivalencing errors
- uncertain code defaults
- code/memory errors
- inaccurate documentation

Numerical solution errors

- finite spatial resolution
- finite temporal discretization
- iterative tolerance for eigensolver
- stiff/compliant elements increasing round-off errors

4.2 Assessment of Model Uncertainty: Frequencies of a Bracket

The second problem is to predict the free-free frequencies of vibration of a component mounting bracket and to determine confidence bounds on the predicted frequencies due to considering multiple models. In particular, we will consider variations in the geometric features included in the model, the topology and fidelity of the spatial mesh, and the types of elements used. In order to summarize the results of the different models, we will use the Bayesian model averaging approach detailed in Section 3.

The bracket, shown in Figures 3, 4 and 5, is composed of a phenolic material, whose properties are given as:

$$E \sim N(3.425 \times 10^6 \text{ psi}, 1.5667 \times 10^9 \text{ psi}^2) \quad COV = 1.116\% \\ \rho = 0.0689 \text{ (lb/in}^3\text{)}$$

That is, the elastic modulus is defined by a normal distribution with a mean of 3.425×10^6 psi and a variance of 1.5667×10^9 psi². The coefficient of variation (COV), which is the standard deviation as a percentage of the mean, is 1.116%. The density is as given without any associated variability.

The three models considered are as follows:

- Model I: This model was spatially discretized using surface elements with a paving algorithm. The surface elements were then extruded into 8-noded hexahedral solid elements. The surfaces contain all of the thru holes, but fillets were not modeled. The resulting model has 31554 degrees of freedom (DOF).
- Model II: This model was meshed from a ProEngineer solid geometry definition, which contained all holes and major fillets. The solid was meshed using 10-node tetrahedral elements. The resulting model has 50565 DOF.
- Model III: This model was defined geometrically using triparametric solid regions. No holes or fillets were included in the geometry definition. These solids were then meshed with 8-node hexahedral elements. The resulting model has 53202 DOF.

Note that these three models have different levels of simplifying assumptions in their PDE form, as well as different spatial meshes and different model orders (i.e. number of discrete solution variables). Model II is the least simplified model as compared to the design problem, but it probably contains the largest solution errors. This is because tetrahedral elements are generally known to possess much poorer accuracy than hexahedral elements for the same number of grid points. Model I is the next best mathematical model and utilizes hexahedral elements to improve mesh accuracy, but the mesh size is coarser than the other models. Model III has the most significant modeling simplifications, but is superior to the other models both in terms of grid spacing and grid regularity.

In order to integrate the results of the three models together, we chose equal weightings for the models. This reflects the mixture of subjective judgments on the relative merits of the models. In addition to the three models considered, the uncertainty in the elastic material modulus was also propagated through the analysis via linear sensitivity. The results are summarized in Figure 6. The solid line represents the mean value for the predicted modal frequencies, while the dashed lines represent the ± 2 standard deviation interval due to the uncertainty in the elastic modulus. Finally, the dot-dash lines are the total ± 2 standard deviation "uncertainty" interval due to both the modulus and the difference between the model predictions. Note that for some modes, there is a negligible contribution from the modeling variance, while for other modes this variance is larger than the uncertainty caused by the elastic modulus. As might be expected, the model "uncertainty" increases as the mode number increases. This type of analysis could be valuable in determining where the predictive accuracy of a set of models begins to break down.

5. Conclusions

This paper has presented a framework for the phases of modeling and simulation in structural dynamics and other computational mechanics disciplines. Using this framework, sources of uncertainties and errors have been categorized. Documenting these sources is the first step towards the goal of quantifying their effect on the accuracy of the simulation. Quantification of uncertainties and errors is, we

believe, a proper context for developing new principles for code verification and model validation and for ultimately enhancing the predictive accuracy of modeling and simulation. Furthermore, this paper begins to address the problem of assessing and propagating modeling errors and uncertainties, which are significant contributors to total simulation errors. This work is closely aligned with other ongoing research in nondeterministic methods, so that ultimately a comprehensive methodology for uncertainty quantification can be developed which applies to structural dynamics, as well as other computational mechanics, heat transfer, and coupled field problems.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the help of the Uncertainty Quantification Working Group at Sandia National Laboratories. This work was supported by the United States Department of Energy under contract No. DE-AC04-94AL85000. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy.

REFERENCES

1. Collins, J. D., Hart, G. C., Hasselman, T. K., and Kennedy, B., "Statistical Identification of Structures," *AIAA Journal*, Vol. 12, No. 2, 1974.
2. Hasselman, T. K., and Chrostowski, J. D., "Effects of Product and Experimental Variability on Model Verification of Automobile Structures," *Proceedings of the 15th International Modal Analysis Conference*, Orlando, FL, February 1997.
3. Oberkampf, W. L., Diegert, K. V., Alvin, K. F., and Rutherford, B. M., "Uncertainty in Error in Computational Simulations," to appear in *Proceeding of the 1998 ASME/AIAA Joint Thermophysics and Heat Transfer Conference*, Albuquerque, NM, 1998.
4. Neelamkavil, F., *Computer Simulation and Modeling*, New York, John Wiley & Sons, 1987.
5. Babuska, I., Strouboulis, T., Mathur, A., and Upadhyay, C.S., "Pollution Error in the h-Version of the Finite-Element Method and the Local Quality of A-Posteriori Error Estimators," *Finite Elements in Analysis and Design*, Vol. 17 (1994), pp. 273-321.
6. Yee, H. C., and Sweby, P. K., "Global Asymptotic Behavior of Iterative Implicit Schemes," *International Journal of Bifurcation and Chaos*, Vol. 4, No. 6, pp. 1579-1611, 1994.
7. Richardson, L. F., "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam," *Transaction of the Royal Society of London, Series A*, Vol. 210, pp. 307-357, 1910.
8. Draper, D., "Assessment and Propagation of Model Uncertainty (with Discussion)," *Journal R. Statistics Soc. B*, Vol. 57, No. 1, pp. 45-97, 1995.
9. Laskey, K. B., "Model Uncertainty: Theory and Practical Implications," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 26, No. 3, 1996.

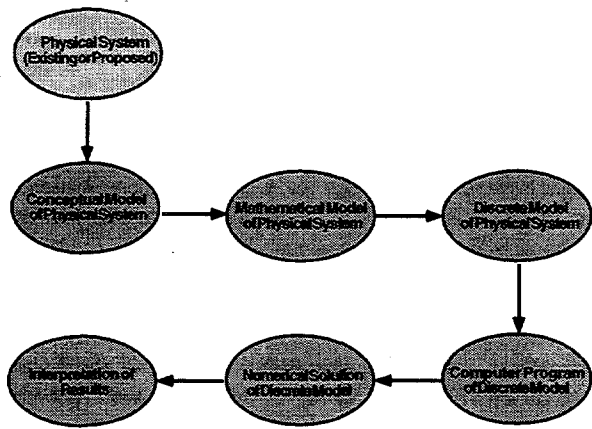


Figure 1: Proposed Phases of Modeling and Simulation

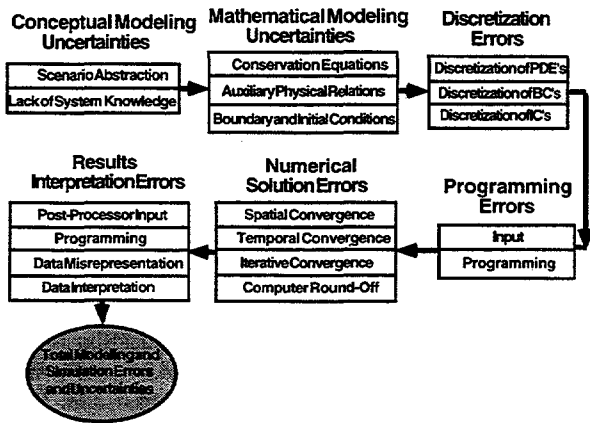


Figure 2: Sources of Uncertainty and Error in Modeling and Simulation

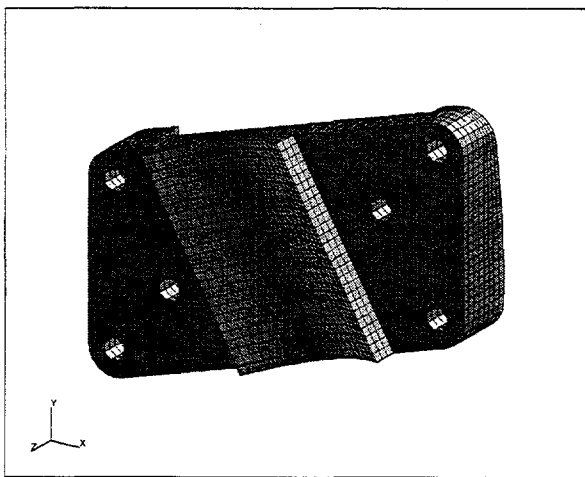


Figure 3: Model I: Paved/Extruded Hex Mesh w/ hole detail (no fillets) n=31554

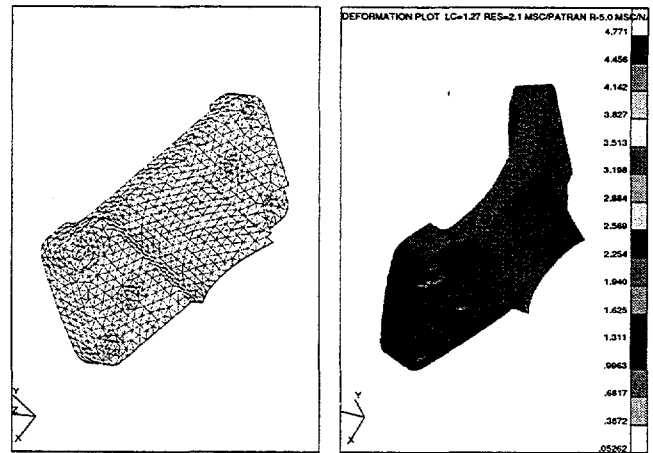


Figure 4: Model II: ProE geometry (holes and fillets) meshed w/10-node tet elements n=50565

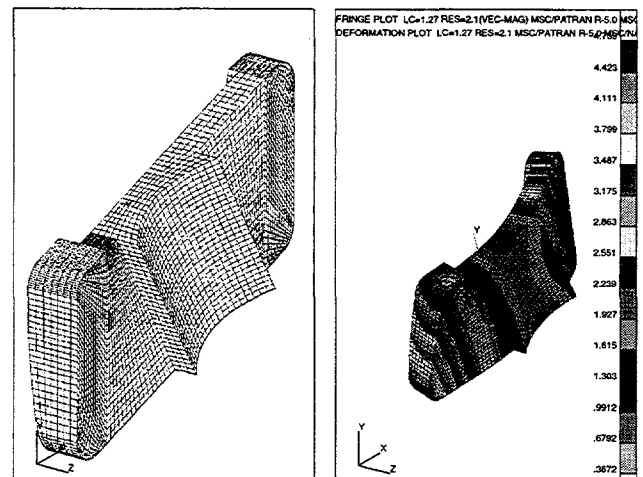


Figure 5: Model III: Triparametric Hex Mesh w/ no hole or fillet details n=53202

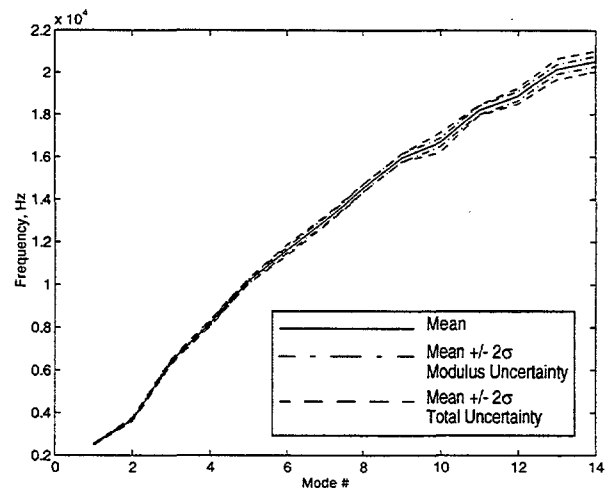


Figure 6: Uncertainty Analysis using Equal Assumed Probabilities

APPENDIX B:

**A Method for Treating Discretization Error in
Non-Deterministic Analysis
to appear in AIAA Journal**

Intentionally Left Blank

A METHOD FOR TREATING DISCRETIZATION ERROR
IN NON-DETERMINISTIC ANALYSIS

Kenneth F. Alvin*

Structural Dynamics and Vibration Control Dept.
Sandia National Laboratories, Albuquerque, NM

Abstract

A response surface methodology-based technique is presented for treating discretization error in non-deterministic analysis. The response surface, or metamodel, is estimated from computer experiments which vary both uncertain physical parameters and the fidelity of the computational mesh. The resultant metamodel is then used to propagate the variabilities in the continuous input parameters, while the mesh size is taken to zero, its asymptotic limit. With respect to mesh size, the metamodel is equivalent to Richardson extrapolation, in which solutions on coarser and finer meshes are used to estimate discretization error. The method is demonstrated on a one dimensional prismatic bar, in which uncertainty in the third vibration frequency is estimated by propagating variations in material modulus, density, and bar length. The results demonstrate the efficiency of the method for combining nondeterministic analysis with error estimation to obtain estimates of total simulation uncertainty. The results also show the relative sensitivity of failure estimates to solution bias errors in a reliability analysis, particularly when the physical variability of the system is low.

* Senior Member Technical Staff, Structural Dynamics and Vibration Control Dept., P.O. Box 5800, MS 0439, Albuquerque, NM 87185-0439; Senior Member AIAA

I. Introduction

Nondeterministic analysis methods are applied to simulations of physical systems in order to quantify the effects of random variations in system parameters and inputs on the predicted output of the simulation. Typically, nondeterministic methods are used to propagate probability or frequency distributions of continuous "physical" variables through a deterministic mapping, such as the discretized numerical solution of a system of partial differential equations (PDEs), plus boundary and initial conditions and auxiliary sub-models. In this case, it is important to have a verified and validated model structure through which to propagate these continuous variabilities. However, all finite discretized models possess some degree of discretization error, and often little or no attempt is made to estimate the magnitude of discretization error in the model. Furthermore, even when some error estimate is available, it is unclear how to apply that estimate to the ensemble of results computed during a nondeterministic analysis. Thus, the effect of discretization error in nondeterministic analysis is practically never treated at the present time.

There has, however, been a significant amount of attention devoted to the problem of estimating errors in numerical methods for solving deterministic partial differential equations. Among these methods are a posteriori error estimators^{1,2}, as well as such classical methods as Richardson extrapolation³⁻⁵. Of particular interest in this study is Richardson extrapolation, in which discretization errors are estimated from the numerical solution. Richardson extrapolation is extremely general in that it can be applied to any output of the model, as well as functionals of the solution. Its primary drawback is that it depends on knowledge of the formal convergence rate of the numerical method and requires that the mesh size used is fine enough that the higher-order terms of the error are negligible compared to the lowest-order term. For this reason, it often requires more than two mesh spacings on the same model to verify the convergence order and thus establish the validity of the extrapolation.

A conservative approach to accounting for discretization error in a nondeterministic analysis (which relies on the numerical solution of PDEs) would be to use Richardson extrapolation (or some other error estimator) for every combination of input values to the model. For example, we might perform a structural dynamics simulation in which the elastic modulus of some material in the design model has some inherent variability. Then, for each particular value of that parameter we could compute the response on two or more spatial discretizations and use these results in some to-be-determined way in our nondeterministic analysis. This approach would then increase the number of analyses to be performed by a factor of two to three. On the other hand, we might consider performing error estimation for only one particular value of the variable parameter (such as its mean value) and then apply that error estimate in a relative or absolute sense to the analyses performed for other values of the input. This approach would require only a modest increase in computational cost compared to the cost of the nondeterministic analysis itself, but cannot account for the dependence of the error estimate on the values of the parameters of the model. It should be noted that the investment in generating multiple discrete models with different mesh spacings might be much more significant than the cost of computing solutions on each of the meshes.

In the present study, an alternative approach is considered in which solutions are computed on different mesh sizes, but not for every parameter value in the nondeterministic analysis. Instead, both mesh size and parameter values are varied for the purpose of building a surrogate model, or metamodel, for interpolation. Once solutions are computed to build the metamodel, a regression is performed to obtain the coefficients of the metamodel. The metamodel can then be used in the nondeterministic analysis in place of the complex full-order model. For example, frequency distributions on input variables can be easily propagated through the metamodel via Monte Carlo analysis, since the cost of computing a response based on the metamodel is negligible. The treatment of discretization error is accomplished by extending traditional re-

response surface methods (RSM) for determining metamodels to include mesh size as a variable input parameter. Then the nondeterministic analysis can be performed for a mesh size of zero, which conforms to a higher-order accurate solution of the governing PDEs.

In this study, it has been found that including mesh size in the metamodel can be an efficient way to estimate discretization error while performing nondeterministic analysis. For example, a traditional metamodel with 3 continuous input variables might require 13 evaluations of the complex simulation with different parameter values to determine the coefficients of the metamodel. If we wished to minimize discretization error, we would perform those 13 evaluations on a fine mesh (i.e. with small element edge lengths). By extending that model to include element edge length as an input variable of the metamodel, we must now perform 25 evaluations of the complex simulation. However, only 6 of those evaluations are performed on the fine mesh model, while the other evaluations are performed on coarser meshes which require much less time to solve. Therefore, it is possible to determine the extended metamodel with less overall computational effort. Furthermore, the extended metamodel can be used to estimate the converged continuum solution, while the traditional metamodel determined strictly from evaluations of the fine mesh model still suffers some overall unquantified error due to the discretization.

The remainder of the study is organized as follows. First, the theory for estimating the effect of discretization error in the numerical solution of differential equations is reviewed. Secondly, the technique for using response surface methodology for nondeterministic analysis is presented. Then, this methodology is extended to include the dependence of the simulation on mesh size. Finally, results are presented for linear dynamics of a prismatic bar with 3 uncertain input parameters. The results demonstrate the importance of treating discretization error when estimating system reliability measures such as probability of failure. In the example, discretization error on the order of 1% of the response quantity results in a probability of failure estimate which is more than one order of magnitude smaller than the exact solution, an

error of more than 90%. Thus, the bias caused by small discretization errors can result in a significant over-prediction of reliability. The technique developed herein yielded an accurate estimate of the error in the probability of failure measure due to the discretization.

II. Discretization Error Models and Extrapolation-Based Error Estimation

Quantifying the effects of approximations on the numerical solutions of ordinary and partial differential equations has been a focus of research and analysis ever since numerical solution procedures were first developed. In Richardson's classic paper from 1910 on a finite difference solution to the partial differential equations associated with stresses in a dam structure³, he introduces a very general extrapolation procedure which estimates the leading error term from multiple solutions using different differencing steps. This procedure, since referred to as Richardson extrapolation, does not exactly estimate the desired continuum solution but rather, by eliminating the leading error term, increases the order of convergence of the numerical method. Richardson extrapolation is still used extensively as both an error estimation technique and as a verification tool for numerical solution methods⁵. The error estimate is simply taken as the difference between the basic numerical solution and the higher-order solution developed from multiple meshes, and has the same order of accuracy as the higher-order solution. It is therefore a reliable estimate as long as the leading error term is the primary source of error. Its strengths are that it estimates the total discretization error, including both the local and global effects of discrete approximations (including finite element approximations) throughout the entire problem domain and boundary conditions. Furthermore, it can also estimate errors not only on the solution field variables but also on many linear functionals of the solution field. Its drawback is that it requires uniform mesh refinement with at least three mesh spacings. Such a requirement may be difficult to meet for some complex problems.

With the advent of the finite element method, other approaches to discretization error estimation have been developed. Most of these procedures are broadly classed as a posteriori error estimation methods¹⁻², because they locally post-process the numerical solution in order to estimate certain norms of the solution error. These methods, which were developed primarily as indicators for adaptive mesh refinement procedures, are usually based on multiple local mesh refinement problems using such techniques as superconvergent patch recovery to estimate local higher-order accurate estimates of solution gradients. These estimates can be compared the recovered solution gradients from the normal finite element interpolation functions and the differences used to estimate local or global solution error measures. The relative localized contributors to the error measure help identify mesh regions which require more refinement, while the global norm measure is used in some sense as a stopping criterion for mesh refinement. One drawback of a posteriori error estimation for global error, however, is that since it relies on local computations it cannot account for so-called pollution error; that is, the propagation of local errors through the global domain of the problem. These errors cannot be detected and, if significant, will reduce the reliability of the error estimate.

It should be noted that the underlying polynomial model for global discretization error is the same regardless of the error estimation procedure and has been developed repeatedly in the various texts on finite element and finite difference methods⁶⁻¹⁰. It is written as

$$y_D = y + \alpha h^q + O(h^{q+1}) \quad (1)$$

where y is the solution to the given differential equation defined on a continuous domain in space and/or time, y_D is the approximate numerical solution on a mesh with characteristic size h , q is the order of accuracy for the numerical method, and α is a sensitivity parameter of the solution which is independent of h . The order of accuracy q is a characteristic of the numerical method which reflects the quality of the interpolation functions for the solution quantity of interest. For example, typical low-order finite elements

for structural mechanics with linear shape functions lead to displacement solutions which are second-order accurate, while the corresponding stress solutions based on the gradient of the displacement interpolation functions are first-order accurate. The order of accuracy of the numerical method is independent of the specific problem only to the extent that we acknowledge the existence of higher-order terms in h . The magnitude of these higher-order terms are very much problem dependent and determining their importance is a necessary consideration in error estimation. This error model is also independent of the dimensionality of the problem; it is equally valid for 1-D and 3-D calculations, although it is not equally easy to estimate errors for 1-D and 3-D problems. That is, for a uniform doubling of the mesh fidelity, the number of algebraic equations for the 1-D problem would approximately double, while the size of the 3-D problem would increase by a factor of 8 (since the mesh would double in each of the 3 spatial directions).

Extrapolation-based estimation of discretization error proceeds as follows. Calculations are performed on at least 3 meshes, where the finer meshes are uniformly refined with respect to the coarser meshes. Then, the error on the finest mesh is given as the difference between that solution and the extrapolated solution

$$e = y_{fine} - y_{extrap} = \alpha h_{fine}^q \quad (2)$$

where α is estimated from the difference between the nominal and fine mesh solutions, viz.

$$\alpha = \frac{y_{fine} - y_{nominal}}{h_{fine}^q - h_{nominal}^q} \quad (3)$$

At least 3 mesh solutions are required in order to verify the order of accuracy of the method, q , which is assumed in the previous equations. This is verified by estimating q , which for mesh doubling is given by

$$q = \frac{\log\left(\frac{y_{nominal} - y_{coarse}}{y_{fine} - y_{nominal}}\right)}{\log 2} \quad (4)$$

Verifying the assumed value of q is equivalent to showing that the higher order terms in the error model are negligible compared to the leading q -order term. One might also seek to estimate two error terms using more than three mesh solutions in order to either improve the error estimate or enable it to be computed on a coarser set of meshes.

III. Response Surface Methodology for Non-Deterministic Analysis

Montgomery¹¹ notes that

Response surface methodology, or RSM, is a collection of mathematical and statistical techniques that are useful for the modeling and analysis of problems in which a response of interest is influenced by several variables and the objective is to optimize the response....

The process yield is a function of the levels..., say

$$y = f(x_1, x_2) + e$$

where e represents the noise observed in the response y . If we denote the expected response by

$E[y] = f(x_1, x_2) = h$, then the surface represented by

$$h = f(x_1, x_2)$$

is called the response surface.

In the present context, we will use RSM as a surrogate or metamodel for the complex physics model of interest, and we will estimate the coefficients of the response surface by performing a limited number of analyses of the complex model. Our goal is to use the response surface to propagate uncertainties in the variables to determine a distribution of the response quantity. From this distribution we can make estimates of failure probability or other statistics of interest.

The use of RSM is logically coupled to Design of Experiments (DOE). In order to apply RSM we must determine a selection of input vectors for the complex simulation. Designing the input vectors for the suite of simulations to be run is the objective of experiment design. A logical goal of DOE is to minimize the variance of the error between the response surface and the discrete responses of the complex model. It is important, however, to also consider the frequency distributions of the input parameters, as well as the type of statistic to be computed from the distribution of the output. Finally, as noted by Sacks¹² there are fundamental differences between physical experiments and computer experiments which influence the issue of experiment design.

In this study, we focus on a simple global surface model which includes second-order terms in the parameters. The model response is given by

$$y = \beta_0 + \sum_i \beta_i p_i + \sum_i \sum_j \beta_{ij} p_i p_j \quad (5)$$

where $\beta_0, \beta_i, \beta_{ij}$ are coefficients of the metamodel to be estimated from analyses performed on the complex model. It should be noted that this metamodel form is just one possibility; other functions could be considered as well as the finite element lattice sampling approach of Romero¹³. In addition, a metamodel could be constructed using solution derivatives obtained from the analysis code, although to obtain quadratic term coefficients it would be necessary to compute the second-order derivatives also.

To this standard metamodel form we will apply the well-known Box-Behnken experiment design¹⁴, which dictates a set of input vectors for which the parameters p_i take on nominal, high, or low values. It is suggested that these levels be taken as the mean and the mean plus or minus one to two standard deviations, respectively. Furthermore, the number of simulation runs is somewhat greater than the number of coefficients being estimated, so that a least-squares estimate for the metamodel coefficients is obtained. The Box-Behnken designs for 3 and 4 input variables are given in Table 1 and Table 2, where 0, +1, -1 represent the nominal, high and low values, respectively. Other possible designs are Central composite and

sub-optimal iterate selection algorithms such as Effective Independence¹⁵ and Subset selection¹⁶. Once an experiment design has been determined, and the computer experiments are performed, the coefficients β_{ij} of the metamodel are estimated, typically using a least-squares method.

IV. Extension of RSM to include Mesh Size

From Section II, the model for spatial discretization error is given as

$$y(h) = y_{exact} + \alpha h^q + O(h^{q+1}) \quad (6)$$

where h is the characteristic mesh spacing or element length, $y(h)$ is the numerical solution resulting from the mesh, y_{exact} is the exact solution of the corresponding continuum model, q is the formal order of the method, and α is some unknown factor. Assuming knowledge of method order q , we could easily construct a metamodel from a small number of simulation runs with different values of h . Since the only unknowns are α and y_{exact} , we require only two different mesh spacings. With three different mesh spacings, we could also confirm the method order q . This approach of estimating the terms of the error model using the results from different mesh spacings is in fact just the classical Richardson extrapolation method.

Thus, given the similarity between RSM and Richardson extrapolation, it is reasonable to combine the two methods into a larger metamodel form in order to treat discretization error within the context of non-deterministic analysis. The form of the extended metamodel is taken to be

$$y(\mathbf{p}, h) = \beta_0 + \sum_i \beta_i p_i + \sum_i \sum_j \beta_{ij} p_i p_j + \alpha_0 h^q + \sum_i \alpha_i p_i h^q \quad (7)$$

which allows for the metamodel to account for the coupling between the constant and linear terms of the nominal metamodel and the discretization error, but neglects the terms of $O(p^2 h^q)$ and above.

A. Experiment Design for Extended Metamodel

For experiment design purposes, we will again use the Box-Behnken design, where the number of input variables is increased by one for input value h . This requires that we develop discretized models with 3 different mesh spacings. As with Richardson extrapolation, it is usually desirable to perform mesh doubling. The values $-1, 0, +1$ with respect to h are strictly qualitative and correspond to coarse, nominal and fine mesh models, respectively. For example, re-interpreting Table 2 for the case of 3 input parameters plus variation in mesh fidelity, we have the experiment design given in Table 3. The value h/h_{nom} implicitly assumes the use of mesh doubling, but this is not specifically required.

The efficiency of the present method can be seen in examining Tables 1 and 3. When the metamodel is extended by one input variable in order to include the mesh size parameter h , the total number of analyses increases, but the number performed with a fine mesh is only a small fraction of the total and less than the total number of runs of the non-extended metamodel. Thus, a traditional response surface (without h) estimated by analyses performed using the fine mesh can actually be less efficient than extending the model to include h , since the total computational effort is dictated by the number of fine mesh analyses performed. This comparison is summarized in Table for different numbers of continuous input variables. Note that as the number of input parameters increase, the efficiency gains become dramatic.

B. Verification of the Assumed Order of Accuracy

In a real sense, the nominal mesh model, together with the coarse mesh model, is being used to infer or interpolate the effects of the parameter variations which would be seen at the fine mesh level. The mechanism for performing this inference is the form of the response surface, which is grounded in the error model Eqn. (6). This error model is valid, however, only in the asymptotic range and thus imposes a restriction on the fidelity of the meshes considered. In a sobering demonstration of the limitations of Richardson's method, Oberkampf and Blottner¹⁷ showed that for a particular system of nonlinear equations

with large local gradients, reaching the asymptotic range for Richardson's method required very fine grids with relative errors on the order of 0.1%. Therefore, computational meshes which simply meet an analyst's subjective criteria for "goodness" cannot be immediately assumed to meet the asymptotic range requirements of Richardson's method and the present extended response surface technique. It is important to assess the convergence characteristics for the range of meshes considered and demonstrate the formal convergence of the method before proceeding with extrapolative procedures.

There are two approaches to verifying the convergence order using the experiment design procedure discussed above. Examining Table 3 for the particular case of 3 metamodel input parameters, there are 6 points in the design space where solutions are computed using both the coarse and fine meshes of the model. Two mesh sizes are not sufficient, however, to independently estimate both the mesh error and the convergence order q . In order to independently estimate q we also require a solution at these design points for the nominal mesh. These can be obtained by performing additional computer experiments using the nominal mesh at these design points (corresponding to extreme values of each parameter). Alternatively, we could use the results from the runs on the nominal mesh to build a traditional metamodel with respect to the input parameters conditioned on the nominal mesh, and then use that metamodel to interpolate the response for the nominal mesh at the design points where coarse and fine mesh runs have been performed.

V. Non-Deterministic Analysis with Extended Metamodel

The final stage of the non-deterministic analysis involves the use of the metamodel to propagate the parametric uncertainties through to the simulation output. From the distribution of the output, probabilistic or statistical quantities such as probability of failure, expected output value and variance can be estimated. One straightforward approach is what Romero¹³ terms decoupled Monte Carlo (DMC) analysis. Decoupled refers to the fact that the building of the response surface from complex simulation runs and the Monte Carlo analysis using the response surface are separate activities. For the Monte Carlo analysis, we would

generate a very large number of samples from the joint distribution function of the uncertain variables and compute the metamodel response for each sample to build the output distribution. The only modification required for the extended metamodel is determining how to sample the mesh spacing parameter h .

Of course, as a solution method parameter, h is not a physical variable and has no associated distribution. Instead, we will use variations in h to estimate bias error in the non-deterministic solution. The nominal or fine mesh-based non-deterministic solution is obtained by holding h fixed at $h_{nominal}$ or h_{fine} respectively. Then an extrapolated non-deterministic solution is obtained by holding h fixed at zero. Finally, the biases in the nominal or fine mesh-based non-deterministic solutions are given by the differences between these solutions and the extrapolated ($h = 0$) solution. While this procedure involves extrapolation of the metamodel, which is a practice normally avoided in such use of function approximations, the potential problems related to extrapolation are mitigated in two ways. First, by the verification of accuracy order q , which specifically addresses the validity of the error model form for use in extrapolation, and second, by the use of the extrapolation strictly as a means to estimate solution error, rather than as a means to obtain a more exact solution.

VI. Example: 1-D Prismatic Bar

The first example is a uniform prismatic bar with fixed-free boundary conditions as shown in Figure 1. The objective of the analysis is to estimate the probability of failure of the bar, where failure is defined by the condition $f_3 < 16,000$ Hz, where f_3 is the frequency of the third mode of vibration. The physical parameters defining the system are given in Table 5, where the coefficient of variation is equal to the standard deviation of the distribution divided by the mean. The mesh chosen for the nominal model is also shown in Figure 1. Note that the mesh was chosen to be non-uniform. There are two reasons for this. First, in most complex problems it is difficult or impossible to achieve a uniform grid, and often the mesh is refined in some areas to enhance accuracy for some output of the model. Secondly, it is understood that nu-

merical methods tend to behave better in a theoretical sense on uniform grids, thus the non-uniform grid in this problem is intended to make a very simple problem somewhat more difficult. From the fixed end to the free end, the element lengths are: 4 elements @ 1.25 in, 4 elements @ 1.00 in, 4 elements @ 0.75 in, 4 elements @ 0.50 in, 4 elements @ 0.25 in. Thus there is a total of 20 elements and 20 degrees of freedom (d.o.f.) in the nominal model. The coarse and fine meshes for this problem are related to the nominal mesh by doubling and halving the element lengths, respectively. Therefore, the coarse mesh has 10 d.o.f. and the fine mesh has 40 d.o.f.

The experiment design given in Table 3 was used for estimating the extended metamodel, with the interpolation function given by Eqn. (7) and $q = 2$ as the formal order of the method. Then the decoupled Monte Carlo analysis was performed with 100,000 samples of the joint density function of E , ρ , and L . All samples were evaluated at $h = 0$. For comparison, a direct Monte Carlo analysis was performed using the exact continuum solution¹⁸:

$$f_3 = \frac{5\sqrt{E/\rho}}{4L} \quad (8)$$

Also for comparison, two approximate solutions are treated. First, a direct Monte Carlo analysis was performed using the nominal mesh model. Second, a decoupled Monte Carlo analysis was performed for the fine mesh model, where the response surface was estimated with respect to the 3 physical variables E , ρ , and L using Eqn. (5) and the experiment design given in Table 1. Finally, histograms of each of the 4 Monte Carlo analyses were computed using 1000 bins, and the results input to a kernel density estimator to arrive at estimates of the output probability density functions. These final results are shown in Figure 2.

VII. Discussion of Results

The results from this example are summarized in Table 6 and Table 7. In Table 6, for each of the approaches, the total number of finite element analyses are given, as well as the number of analyses using the fine resolution grid. This is important because the computational effort is primarily associated with the number of finite element analyses performed on the most refined grid. Then the total number of computations required for the nondeterministic analysis are given in Mflop (10^6 floating point operations). Finally, the error in the estimated mean of the distribution is given, as well as the estimated probability of failure, which is the statistic of interest in the problem. The extended metamodel result given is the estimated solution at $h = 0$ obtained via extrapolation using the extended response surface methodology (ERSM). Note first of all that the continuum solution is the exact result, subject to Monte Carlo sampling errors (which are small because of the number of samples). The rest of the methods are given down the table in order of their accuracy.

The extended RSM method is not only the most accurate approximation, but is also the most efficient. There is some remaining bias in this solution since extrapolation does not calculate the exact solution, but rather a high-order accurate estimate of the exact solution. This bias could be further reduced by either refining the set of meshes considered or by performing additional mesh variations and using the results to estimate higher-order error terms. The extrapolated solution is sufficiently accurate, however, for the purpose of error estimation. The more traditional RSM based on the fine mesh is still a reasonable result, although it requires more computations and is somewhat less accurate than the extended RSM. Finally, the direct Monte Carlo method based on the nominal mesh is not only very inefficient, but also leads to a significant error in the probability of failure estimate. It should be noted that all of the results have negligible

sampling errors, and the errors introduced by the limitations of the metamodel forms were also small. The errors shown on both the estimated mean and failure probabilities are primarily due to the biases in the finite element solutions caused by the fidelity of the discretizations.

In Table 7, the results are presented in terms of estimating the bias error on the non-deterministic solution using the extended metamodel. The first result given for the fine mesh is obtained from the extended metamodel by fixing h at h_{fine} . The estimated bias on this solution is then obtained by comparison to the extrapolated solution using the extended metamodel. The true bias is given by comparison to the continuum solution. The second result given for the fine mesh is that obtained by estimating a traditional response surface without the mesh size parameter. For this approach, there is no estimate of the bias error due to discretization because there are no other mesh solutions considered. The true bias is again obtained by comparison to the continuum solution. Note that the fine mesh solutions obtained from both response surface models are essentially identical, but, referring back to Table 6, the solution obtained using the extended metamodel not only required less finite element computations on the fine mesh, but also allowed for the estimation of the discretization bias error on the solution. The last two rows of the table give similar results for the nominal mesh size. Again, the accuracy of the extended metamodel is seen by its similarity to the direct Monte Carlo results for the nominal mesh (i.e. with no function approximation employed).

Finally, estimates for the order of accuracy of the set of discretizations is given in Table 8. These estimates were obtained by fitting an independent response surface model to the nominal mesh experiments and then using this model to interpolate the nominal mesh results to the experiments performed on the coarse and fine meshes. Then the three different mesh solutions for each of 6 points in the continuous parameter space were used with Eqn. (4) to estimate q , the exponent of the characteristic mesh size h in the discretization error model. These results verify that the set of discretizations are within the sphere of convergence of the numerical method, for which $q = 2$.

Apart from the comparison of methods, one of the most interesting aspects of these results is the effect the prediction error, as measured by the error in the estimated mean of the distribution, has on the relative error in the probability of failure estimate. In the case of the direct Monte Carlo analysis based on the nominal mesh, the discretization error results in a prediction error of about 1% of the exact solution. This would normally be considered more than adequate for engineering purposes. However, this error leads to a probability of failure estimate which is over one order of magnitude smaller than the exact solution. Thus, the reliability of the system might be judged to be more than 10 times higher than its actual reliability. The heightened sensitivity is a function of the probability of failure magnitude we are trying to estimate, as well as the coefficient of variation (the ratio of the standard deviation of the distribution to the mean value) of the simulation output used in the failure calculation.

This can be seen by considering the effect of a relative bias error on the value of a standard normal random variable z at the point of failure, viz.

$$z = \frac{y_F - \mu}{\sigma} \quad (9)$$

where μ is the mean of the distribution of the response y , σ is the standard deviation of the distribution of y , and y_F is the value of the response which defines the failure boundary. If the computational prediction of y is given by y_c and suffers a relative bias error ε , we have

$$\begin{aligned} y_c &= y(1 + \varepsilon) \\ \mu_c &= \mu(1 + \varepsilon) \\ \sigma_c &= \sigma(1 + \varepsilon) \end{aligned} \quad (10)$$

Substitution into Eqn. (9) yields

$$z_c = \frac{y_F - \mu}{\sigma} = \frac{z - \frac{\varepsilon}{COV}}{1 + \varepsilon} \approx z - \frac{\varepsilon}{COV} \quad (11)$$

for small ε , where $COV = \sigma/\mu$ is the coefficient of variation of the distribution y . Using this result, we

can estimate the relative error in the probability of failure due to the computational bias. Using a first-order Taylor series expansion of the cumulative distribution function, we have

$$\begin{aligned}\Phi(z_c) &= \Phi\left(z - \frac{\varepsilon}{COV}\right) \approx \Phi(z) - \left(\frac{\varepsilon}{COV}\right) \frac{d\Phi(z)}{dz} \\ \therefore \Phi(z_c) &\approx \Phi(z) \left(1 - \frac{\varepsilon}{COV} \left(\frac{\phi(z)}{\Phi(z)}\right)\right)\end{aligned}\tag{12}$$

So the relative error in the failure probability is

$$\frac{\Delta\Phi(z)}{\Phi(z)} \approx -\left(\frac{\varepsilon}{COV}\right) \left(\frac{\phi(z)}{\Phi(z)}\right) \approx \frac{\varepsilon}{COV} z\tag{13}$$

where $\frac{\phi(z)}{\Phi(z)} \approx -z$ is a reasonably good approximation for $z < -3$ and a lower bound for all $z < 0$. This approximation to the relative error in the probability of failure clearly demonstrates that the sensitivity of the probability of failure to bias error is exacerbated by two factors: a small coefficient of variation (or one which is of similar magnitude to the bias error ε), and a small failure probability for the exact solution. For the sample problem on the fine mesh, with $COV = 1.22\%$, $\varepsilon = 0.27\%$, and $z = -1.92$, the relative error in the probability of failure approximated by Eqn. (13) is -42.5% , which is equivalent to the results in Table 7.

Although the example problem studied herein is relatively simple, there are no issues which limit the present procedure to 1-D problems or to few parameters. Richardson extrapolation has been used extensively in 2-D and 3-D computational fluid dynamics, both for verification of codes and calculations⁵. Furthermore, as noted previously and in Table , there is a potential for improvement in efficiency by utilizing multiple meshes over a large dimensional parameter space, rather than limiting a probabilistic analysis to function evaluations on fine meshes alone. It should be noted that not every potential response quantity will use the same error model; stress and displacement for example differ in their order of accuracy and

require different coefficients for h in the error models. But the basic approach is the same, given an understanding of the formal order of accuracy of the numerical methods employed in the computational analysis.

VIII. Summary and Conclusions

A technique has been presented for treating discretization error in nondeterministic analysis. The technique involves the use of response surface methodology (RSM), in which a metamodel representation of a complex simulation model is estimated from a limited number of computer experiments on the complex model. The metamodel is then extended to include the characteristic element edge length as an model input parameter. The extended metamodel is used to propagate the variabilities in the continuous input parameters, while the effect of discretization error is estimated by taking the mesh size to zero, its asymptotic limit. The technique is demonstrated on a one dimensional prismatic bar, in which the uncertainty in the frequency of the third mode of vibration is estimated by propagating variations in the elastic modulus and mass density of the material, together with variation in the total length of the bar. Results are compared to the closed-form solution, a direct Monte Carlo analysis using a nominal mesh size, and a traditional response surface without a mesh size parameter built from computer experiments using a fine mesh size. The results demonstrate the importance of treating discretization error when estimating system reliability measures such as probability of failure, and the efficiency of the present technique for combining nondeterministic analysis with error estimation to obtain more accurate estimates of total simulation uncertainty.

Acknowledgments

The author would like to thank Bill Oberkampf and Brian Rutherford at Sandia, as well as Dan Kammer of UW-Madison, for contributing to this work through many discussions, as well as reviewing its content. Any errors and omissions, however, are the sole responsibility of the author. The support of the

Sandia Computer Sciences LDRD program is also gratefully acknowledged. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

REFERENCES

1. Zienkiewicz, O. C., and Zhu, J. Z., "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis," *Int. J. Numer. Methods Eng.*, 24, 337-357 (1987).
2. Babuska, I., Strouboulis, T., Upadhyay, C. S., Gangaraj, S. K., and Copps, K., "Validation of A Posteriori Error Estimators by Numerical Approach," *Int. J. Numer. Methods Eng.*, 37, 1073-1123 (1994).
3. Richardson, L. F., "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations with an Application to the Stresses in a Masonry Dam," *Trans. R. Soc., Lond., Ser. A*, 210 (1910).
4. Richards, S. A., "Completed Richardson Extrapolation in Space and Time," *Comm. Numer. Methods Eng.*, 13, 573-582 (1997).
5. Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Hermosa, Albuquerque, NM, 1998, pp. 108-136.
6. Isaacson, E., and Keller, H. B., *Analysis of Numerical Methods*, John Wiley and Sons, New York, 1966, pp. 364-383.
7. Iserles, A., *Numerical Analysis of Differential Equations*, Cambridge University Press, Cambridge, UK, 1996, pp. 7-8.
8. Hughes, T. J. R., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987, pp.188-190.

9. Strang, G., and Fix, G. J., *An Analysis of the Finite Element Method*, Wellesley-Cambridge Press, Wellesley, MA, 1988, pp. 165-171.
10. Cook, R. D., Malkus, D. S., and Plesha, M. E., *Concepts and Applications of Finite Element Analysis*, 3rd Ed., John Wiley and Sons, New York, 1989, pp. 553-563.
11. Montgomery, D. C., *Design and Analysis of Experiments*, 3rd Ed., John Wiley & Sons, New York, 1991.
12. Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, pp. 409-435 (1989).
13. memo, Sandia National Labs., from V. J. Romero and S. D. Bankston, to D. R. Martinez, subject: Application of decoupled Monte Carlo analysis with Finite Element / Lattice Sampling response surface for multimodal test problem (Function 1), dated July 7, 1997.
14. Fox, E. P., "Issues in Utilizing Response Surface Methodologies for Accurate Probabilistic Design," Paper AIAA-96-1496-CP, AIAA, Washington DC (1996).
15. Kammer, D. C., "Sensor Placement for On-Orbit Modal Identification and Correlation of Large Space Structures," *Journal of Guidance, Dynamics and Controls*, 14(2), pp. 251-259 (1991).
16. Red-Horse, J. R., "Modal Experiment Design Using Subset Selection," *Proceedings of the 14th Int. Modal Analysis Conf.*, Detroit, MI, Feb. 1996.
17. Oberkampf, W. L., and Blottner, F. G., "Issues in Computational Fluid Dynamics Code Verification and Validation," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 687-695.
18. Weaver, W., Timoshenko, S. P., Young, D. H., *Vibration Problems in Engineering*, 5th Ed., John Wiley & Sons, New York, 1990, pp. 369-370.

Table 1: Box-Behnken Design for 3 Input Variables

Runs	p(1)	p(2)	p(3)
1-4	+/- 1	+/- 1	0
5-8	0	+/- 1	+/- 1
9-12	+/- 1	0	+/- 1
13	0	0	0

Table 2: Box-Behnken Design for 4 Input Variables

Runs	p(1)	p(2)	p(3)	p(4)
1-4	+/- 1	+/- 1	0	0
5-8	0	+/- 1	+/- 1	0
9-12	0	0	+/- 1	+/- 1
13-16	+/- 1	0	+/- 1	0
17-20	0	+/- 1	0	+/- 1
21-24	+/- 1	0	0	+/- 1
25	0	0	0	0

Table 3: Box-Behnken Design for 3 Input Variables plus Mesh Fidelity Variation

Runs	p(1)	p(2)	p(3)	Mesh	h/h_{nom}
1-4	+/- 1	+/- 1	0	nominal	1.0
5-8	0	+/- 1	+/- 1	nominal	1.0
9-10	0	0	+/- 1	coarse	2.0
11-12	0	0	+/- 1	fine	0.5
13-16	+/- 1	0	+/- 1	nominal	1.0
17-18	0	+/- 1	0	coarse	2.0
19-20	0	+/- 1	0	fine	0.5
21-22	+/- 1	0	0	coarse	2.0
23-24	+/- 1	0	0	fine	0.5
25	0	0	0	nominal	1.0

Table 4: Efficiency Gains by using Extended Metamodel with Mesh parameter and Box-Behnken Experiment Design

Number of Metamodel inputs (not including h)	Efficiency ^a
3	2.2
4	3.1
5	6.7
6	8.3

a. $\left[\frac{\# \text{ fine mesh runs in extended model}}{\text{total \# runs in non-extended model}} \right]^{-1}$

Table 5: Parameter Information for 1-D Bar

Parameter	Distribution	Mean Value	Coefficient of Variation
Elastic Modulus	E	Normal 10×10^6 psi	1.0%
Mass Density	r	Normal 0.000259 lb-s ² /in ²	1.0%
Total Length	L	Normal 15.0 in	1.0%

Table 6: Nondeterministic Analysis Results for 1-D Bar Example

Method	Total # FEM Analyses	# Fine Grid FEM Analyses	# Computations (MFlop)	% Error in mean(f_3)	Estimate of $P_{failure}^a$
Continuum Solution	n/a	n/a	n/a	n/a	2.75%
Extended RSM (with $h=0$)	25	6	59	0.01%	2.69%
RSM on fine mesh	13	13	94	0.27%	1.61%
Direct MC on nominal mesh	100,000	0	90,000 (est)	1.09%	0.23%

a. $P_{failure} = P(f_3 < 16,000 \text{ Hz})$

Table 7: Bias Error Estimation for Non-Deterministic Analysis

Mesh	Estimated bias error in $\text{mean}(f_3)$	Actual bias error in $\text{mean}(f_3)$	Estimated bias error in $P_{failure}$	Actual bias error in $P_{failure}$
Fine (using ERSM)	0.27%	0.27%	-41.3%	-42.5%
Fine (using RSM)	unknown	0.27%	unknown	-41.5%
Nominal (using ERSM)	1.07%	1.08%	-91.6%	-91.7%
Nominal (using DMCS)	unknown	1.09%	unknown	-91.6%

Table 8: Order of Accuracy Estimates from Experiments used to Estimate Extended Metamodel

Reference Runs	p(1)	p(2)	p(3)	q
9,11	0	0	+1	1.9816
10,12	0	0	-1	1.9813
17,19	0	+1	0	1.9817
18,20	0	-1	0	1.9811
21,23	+1	0	0	1.9810
22,24	-1	0	0	1.9819

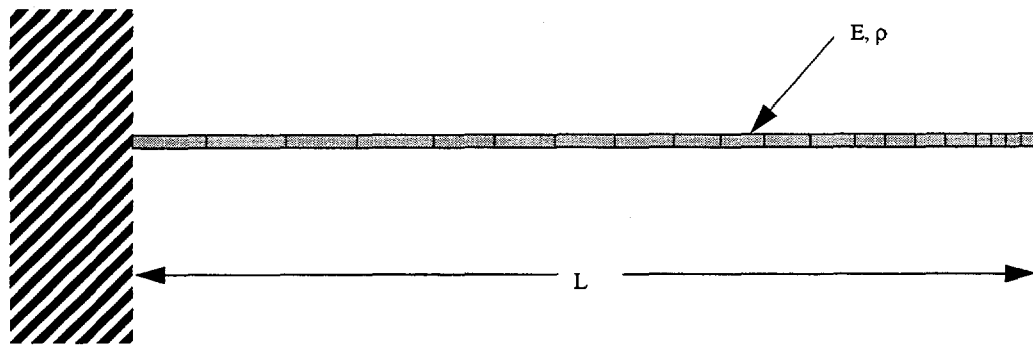


Figure 1: 1-D Prismatic Bar Model with Nominal Mesh

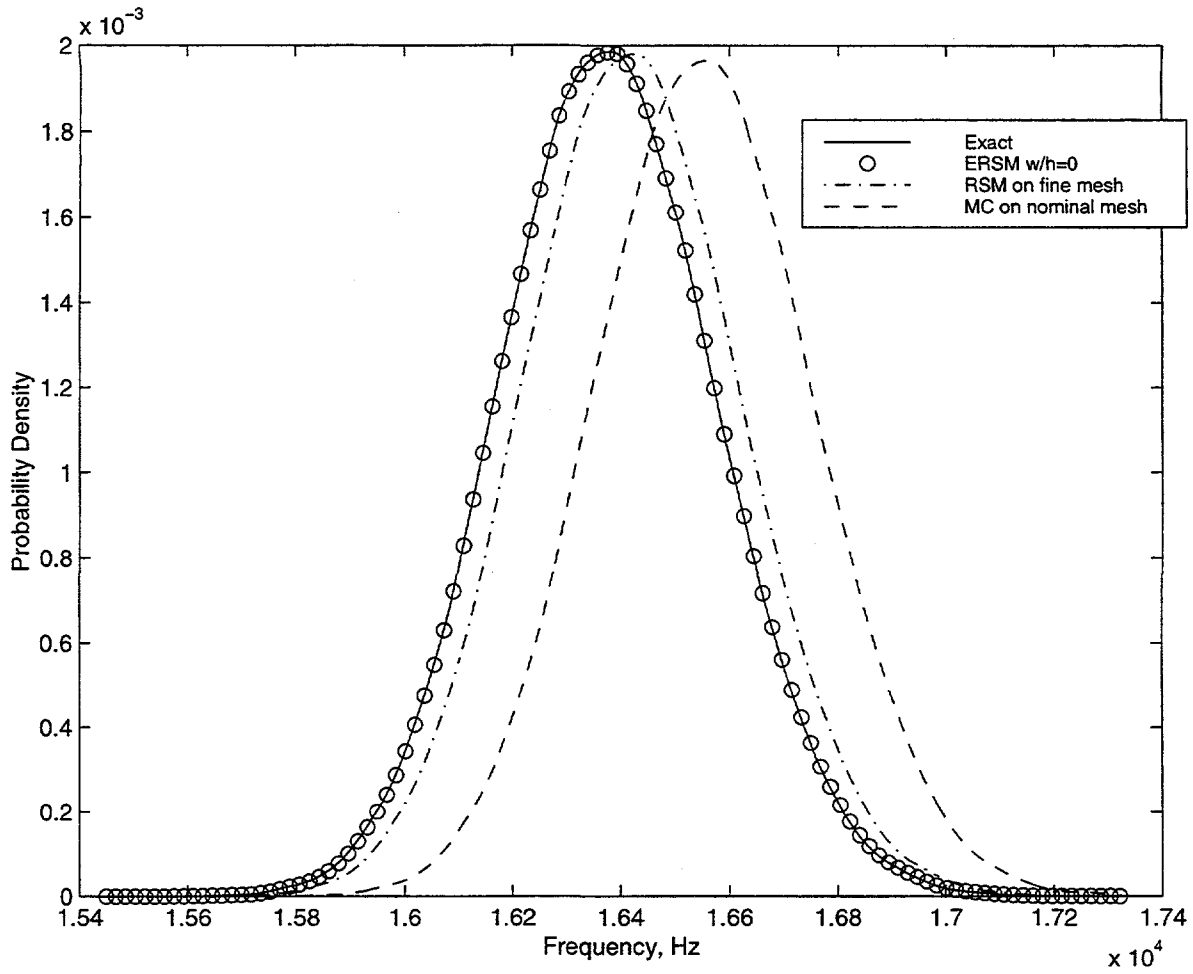


Figure 2: Results from Nondeterministic Analysis of 1-D Bar

LIST OF FIGURES

Figure 1: 1-D Prismatic Bar Model with Nominal Mesh

Figure 2: Results from Nondeterministic Analysis of 1-D Bar

Intentionally Left Blank

APPENDIX C:

A New Methodology for the Estimation of Total Uncertainty in Computational Simulation

**presented at the 1999 AIAA Forum on
Non-deterministic Approaches,
St. Louis, MO, April 1998**

Intentionally Left Blank

A New Methodology for the Estimation of Total Uncertainty in Computational Simulation

William L. Oberkamp^{*}
Aerosciences Dept.

Sharon M. DeLand^{**}
Mission Analysis and Simulation Dept.

Brian M. Rutherford[†]
Statistics and Human Factors Dept.

Kathleen V. Diegert^{††}
Statistics and Human Factors Dept.

Kenneth F. Alvin[§]
Structural Dynamics Dept.

Sandia National Laboratories
Albuquerque, New Mexico 87185-0825

ABSTRACT

This paper develops a general methodology for estimating the total uncertainty in computational simulations that deal with the numerical solution of a system of partial differential equations. A comprehensive, new view of the general phases of modeling and simulation is proposed, consisting of the following phases: conceptual modeling of the physical system, mathematical modeling of the conceptual model, discretization and algorithm selection for the mathematical model, computer programming of the discrete model, numerical solution of the computer program model, and representation of the numerical solution. In each of these phases, general sources of variability, uncertainty, and error are identified. Our general methodology is applicable to any discretization procedure for solving ordinary or partial differential equations. To demonstrate this methodology, we describe a system-level analysis of an unguided, rocket-boosted, aircraft-launched missile. In the conceptual modeling phase, a wide variety of analysis options are considered, but only one branch of the analysis is computed: rigid body flight dynamics. We choose two parameters as nondeterministic elements of the system: one has variability that is treated probabilistically and one has uncertainty that is represented as a set of possible alternatives. To illustrate mathematical modeling uncertainty, we pursue two models with differing levels of physics: a six-degree-of-freedom and a three-degree-of-freedom model. We also examine numerical solution error in the analysis, which is ubiquitous in computational simulations.

^{*} Distinguished Member Technical Staff, Associate Fellow

^{**} Principle Member Technical Staff

[†] Senior Member Technical Staff

^{††} Manager

[§] Senior Member Technical Staff, Senior Member

Copyright © 1999 The American Institute of Aeronautics and Astronautics Inc. All rights reserved.

1. INTRODUCTION

Historically the primary method of evaluating the performance of a proposed system design has been to build the design and then test it in the use environment.

This testing process is often iterative, as design flaws are sequentially discovered and corrected. The number of design-test iterations has been reduced with the advent of computer simulation through numerical solution of the mathematical equations describing the system behavior. Computational results can identify some flaws and they avoid the difficulty or safety issues involved in conducting certain types of physical tests. Examples include the atmospheric entry of a space probe into another planet, structural failure of a full-scale containment vessel of a nuclear power plant, failure of a bridge during an earthquake, and exposure of a nuclear weapon to certain types of accident environments.

Modeling and simulation are valuable tools in assessing the survivability and vulnerability of complex systems to either natural, abnormal, and hostile events. However, there still remains the need to assess the accuracy of simulations by comparing computational predictions with experimental test data through the process known as validation of computational simulations. Experimental validation, however, is continually increasing in cost and time required to conduct the test. For these reasons modeling and simulation must take increasing responsibility for the safety, performance, and reliability of many high consequence systems.

Uncertainty estimation in computational simulation has been conducted in certain engineering and science disciplines for a number of years. Examples of these are nuclear reactor safety,¹⁻⁹ civil and marine engineering,¹⁰⁻¹³ and environmental engineering.¹⁴⁻¹⁸ The emphasis in these fields has been directed toward representing and propagating parameter uncertainties in mathematical models. The vast majority of this work has used probabilistic methods to represent the uncertainty and then sampling methods, such as Monte Carlo sampling, to propagate the uncertainty.

Our focus is on the estimation of total modeling and simulation uncertainty composed of variabilities, uncertainties, and errors from many sources. The issue of numerical solution error is generally ignored in uncertainty estimation. Neglecting numerical solution error is particularly detrimental to total uncertainty estimation when the mathematical models of interest are cast in terms of partial differential equations (PDEs). Types of numerical error that are of concern in the numerical solution of PDEs are: spatial discretization error in finite element and finite difference methods, temporal discretization error in time dependent simulations, and error due to discrete representation of strongly nonlinear features. It is fair to say that the field of numerical error estimation is considered to be completely separate from uncertainty estimation. Although many authors in the field of numerical error estimation refer to solution error as "numerical uncertainty," we believe this confuses the issue.

Also of concern in the present work is a type of uncertainty that occurs in complex modeling and simulation: uncertainty resulting from the model itself, which is also referred to as "model form uncertainty" or "structural uncertainty." Recent work is beginning to address the issue of competing models set in a probabilistic framework.^{19,20} We believe that nonprobabilistic mathematical representations may be more appropriate when the uncertainty derives from lack of knowledge or from errors, such as numerical solution errors. In this paper we provide definitions that distinguish variability (i.e., irreducible uncertainty), uncertainty, and error in the modeling and simulation process.

This paper proposes a new framework, or structure, of the general phases of modeling and simulation. This structure is composed of six phases, which represent a synthesis of the tasks recognized in the operations research community; the methods associated with uncertainty estimation; and the activities appropriate to the numerical solution of PDEs. It extends the recent work of Ref. 21 by addressing the nondeterministic aspects of the modeling and simulation phases. By "nondeterministic" we mean that the response of the system is not precisely predictable because of the existence of variability or uncertainty in the system or the environment. With our proposed structure one can more clearly identify where and how in the modeling and simulation process the various sources of variability, uncertainty, and error might occur, and also determine their effect on the total estimate of uncertainty. To demonstrate this new framework, we describe a system-level analysis of an unguided, rocket-boosted, aircraft-launched missile. We consider the missile to be a relatively short range, i.e., 20 n.m., air-to-ground rocket. In the conceptual modeling phase for this example, we

discuss alternative system/environment specifications, scenario abstractions, physics coupling specifications, and nondeterministic specifications. After discussing varying conceptual models, only one branch of the analysis is pursued: rigid body flight dynamics. Of the large number of possible nondeterministic phenomena, we consider only two: variability of the initial mass of the missile and uncertainty in the performance of the rocket motor because of ambiguity or non-specificity. To illustrate mathematical modeling uncertainty, we pursue two models with differing levels of physics: a six-degree-of-freedom and a three-degree-of-freedom model. Each of these models is solved through multiple simulation runs to accommodate the mass variability and the motor performance uncertainty. In each case we include the effect of error due to numerical solution of the equations of motion for each model.

2. MODELING AND SIMULATION

2.1 Review of the Literature

The operations research (OR) community has developed many of the general principles and procedures for modeling and simulation. Researchers in these fields have made significant progress in defining and categorizing the various phases of modeling and simulation.^{2,22-25} The areas of emphasis in OR include definition of the problem entity, definition of the conceptual model, assessment of data and information quality, and usage of simulation results as an aid in decision making. From a computational sciences perspective, many feel this work is extraneous because it does not deal with solving PDEs. However, we have found that the OR work is very helpful in providing a constructive philosophical approach for identifying sources of uncertainty and error, as well as developing some of the basic terminology.

In 1979 the Technical Committee on Model Credibility of the Society for Computer Simulation developed a diagram identifying the primary phases and activities of modeling and simulation.²⁶ Analysis is used to construct a conceptual model of reality; programming converts the conceptual model into a computerized model; and computer simulation simulates reality. Although simple and direct, the diagram clearly captures the relationship of two key phases of modeling and simulation to each other, and to reality. The diagram also includes the activities of model qualification, model verification, and model validation. However, the diagram does not address the detailed activities required for the solution of PDEs describing the system nor the activities necessary for uncertainty estimation.

Throughout the 1980s, improvements were made toward generalizing the concepts of modeling and

simulation.^{27,28} A significant extension of the phases of modeling and simulation was made to include the concept of the life cycle of a simulation.^{29,30} Major phases added by Nance and Balci were System and Objectives Definition, Communicative Models, and Simulation Results. The emphasis in this work was on assessing and improving the credibility of the modeling and simulation. Jacoby developed a framework with the following phases of modeling and simulation: modeling purpose, prototype, preliminary model, mathematical model, solution technique, computer program, model, and modeling result.²⁴ We have incorporated some of the concepts from this framework into the present work. The present emphasis, however, is on two additional requirements. First, we wanted to stress the *estimation* of total modeling and simulation uncertainty. This focus requires that one be able to clearly identify where sources of variability, uncertainty, and error might originate, and also how these sources will propagate through the modeling and simulation process. Second, our framework stresses systems whose mathematical formulation is given by a system of PDEs. We can then directly include the errors caused by the numerical solution of PDEs into the uncertainty estimation process.

2.2 Sources of Variability, Uncertainty, and Error

Sources of variability, uncertainty, and error are associated with each phase of modeling and simulation. Examining the literature in OR, uncertainty estimation, and numerical error estimation, one finds that most authors do not carefully distinguish between what they mean by variability, uncertainty, and error, or worse, their definitions contradict one another. Even when these terms have been defined, their definitions are typically couched in the restricted context of the particular subject, or they do not address the issue of error.^{2,31} Some authors have dealt with the distinctions between some of these sources.^{2,15,16,31-33} The present paper follows the recent work of Ref. 21 in carefully distinguishing between variability, uncertainty, and error.

We use the term *variability* to describe the *inherent* variation associated with the physical system or the environment under consideration. Sources of variability can commonly be singled out from other contributors to total modeling and simulation uncertainty by their representation as distributed quantities that can take on values in an established or known range, but for which the exact value will vary by chance from unit to unit or from time to time. Some literature refers to our definition of variability as "irreducible uncertainty." An example of a distributed quantity is the exact dimension of a manufactured part, where the manufacturing process is well understood but variable and the part has yet to be produced. Variability is generally quantified by a

probability or frequency distribution when sufficient information is available to estimate the distribution.

We define *uncertainty* as a *potential* deficiency in any phase or activity of the modeling process that is due to *lack of knowledge*. The first feature that our definition stresses is "potential," meaning that the deficiency may or may not occur. In other words, there may be no deficiency, say in the prediction of some event, even though there is a lack of knowledge if we happen to model the phenomena correctly. The second key feature of uncertainty is that its fundamental cause is incomplete information.^{15,16,34} Incomplete information can be caused by vagueness, nonspecificity, or dissonance.^{32,35} Vagueness characterizes information that is imprecisely defined, unclear, or indistinct. Vagueness is characteristic of communication by language. Nonspecificity refers to the variety of alternatives in a given situation that are all possible, i.e., not specified. The larger the number of possibilities, the larger the degree of nonspecificity. Dissonance refers to the existence of totally or partially conflicting evidence. Dissonance exists when there is evidence that an entity or elements belong to multiple sets that either do not overlap or overlap slightly. Some literature refers to our definition of uncertainty as "reducible uncertainty."

We define *error* as a *recognizable* deficiency in any phase or activity of modeling and simulation that is *not* due to lack of knowledge. Our definition stresses the feature that the deficiency is identifiable or knowable upon examination; that is, the deficiency is not caused by lack of knowledge. Essentially there is an agreed-upon approach or ideal condition that is considered to be more accurate. If divergence from the correct or more accurate approach is pointed out, the divergence is either corrected or allowed to remain. It may be allowed to remain because of practical constraints, such as the error is acceptable given the requirements, or the cost to correct it is excessive. This implies a segregation of error types: an error can be either *acknowledged* or *unacknowledged*. Examples of acknowledged errors are finite precision arithmetic in a computer, approximations made to simplify the modeling of a physical process, and conversion of PDEs into discrete equations. When acknowledged errors are introduced by the analyst into the modeling or simulation process, the analyst typically has some idea of the magnitude of such errors. Unacknowledged errors are blunders or mistakes, that is, the analyst intended to do one thing in the modeling and simulation but, for example, as a result of human error, did another. There are no straightforward methods for estimating, bounding, or ordering the contribution of unacknowledged errors. Sometimes an unacknowledged error can be detected by the person who committed it; e.g., a double-check of coding reveals that two digits have been reversed. Sometimes blunders are caused by

inadequate human interactions and can only be resolved by better communication. Redundant procedures and protocols for operations depending on a high degree of human intervention can also be effective in reducing unacknowledged errors.

2.3 Proposed Phases of Modeling and Simulation

Figure 1 depicts our representation of the phases of modeling and simulation appropriate to systems analyzed by the numerical solution of PDEs. The phases represent collections of tasks required in a large scale simulation analysis. The ordering of the phases implies an information and data flow indicating which tasks are likely to impact decisions and methodology occurring in later phases. These phases follow the recent work of Ref. 21. The paragraphs below provide brief descriptions of each of these phases. The modeling and simulation process is initiated by a question or set of questions, posed by a designer or decision maker, for which the information to address the question can be provided by exercise of a computer simulation.

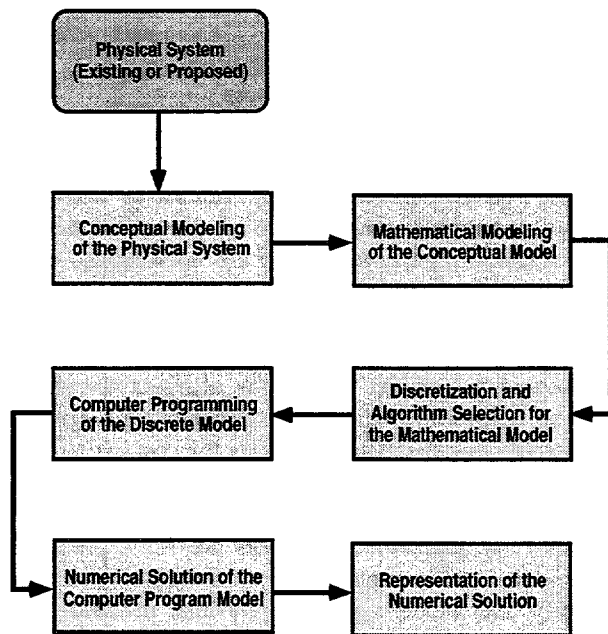


Figure 1
Proposed Phases for Computational Modeling and Simulation

Conceptual Modeling of the Physical System. Our initial phase encompasses developing a specification of the physical system and the environment. It also includes determining which physical events, or sequence of events, and which types of coupling of different physical processes will be considered. These determinations must be based on the general requirements for the modeling and

simulation effort. The physical system can be an existing system or process, or it can be a system or process that is being proposed. During the conceptual modeling phase, no mathematical equations are written, but the fundamental assumptions of the possible events and physical processes are made. Only conceptual issues are considered, with heavy emphasis placed on determining all possible factors, such as physical and human intervention, that could possibly affect the requirements set for the modeling and simulation. Identifying possible event sequences, or scenarios, is similar to developing a fault-tree structure in the probabilistic risk assessment of high consequence systems, such as in nuclear reactor safety analyses. Even if a certain sequence of events is considered extremely remote, it should still be included as a possible event sequence in the fault tree. After the system and environment is specified, options for various levels of possible physics couplings should be identified, even if it is considered unlikely that all such couplings will be considered subsequently in the analysis. If a physics coupling is not considered in this phase, it cannot be resurrected later in the process. Another task conducted in this phase of the analysis is the identification of all of the system and environment characteristics that might be treated nondeterministically. Consideration is given as to whether these characteristics are to be treated as fixed, stochastic, or unknown. However, details concerning their representation and propagation are deferred until later phases.

Mathematical Modeling of the Conceptual Model.

The primary task in this phase is to develop precise mathematical models, i.e., analytical, statements of the problem (or series of event-tree-driven problems) to be solved. Any complex mathematical model of a problem, or physical system, is actually composed of many mathematical submodels. The complexity of the models depends on the physical complexity of each phenomenon being considered, the number of physical phenomena considered, and the level of coupling of different types of physics. The mathematical models formulated in this phase include the complete specification of all of the PDEs, auxiliary conditions, boundary conditions, and initial conditions for the system. For example, if the problem being addressed is a fluid-structure interaction, then all of the coupled fluid-structures PDEs must be specified, along with any fluid or material-property changes that might occur as a result of their interaction.

Another function addressed during this phase of analysis is selecting appropriate representations and models for the nondeterministic elements of the problem. Several considerations might drive these selections. Restrictions set forth in the conceptual modeling phase of the analyses may put constraints on the range of values or types of models that might be used further in

the analysis. Within these constraints the quantity and/or limitations of available or obtainable data will play an important role. A probabilistic treatment of nondeterministic variables generally requires that probability distributions can be established, either through data analysis or through subjective judgments. In the absence of data, qualified "expert opinion" or similar type information from other sources regarding the relative likelihoods may be incorporated. If there is a significant lack of information, it is possible that only bounding or set representations may be appropriate for uncertainties.

Discretization and Algorithm Selection for the Mathematical Model. This phase accomplishes two tasks related to converting the mathematical models into a form that can be addressed through computational analysis. The first task involves conversion of the PDE form of the mathematical model into a discrete, or numerical, model. Simply stated, the mathematics are translated from a calculus problem to an arithmetic problem. In the discretization phase all of the spatial and temporal differencing methods, discretized boundary conditions, discretized geometric boundaries, and grid generation methods are specified in analytical form. In other words, algorithms and methods are prescribed in mathematically discrete form, but the spatial and temporal step sizes are not specified. The discretization phase focuses on the conversion from continuum mathematics to discrete mathematics, not on numerical solution issues. We strongly believe that the continuum mathematical model and the discrete model should be separately represented in the phases of modeling and simulation.³⁶ The discretization phase deals with questions such as consistency of the discrete equations with the PDEs, stability of the numerical method, approximations of mathematical singularities, and differences in zones of influence between the continuum and discrete systems.

The second task of this phase of the analysis is the specification of the methodology that will dictate computer runs to be performed in a later phase of the analysis in order to accommodate the nondeterministic aspects of the problem. For example, a Monte Carlo method or response surface method could be chosen for propagating variabilities. Nondeterministic sources include system parameters, boundary conditions, and initial conditions that may vary randomly from component-to-component and/or system-to-system. Modeling too can be nondeterministic in nature when alternative models are constructed to address the same aspects of the problem. Presumably only one model is correct (or more correct) for the task, but this is not generally known beforehand, i.e., in a prediction. In addition, a preferred model may be too expensive to be used exclusively in the analysis and, as a result, less accurate models would be used in portions of the

analysis.

Computer Programming of the Discrete Model. This phase is common to all computer modeling: algorithms and solution procedures defined in the previous phase are converted into a computer code. The computer programming phase has probably achieved the highest level of maturity because of decades of programming development and software quality assurance efforts.^{37,38} These efforts have made a significant impact in areas such as commercial graphics, mathematics, and accounting software, telephone circuit-switching software, and flight control systems. On the other hand, these efforts have had little impact on corporate and university-developed software for computational fluid dynamics, solid dynamics, and heat transfer simulations, as well as most applications of massively parallel processing.

Numerical Solution of the Computer Program Model. In this phase the individual numerical solutions are actually computed. No quantities are left arithmetically undefined or continuous; only discrete parameters and discrete solutions exist with finite precision. For example, a spatial grid distribution and a time step is specified; space and time exist only at discrete points, although these points may be altered during subsequent computer runs.

Multiple computational solutions are normally required for nondeterministic analyses. These multiple solutions are dictated by the propagation methods and input settings determined in the discretization and algorithm selection phase. Multiple solutions can also be required from the mathematical modeling phase if alternative models are to be investigated. For some propagation methods the number and complete specification of subsequent runs is dependent on the computed results. When this is the case, these determinations are made as part of this phase of the analysis.

Representation of the Numerical Solution. The final phase of the modeling and simulation process concerns the representation and interpretation of both the individual and collective computational solutions. The collective results are ultimately used by decision makers or policy makers, whereas the individual results are typically used by engineers, physicists, and numerical analysts. Each of these audiences have very different interests and requirements. The individual solutions provide detailed information on deterministic issues such as the physics occurring in the system, the adequacy of the numerical methods to compute an accurate solution to the PDEs, and the system's response to the deterministic boundary and initial conditions. For the individual solutions the primary task is the construction of continuous functions based on the discrete solutions obtained in the previous phase. Here the continuum

mathematics formulated in the mathematical modeling phase is approximately reconstructed based on the discrete solution.

The collective solutions provide information on the nondeterministic response of the system. For the collective solutions the primary task is the assimilation of individual results to produce summary data, statistics, and graphics portraying the nondeterministic features of the system. These results are utilized to assess the simulation results from a high-level perspective and compare them to requirements of the analysis.

3. MISSILE FLIGHT ANALYSIS EXAMPLE

3.1 Description of the Problem

The proposed new framework for conducting modeling and computational simulation is applicable to an extremely wide range of complex engineering systems. To demonstrate how this framework can be useful for large-scale, real world system analyses, we apply it to an example problem. We consider an analysis of the flight of a rocket-boosted, aircraft-launched missile. We make the following assumptions concerning the missile:

- 1) The missile is unguided during its entire flight, i.e., only ballistic flight is considered.
- 2) The missile is propelled by a solid fuel rocket motor for the initial portion of its flight, and it is unpowered during the remainder of the flight.
- 3) The missile is fired from a launch rail attached to the aircraft in flight.
- 4) The only aerodynamic surfaces on the missile are fins to provide flight stability.

The analysis considers the missile flight to be in the unspecified future, i.e., the analysis is an attempt to predict future plausible events, not analyze an event in the past.

Figure 2 illustrates the activities that are conducted in each of the six phases of modeling and simulation. Also shown for each activity are the dominant sources of variability, uncertainty, and error that typically occur in each activity. Figure 2 extends the ideas presented in Ref. 21 by stressing the nondeterministic features of modeling and simulation. For the remainder of the paper we will refer to the aggregation and propagation of variability, uncertainty, and error simply as total uncertainty. We now discuss in detail the activities that are conducted in each of the phases and how these activities are applied to the missile flight example problem.

3.2 Conceptual Modeling Activities

As seen in Fig. 2, we have identified four major activities that are conducted in the conceptual modeling

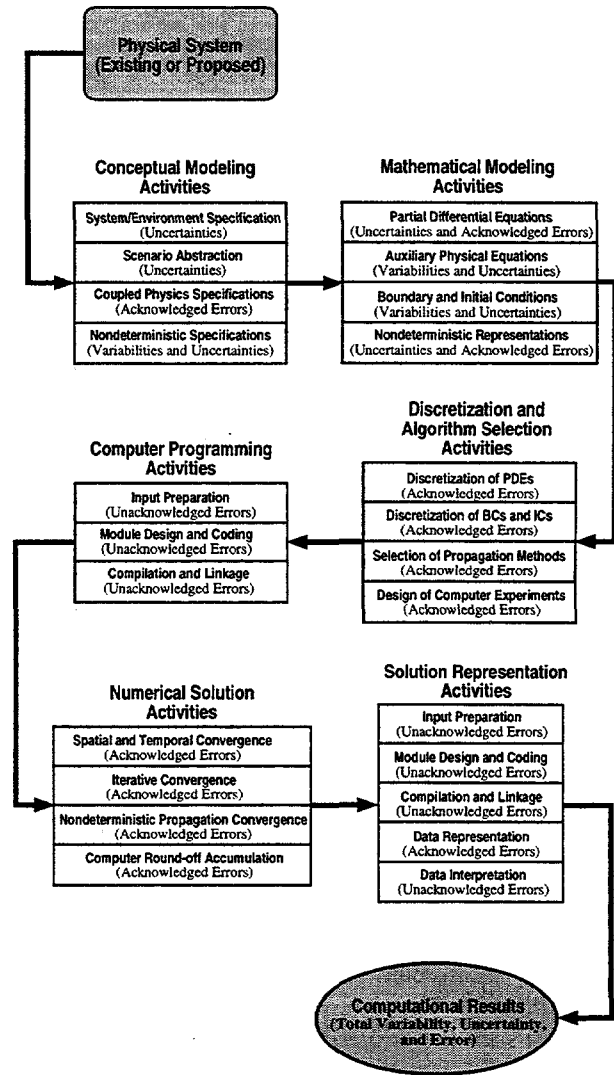


Figure 2
Activities Conducted in the Phases of Computational Modeling and Simulation

phase: system/environment specification, scenario abstraction, coupled physics specification, and nondeterministic specifications. The system/environment specification activity consists primarily of careful specification of what physical or conceptual elements are considered as part of the system and what is considered part of the environment. When we say physical or conceptual elements are part of the system we mean that it is possible that any of the elements can interact with one another. This concept is similar to a system as defined in thermodynamics. The state of a system is influenced by processes internal to the system, i. e., endogenous processes, and also processes or activities external to the system, i. e., exogenous effects. Exogenous processes or activities are considered to be part of the environment. A system is influenced by the

environment, but the environment cannot be influenced by the system.³⁹ In other words, the system and the environment do not interact; the system can respond to the environment, but the environment cannot respond to the system. System/environment specifications are a matter of engineering judgement and are not unique. As a result, they are one of the most difficult conceptual issues in modeling and simulation.

Figure 3 shows three possible system/environment specifications for the example problem. They are listed from the most inclusive to least inclusive. System/environment Specification 1 considers the missile and the atmosphere near the missile to be part of the system, whereas the launching aircraft and target are considered part of the environment. An example of an analysis that would be allowed with this specification is the flow field of the missile and the rocket exhaust coupled to flow field of the launching aircraft, i.e., the missile and rocket exhaust could interact with the aircraft flow field. Another example allowed by this system/environment specification would be the analysis of the missile flight inside an enclosure or tunnel, e.g., near the target. System/environment Specification 2 considers the missile and the aerothermal processes occurring on the missile to be part of the system, whereas the atmosphere near the missile, the launching aircraft, and the target are considered part of the environment. This specification allows analyses that couple the missile and its aerothermal effects. For example, one could consider the deformation of the missile due to aerodynamic loading and thermal heating, and then couple these deformations into recomputing the aerodynamic loading and thermal heating. System/environment Specification 3 considers the missile to be the system, whereas the aerothermal processes, the atmosphere near the missile, the launching aircraft, and the target are considered part of the environment. Even though this is the simplest specification considered, it still allows for significant complexities in the analysis.

Scenario abstraction, Fig. 2, attempts to identify all possible physical events, or sequences of events, that may affect the goals of the analysis. For relatively simple systems, isolated systems, or systems with very controlled environments or operational conditions, scenario abstraction can be straightforward. Complex engineered systems, however, can be exposed to a variety of natural and abnormal operating conditions, hostile environments, or a myriad of human-caused or accidentally caused failure modes. Scenario abstraction for these complex systems is a mammoth undertaking. The field of engineering that has achieved the highest development of scenario abstraction is probabilistic risk assessment (PRA) of nuclear power plants. PRA techniques construct a many-branched event tree for

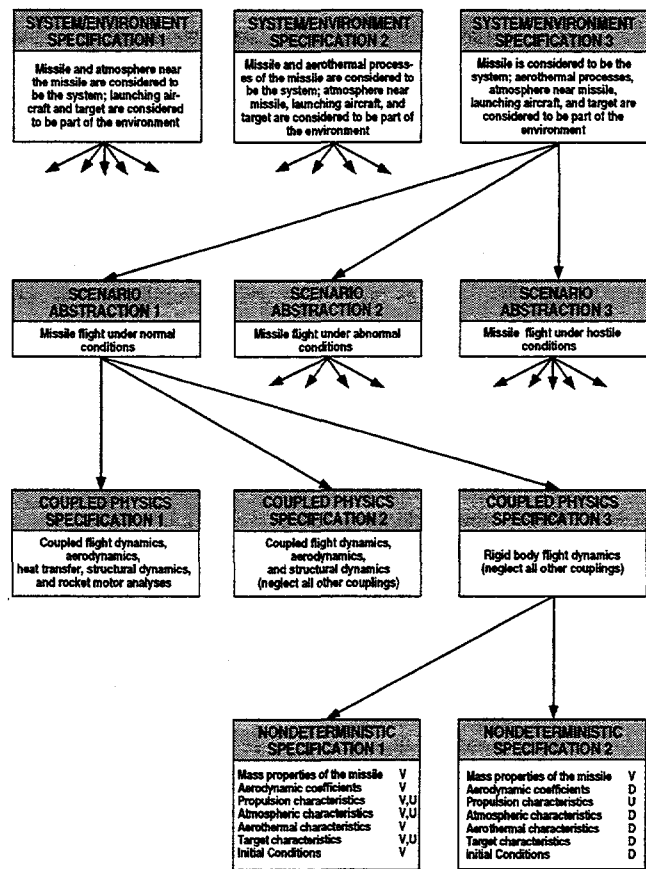


Figure 3
Conceptual Modeling Activities for the Missile Flight Example

complex operating and failure scenarios. Even though the probability of occurrence of certain events may be extremely low, these events must be considered and analyzed for failure of nuclear power plants and other high consequence systems. The scenario abstraction considered here includes both event tree construction and decision tree construction, that is, decision tree construction does not necessarily depend on events, but it can identify possible results based on decisions or analyses that could be pursued.

As shown in Fig. 3, the missile flight example identifies three broad classes of scenarios; missile flight under normal, abnormal, and hostile conditions. By normal conditions we mean missile flight under reasonably expected conditions, such as typical launch conditions from aircraft that are expected to carry the missile, near nominal operation of the propulsion system, and reasonably expected weather conditions. Examples of flight under abnormal conditions would be

improperly assembled missile components; explosive failure of the propulsion system; and flight through adverse weather conditions, such as snow or lightning. Examples of flight under hostile conditions would be detonation of near by explosive systems, damage to missile components due to small arms fire, and damage from laser or microwave defensive systems. The three scenario categories considered here have been commonly used for military systems, e.g., nuclear weapons. With the increasing concern of terrorist attacks on civilian systems, such as buildings, commercial aircraft, bridges, and dams, this categorization may, unfortunately, prove to be more broadly useful in the future.

Coupled physics specification, Fig. 2, clarifies and carefully distinguishes the possible alternatives for physics (and chemistry) coupling for the system/environment specification and scenario abstraction under consideration. A clear statement of the possible levels of physics coupling is required because of the wide variety of physics that may occur in a complex system. In the missile flight example, Fig. 3, we identify three levels of physics coupling, although more alternatives could be identified. Coupled physics Specification 1 couples essentially all of the physics that could exist in the problem. For example, this specification could couple the structural deformation and dynamics with the aerodynamic loading and thermal loading due to atmospheric heating. It could also couple the deformation of the solid fuel rocket motor case due to combustion pressurization, the heat transfer from the motor case into the missile airframe structure, and the non-rigid body flight dynamics on the missile. Coupled physics Specification 2 couples the missile flight dynamics, aerodynamics, and structural dynamics, neglecting all other couplings. This coupling permits the computation of the deformation of the missile structure due to inertial loading and aerodynamic loading and then the aerodynamic loading and aerodynamic damping due to the deformed structure. This would be a coupled, time dependent, fluids/structures interaction simulation. Coupled physics Specification 3 assumes a rigid missile body; not only is physics coupling disallowed, but the missile structure is assumed rigid. The missile only is allowed to respond to inputs or forcing functions from the environment. Structural dynamics is removed from the analysis; i.e., only rigid body dynamics is considered.

Before we consider the last activity, nondeterministic specification, consider the possible sources of total uncertainty that could occur in the three activities discussed so far. System/environment specification and scenario abstraction activities introduce uncertainties into the modeling and simulation process. This occurs primarily because of what is *not* included, or scenarios *not* imagined. The wider the scope of the analysis or the

more complex the system, the more possibilities there are for uncertainties due to lack of knowledge about aspects of the modeled system and environment. Indeed, an underlying weakness of modern technological analysis is events, effects, and possibilities not initially considered.⁴⁰ For example, automatic control systems designed to ensure safe operation of complex systems during normal operation or during maintenance of the system can fail, or the safety system can be overridden. During coupled physics specification, the primary source of total uncertainty introduced into the analysis is acknowledged error. A hierarchal ordering of levels of physical coupling in conceptual models can commonly be constructed. Based on experience with similar systems, previous analyses, risk of adverse safety, performance, economic consequences, and budget and schedule considerations, decisions are then made concerning which physics coupling is chosen. However, when physics couplings are neglected, an acknowledged error is introduced.

In the nondeterministic specification activity, Fig. 2, decisions are made concerning what aspects of the system and environment will be considered deterministic or nondeterministic. By a deterministic system and environment we mean one which exhibits one system response given a specification of all mathematical models and parameters of the system. Certain nonlinear systems, even when all models and parameters are specified, can yield multiple responses or even chaotic responses. We include these types of systems in the class of nondeterministic systems, although they are not addressed in the present work. The predominant cause of nondeterministic system response is inherent randomness in model parameters, initial conditions, or parameters specifying the environment. We have referred to these types of sources producing nondeterministic features as variabilities. Uncertainties in model parameters, initial conditions, or the environment also occur because of a lack of knowledge about these factors. The uncertainty can often be reduced by obtaining relevant data or through actions taken that limit the state of key elements of the system or the environment. Often these are policy or procedural decisions. Examples of these cases are: Was the system incorrectly manufactured or assembled? Has inadequate or improperly conducted system maintenance significantly altered certain system characteristics? Was the system damaged in the past, but the damage is unknown? In other situations it may not be possible to reduce the lack of knowledge, and reduce the uncertainty.

For the missile flight example we list only two alternative nondeterministic specifications shown in Fig. 3. Nondeterministic Specification 1 includes the following variabilities (indicated by a V in Fig. 3): mass properties of the missile, aerodynamic force and moment coefficients, aerothermal heating characteristics, and

initial conditions at missile launch. These are considered variabilities because they are usually associated with random variation due to manufacturing processes or physically random processes. If a large number of missiles are manufactured, for example, sufficient inspection data would normally exist so that a representative probability distribution for each parameter could be constructed. Nondeterministic Specification 1 also includes the following as variabilities and/or uncertainties (indicated by a V,U in Fig. 3): propulsion characteristics, atmospheric characteristics, and target characteristics. These quantities could be considered as variabilities, but their nondeterministic feature is usually dominated by lack of knowledge. For example, propulsion characteristics of solid rocket motors can vary substantially with age and temperature of the propellant. Suppose that statistical models which incorporate age and temperature of the propellant have been constructed. If the age of the propellant in a particular motor is not known or the temperature of the propellant is not known, a statistical model is of little value in estimating the variation in the performance of the motor. A similar argument can be made for estimating the uncertainty in atmospheric characteristics, for example, wind conditions. Without specifying additional knowledge, such as location on earth, month of the year, or even time of the day, statistical models are of limited value.

For nondeterministic Specification 2, Fig. 3, we chose one parameter of the mass properties of the missile as a variability, and one characteristic of the propulsion system as an uncertainty. We pursue Specification 2 in the example problem to distinguish the characteristics of each and to show how they might be represented differently in a computational simulation. All other parameters are considered deterministic (indicated by a D in Fig. 3).

3.3 Mathematical Modeling Activities

As shown in Fig. 2, we have identified four major activities in the mathematical modeling phase: formulation of the partial differential equations, choice of all the auxiliary equations that supplement the differential equations, formulation of all the initial and boundary conditions required to solve the PDEs, and selection of the mathematical representation of nondeterministic elements of the analysis. The PDEs commonly represent conservation equations for mass, momentum, and energy, but they can originate from any mathematical model of the system. The auxiliary equations are equations which are required to complete the PDEs. Examples of these would be turbulence modeling equations in fluid dynamics, equations of state in hydrodynamics, material constitutive equations in solid dynamics, and neutron cross-sections in neutron transport. The auxiliary equations can be of any type, e.g., algebraic equations,

integral equations, or PDEs. The boundary and initial conditions provide the required closure equations needed for all PDEs.

Formulation of the nondeterministic representations is based on the needs of the analysis together with the quantity and quality of relevant information available. When the nondeterministic specification indicates that a range of values are of interest for a parameter or characteristic of the analysis, it may or may not include constraints on this range. Within these constraints an appropriate representation for the nondeterministic element will depend on available and/or obtainable relevant information. Probabilistic models (distributions or frequency functions) are appropriate only when enough information is available to determine relative likelihoods of different values over a range. In the absence of this information, one attempts to produce bounding values, or may hypothesize distributions for these elements and perform further analysis conditional on these bounding values or hypothesized distributions.

Variabilities commonly dominate the nondeterministic features of the auxiliary physical equations and boundary and initial condition activities. The most common variabilities are those due to inherent randomness of continuous parameters in these equations. Variabilities are nearly always represented by probability distributions. In some cases the form of these distributions are inferred from first principles of the processes involved in determining the parameter values. In most cases the distributions are chosen based on convenience. Parameters associated with the probability distributions are then estimated when sufficient data are available or assigned values based on a subjective assessment.

Uncertainties can have a large impact on the nondeterministic formulation of the PDEs because the key issue can be limited, or inadequate, knowledge of the physical processes involved. Examples of uncertainties that occur in the PDEs are limited knowledge of the equations for turbulent reacting flow, conflicting models for crack propagation in materials, and competing models for elastic deformation of composite materials. For physical processes that are relatively well understood, deficiencies in certain models should be considered as errors rather than uncertainties. This is based on the argument that if sufficient knowledge of the process exists, a set of alternative models can be ordered in terms of increasing accuracy. For example, in the modeling of fluid dynamic turbulence, the models can be ordered in terms of increasing accuracy as follows: algebraic models, two-equation models, Reynolds stress models, and large eddy simulation models. In general, this ordering is appropriate, but for individual flow fields there is no guarantee that any one model will be more

accurate than any other because certain lower order models can be very accurate for specific cases.

Acknowledged errors in PDE models are those due to mathematically representing the physics in a more simplified or approximate form than the best available. It is invariably the case that for any mathematical model chosen to represent some physical process, one can identify higher fidelity models that are known to exist. In our definitions given in Section 2.2, this is precisely what is meant by acknowledged error. Higher fidelity models are usually not chosen because of the higher computational costs associated with their solution. The ratio of computational cost for a higher fidelity model to a lower fidelity model is commonly high, sometimes exceeding a factor of a 100. Analysts ordinarily choose a given level of model fidelity based on practical issues, such as computational resources available, options in computer codes they are familiar with, and schedule constraints, as well as technical issues. Some examples of acknowledged errors in mathematical modeling are the modeling of a process in two spatial dimensions when three spatial dimensions may be needed; the assumption of a steady state when unsteady effects may be important; and the assumption of homogenous material properties when mesoscale features play a substantial part. These examples of acknowledged errors are all characteristic of situations in which physical modeling approximations were made to simplify the mathematical model and the subsequent solution.

For the missile flight example, two mathematical models are chosen; a six-degree-of-freedom (6-DOF) model and a three-degree-of-freedom (3-DOF) model, Fig. 4. Both of these models are consistent with the conceptual model being analyzed: system/environment Specification 3, scenario Specification 1, coupled physics Specification 3, and nondeterministic Specification 2 (Fig. 3). The translational equations of motion can be written as

$$m \frac{d\vec{V}}{dt} = \sum \vec{F}, \quad (1)$$

where m is the mass of the vehicle, \vec{V} is the velocity, and $\sum \vec{F}$ is the sum of all forces acting on the vehicle. The rotational equations of motion can be written as

$$[I] \frac{d\vec{\omega}}{dt} = \sum \vec{M} + \vec{\omega} \times \{ [I] \cdot \vec{\omega} \}, \quad (2)$$

where $[I]$ is the the inertia tensor of the vehicle, $\vec{\omega}$ is the angular velocity, and $\sum \vec{M}$ is the sum of all moments acting on the vehicle. Eq. (1) represents the 3-DOF equations of motion, and Eqs. (1) and (2) represent the 6-DOF equations of motion. Although these are ordinary differential equation models instead of the PDE

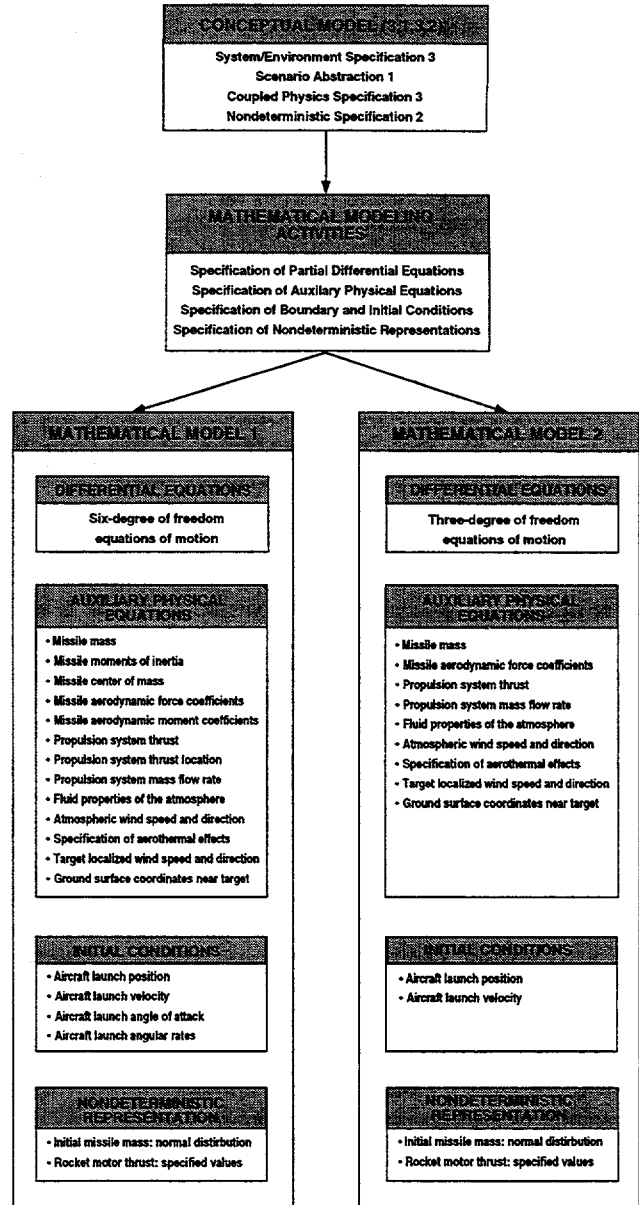


Figure 4
Mathematical Models for the Missile Flight Example

models stressed in the present work, key aspects of the present framework can still be exercised.

For these two mathematical models of flight dynamics, one can unequivocally order the models in terms of fidelity. Indeed, the physics and mathematics of the 6-DOF equations are so well understood that there is no need for experimental validation of these models. Their accuracy is only limited by the accuracy of the assumption of a rigid body, accuracy of the measured mass properties, and accuracy of the forces and moments acting on the vehicle. However, as mentioned above, for models of complex physical processes or systems this

ordering is commonly not possible.

Figure 4 lists all of the auxiliary equations and initial conditions that are needed for each mathematical model. As is to be expected of higher fidelity models, the 6-DOF model requires physical information beyond that which is required by the 3-DOF model. This poses the question: When does the lack of information of the additional needed parameters in a higher fidelity model counteract its accuracy when compared to a lower fidelity model? Although this question is not addressed in the present work, it is an issue that must be weighed in many analyses. It is fallacious to claim that the higher the fidelity of the physics model, the better the results. Uncertainty of parameters and greater computer resources needed to solve higher fidelity models are critical factors in total uncertainty estimation. In addition, constraints on computer resources can obviate the usefulness of a higher quality model.

Two nondeterministic parameters will be considered in the missile flight example: the initial mass of the missile and the propulsion thrust characteristics. Both parameters appear in each of the mathematical models chosen so that direct comparisons of their effect on each model can be made. It is assumed that sufficient inspection data of manufactured missiles is available for the missile mass to justify a normal distribution with known mean and standard deviation. Thrust characteristics are considered to be an uncertainty that derives from nonspecificity, i.e., multiple situations are possible.^{32,35} We assume the nonspecificity is due to the temperature of the solid propellant. We choose a nominal value and two bounding values: normal operating temperature, highest allowed temperature within the manufacturer's specification, and the lowest allowed temperature. The high temperature condition causes the thrust to be higher and the burn time to be shorter, and the low temperature condition causes the thrust to be lower and the burn time to be longer. We assume that the thrust versus time profiles of the high and low temperature motors are accurately known from experimental data. It is clear that the uncertainty in propulsion thrust can be steadily reduced as information is added to the analysis. For example, if the temperature at launch could be specified within some experimental measurement uncertainty, then the propulsion uncertainty could be greatly reduced or eliminated. If age of the specific motor to be fired were known, then the uncertainty in performance due to age of the propellant could be eliminated. With this combined level of information, one could characterize the propulsion characteristics as a variability.

3.4 Discretization and Algorithm Selection Activities

The discretization and algorithm selection phase

accomplishes two related tasks. First, it converts the continuum mathematics model, i.e., the differential equations, into a discrete mathematics problem suitable for numerical solution. Second, it provides the methodology that will be used to determine how a discrete set of computer solutions can be most appropriately used to accommodate the nondeterministic features of the analysis. The conversion from continuous to discrete mathematics is fundamentally a mathematics-approximation topic; errors and not uncertainties are the dominant loss-of-confidence issue in this phase. (Note that for the remainder of the paper when we refer to "errors," we will only be referring to *acknowledged* errors, unless otherwise stated.) Some may question why this conversion process should be separated from the solution process. We argue that this conversion process is the root cause of more difficulties in the numerical solution of nonlinear PDEs than is generally realized.^{41,42} When traditional nondeterministic methods are applied to systems described by differential equations, then one is dealing with stochastic differential equations. The discrete solution to these type equations, however, is much less developed than for deterministic differential equations.⁴³

As shown in Fig. 2, we identify four activities in the discretization and algorithm selection phase: discretization of the PDEs, discretization of the boundary conditions and initial conditions, selection of the propagation methods, and design of computer experiments. The types of errors that should be identified in the discretization of the PDEs, BCs, and ICs are those associated with possible inconsistencies between the discrete form of the equations in the limit and the continuum form of the equations. This normally is evaluated by analytically proving that the numerical algorithm approaches the continuum equations as the discretization size approaches zero. For simple differencing methods, this is straightforward. For complex differencing schemes, such as essentially non-oscillatory schemes, flux limiter schemes, and second-order, multidimensional upwind schemes, determining the consistency of the schemes can be difficult. For complex multiphysics in coupled PDEs, it is impossible to prove. Related issues are also treated in the discretization activities of differential equations, such as: Are the conservation laws satisfied for finite spatial grid sizes, or are mass, momentum, and energy only conserved in the limit? Does the numerical damping approach zero as the mesh size approaches zero? Note that discretization of PDEs are also involved in the conversion of Neumann and Robin's, i.e., derivative, boundary conditions into discrete equations. We have included the conversion of continuum initial conditions to discrete initial conditions not because there are derivatives involved, but because spatial singularities may be part of the initial conditions. An example is the

time dependent decay of a vortex for which the initial condition is given as a singularity. Our point is also valid, indeed much more common, when singularities or discontinuities are specified as part of the boundary conditions.

The selection of propagation methods and design of computer experiments in Fig. 2 both address the conversion of the nondeterministic elements of the analysis into multiple runs, or solutions, of a deterministic computational simulation code. Selection of a propagation method involves the determination of an approach, or approaches, to propagating variabilities and uncertainties through the computational phases of the analysis. Examples of methods for propagating variabilities include: reliability methods,⁴⁴ sampling methods such as Monte Carlo or Latin Hypercube,^{45,46} or statistical design approaches.⁴⁷ Methods for the propagation of uncertainties defined using non-probabilistic representations, e.g., possibility theory and fuzzy sets, are a subject of current research.⁴⁸⁻⁵¹ The design of computer experiments task performed as a part of this phase is driven to a large extent by the availability of resources and by the requirements of the analysis. Establishing an experimental design often involves more than just implementation of the propagation method specified above. The problems associated with large analyses can often be decomposed in a way that permits some variables and parameters to be investigated using only portions of the code or, perhaps, simpler models than are required for others. This decomposition of the problem and selection of appropriate models, together with the formal determination of inputs for the computer runs, can have a major effect on the uncertainty introduced into the analysis in this phase. This activity is performed here because this detailed specification of inputs and models will impact programming requirements, as well as the running of the computer model in the numerical solution phase. These tasks may be performed differently for different mathematical models and may involve the specification of probabilities associated with different model choices.

For the missile flight example, the same discretization method was applied to both 6-DOF and the 3-DOF mathematical models. This resulted in two discretized models, but they only differ in the differential equations being solved. A Runge-Kutta-Fehlberg 4(5) method was chosen to solve each system of ODEs.⁵² The RKF method is fifth order accurate at each time step, and the integrator coefficients of Cash and Karp were used.⁵⁸ The method provides an estimate of the local truncation error, i.e., at each step, so that the estimated numerical solution error can be directly controlled by adjusting the step size as the solution progresses. The local truncation error is computed by comparing a fourth

order accurate solution with the fifth order accurate solution. The method chosen for propagation of variability was the Latin Hypercube Sampling (LHS) method. LHS is a random sampling method for choosing discrete values from a probabilistically defined nondeterministic variable or parameter, and often provides an advantage in efficiency over strict Monte Carlo sampling. For propagation of the uncertainty, we simply chose three possible propulsion characteristics to bound the solution and provide a nominal result. The experimental design task for this example is simple because one of our objectives is to compare models of different fidelity. Hence, the experimental design calls for performing the same number of Latin Hypercube calculations for both the 3-DOF and 6-DOF models. In an actual analysis this phase would include selecting how to mix computer runs between the 3-DOF and 6-DOF models and determination of how results from both models might be combined to maximize the value of the computations. This maximization process is a research topic of major importance for complex systems.

3.5 Computer Programming Activities

Figure 2 identifies three activities in the computer programming phase: input preparation, module design and coding, and compilation and linkage. Input preparation refers to the analyst's conversion of the mathematical and discrete model elements into equivalent data elements usable by the application code. The second and third activities relate to the building of the application code itself. Here subroutine modules are designed and implemented through a high-level programming language. This high-level code is then compiled into object code and linked to the operating system and libraries of additional object code to produce the final executable code.

The correctness of the computer programming phase is most influenced by unacknowledged errors, i.e., mistakes. The potential for mistakes in all three of these activities is enormous. In addition to the most obvious programming bugs (which still occur frequently, despite being obvious), there is the more subtle problem of undefined code behavior. This occurs when a particular code syntax is undefined within the programming language, leading to executable code whose behavior is compiler-dependent. Compilation and linkage introduce the potential for further errors unbeknownst to the developer. Primary among these are bugs and errors in the numerous libraries of object code linked to the application. These libraries can range from the ubiquitous, such as trigonometric functions, to matrix inversion and the solution of special classes of ODEs and PDEs. Such libraries allow the developer to reuse previously developed data handling and numerical analysis algorithms. Unfortunately, the developer also

inherits the undiscovered or undocumented errors in these libraries. There is also the possibility that the developer misunderstands or makes an error in the values passed to the library routines.

The recognition and elimination of programming errors, although not generating much excitement with computational physicists, remains a major cost factor in producing highly verified software. Even with the maturity of the software quality assurance methods, assessing software quality is becoming more difficult because of massively parallel computers. In our opinion, the complexities of optimizing compilers for these machines, of message-passing and of memory-sharing, are increasing faster than the capabilities of software quality assessment tools. As a case in point, debugging computer codes on massively parallel computers is moving toward becoming a nondeterministic process. That is, the code does not execute identically from one run to another because of other jobs executing simultaneously on the massively parallel machine. It is still a fundamental theorem of programming that the correctness of a computer code and its input cannot be proven except for trivial problems.

The computer code that was used for the missile flight example was the TAOS code.⁵⁴ This is a general-purpose flight dynamics code that can be used for a wide variety of guidance, control, and optimization problems for flight vehicles. We used only the ballistic flight option to solve both the 6-DOF and 3-DOF equations of motion. Concerns with coding, compilation, and linkage on massively parallel computers were not a factor in this example problem because program execution was performed only on Unix workstations.

3.6 Numerical Solution Activities

As shown in Fig. 2, we have identified four activities occurring in the numerical solution phase: spatial and temporal convergence, iterative convergence, nondeterministic propagation convergence, and computer round-off accumulation. Spatial and temporal convergence addresses the accuracy of numerical solutions using finite spatial grids and finite time steps. These two can be grouped into the general category of truncation error due to the discrete solution of PDEs. By iterative convergence we mean the finite accuracy to which nonlinear algebraic, or transcendental, discrete equations are solved. Iterative convergence error normally occurs in two different procedures of the numerical solution: 1) during the iterative convergence which must be achieved within a time step and 2) during the global iterative convergence of an elliptic PDE, i.e., a boundary value problem. Examples of the iterative convergence which must be achieved during a time step are: intra-time step iteration to solve the unsteady heat conduction equation

when the thermal conductivity depends on temperature, and the iterative solution of nonlinear constitutive equations. Iterative convergence error is different from error caused by finite precision arithmetic, i.e., round-off error.

Nondeterministic propagation convergence refers to activities related to adjustments in, or further specification of, inputs determining specifics of the multiple deterministic computer runs. Some methods for uncertainty propagation and experimental design rely on run-time results to help direct further computer experimentation. Reliability methods, for example, focus on finding a specific point (for functional expansion) that provides a 'best approximation' to system performance. Convergence to this point is determined by the change in the movement of the approximation to this point from one computer run to the next. It is clear that the nondeterministic propagation convergence error, as well as those discussed in the previous paragraph, are all acknowledged errors.

For the flight dynamics example, the numerical solution method used a variable time step so that the local truncation error could be directly controlled at each step. The local truncation error is estimated at each step for each state variable for each system of differential equations. For the 6-DOF model there are 12 state variables, and for the 3-DOF model there are 6 state variables. Before a new time step can be accepted in the numerical solution, a relative error criterion must be met for each state variable. If the largest local truncation error of all the state variables is less than 0.6 of the error criterion, then the step size is increased. Quantification of local solution error is important not only to measure its impact on an individual solution, but also to precisely determine its interaction with the variability and uncertainty in the problem. In the solution of PDEs for complex systems, general procedures for estimating solution error are very difficult to develop and compute. Global estimates of a posteriori solution error are commonly made with finite element methods, but local error estimates are not available. For finite difference and finite volume methods Richardson's method can be used to estimate local truncation error, but this becomes quite computationally expensive for complex problems.

3.7 Solution Representation Activities

In the solution representation phase shown in Fig. 2, we have identified five activities: input preparation, module design and coding, compilation and linkage, data representation, and data interpretation. The first three activities are very similar to those discussed in the computer programming phase. The data representation task includes two types of similar activities: first, the representation of individual solutions over the independent variables of the PDEs and, second, a

summary representation that combines elements of the multiple individual deterministic computer runs. Representation of individual solutions refers to the construction of a continuum solution based on the numerical solution at discrete points in space and time. Data representation errors originate as a result of the inaccurate or inappropriate construction of continuous functions from the discrete solution of the PDEs in the post processor. Examples are oscillations of the continuous function between discrete solution points due to the use of a high-order polynomial function in the post processor and interpolation of the discrete solution between multiblock grids such that conservation of mass, momentum, and energy are not conserved. Note that we mean inaccurate construction with respect to the discrete solution, *not* with respect to the continuum PDEs. To clarify this point, consider the numerical solution of a shock wave passing through a fluid or a solid and the shock wave is physically modeled as a discontinuity in the continuum PDEs. If the discretization method approximates the discontinuity with a continuous function, e.g., a shock capturing method, then in the discrete representation the shock wave is no longer discontinuous. As a result, the construction error should be judged with respect to the continuous function approximation of the discrete solution; the discontinuity was lost in the discretization and it cannot be recovered here.

Representation of a nondeterministic simulation from the individual deterministic computer runs refers to the compilation of these multiple solutions into statistical or probabilistic measures that can be used to address the requirements of the analysis. This can include developing summary descriptions of the solution and discriminating which parts of the represented solutions will be reported through tables and figures. Errors can occur in the representation of a nondeterministic solution as a result of integrating the ensemble of individual solutions in a way which is inconsistent with the specified propagation method. Data representation errors are principally acknowledged errors in that a correct or consistent discrete-to-continuum mapping is known from the choice of discretization methods.

The data interpretation activity refers to the human perceptions or impressions that are formed based on the observation of the represented solutions. If the perceptions or impressions are correct, then knowledge or understanding is generated. If they are incorrect, then an unacknowledged error has occurred. In other words, data interpretation errors occur when a user incorrectly interprets the numerical solutions. Examples of interpretation errors are: concluding that a computed solution is chaotic when it is not, or interpreting a computed flow as turbulent when it is only a spurious numerical solution. Importantly our definition of data

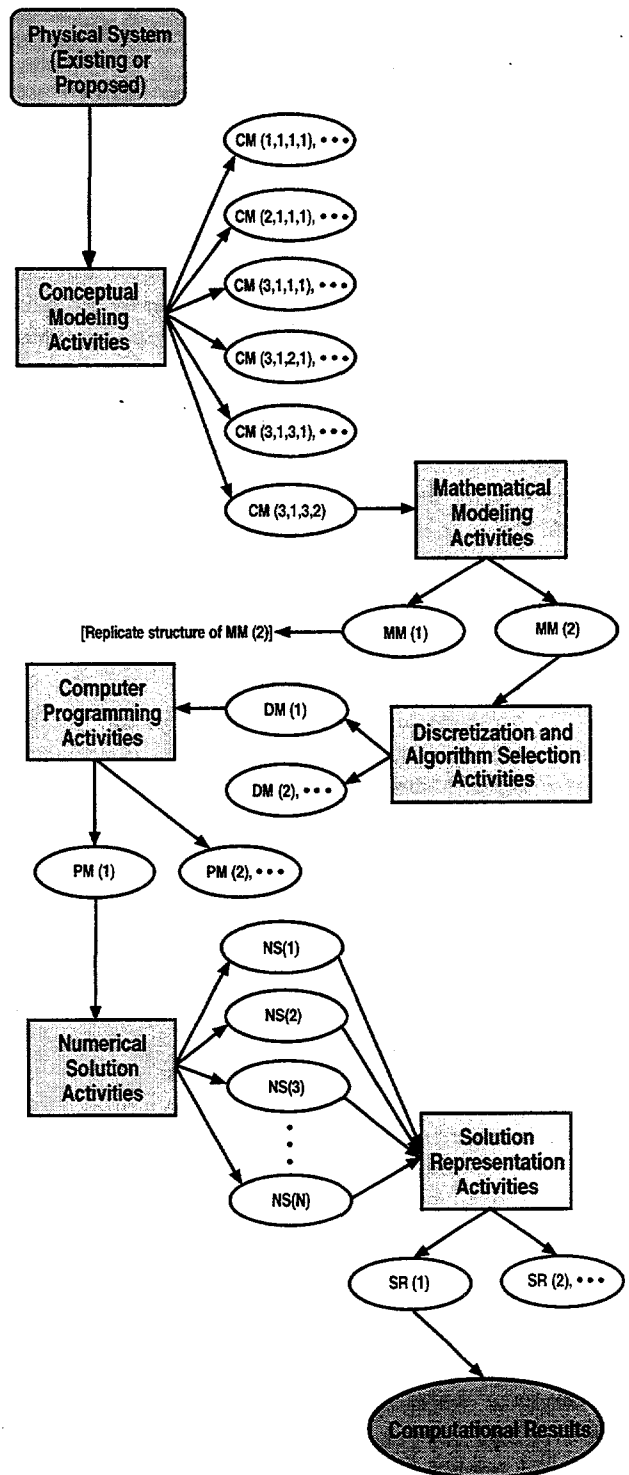


Figure 5
Tree-Structure for Models, Solutions, and Representations in the Missile Flight Example

interpretation errors does not include inappropriate decisions made by the user based on the interpretation, such as incorrect design choices or inept policy decisions.

3.8 Summary Comments

Figure 5 illustrates the multiple models, numerical solutions, and solution representations that are addressed in the missile flight example. For this example problem six conceptual models are identified, many more are implied, but only one is selected for further development and analysis. This single conceptual model spawns two alternative mathematical descriptions, both of which are carried through the remaining phases of the modeling and simulation process. For simplicity Figure 5 then shows the further development of only one of these mathematical models, although it is understood that identical development of the mathematical Model 1 is taking place in parallel with the mathematical Model 2. The discretization and programming phases identify alternative model choices that are not considered further. Continuing into the numerical solution phase, nondeterministic effects that were identified in the conceptual model and further defined in the mathematical modeling phase are computed via multiple deterministic numerical solutions. How these solutions were computed was specified in the propagation method identified in the discretization and algorithm selection phase. Finally, in the solution representation phase, the multiple solutions are reintegrated in order to represent the nondeterministic solution.

It is clear from this example that the modeling and simulation process for complex systems is fundamentally the identification and use of multiple scenarios, analyses, and computations. At each phase of this process it is frequently possible to identify more than one viable choice of models or parameters that can be used to obtain a computational result. As these multiple model choices propagate through subsequent phases, a tree structure of potential computational results is developed.

4. COMPUTATIONAL RESULTS

Before the computational results from the missile flight example are presented, a few details must be given concerning the calculations. The missile is assumed to be launched from an aircraft flying straight and level at an altitude of 30 kft. above sea level and at a speed of 700 ft/sec. Assume a spherical, non-rotating earth. Define an earth fixed, three-dimensional, cartesian coordinate system, where x is vertical, z is in the direction of the aircraft flight, and y is normal to the xz plane. Let the origin of the xyz coordinate system be at sea level. Assuming zero disturbance of the aircraft on the missile during launch and assuming uniform freestream flow approaching the missile, then the initial conditions for the 6-DOF equations of motion are

$$x = 30,000 \text{ ft.}, y = z = 0.$$

$$\begin{aligned} V_x = V_y = 0., V_z = 700 \text{ ft/sec.} \\ \alpha = \beta = \phi = 0. \\ p = q = r = 0. \end{aligned}$$

α , β , and ϕ are the pitch, yaw, and roll angles of the missile, respectively. p , q , and r are the roll rate, pitch rate, and yaw rate of the missile, respectively. The initial conditions for the 3-DOF equations of motion are given by the x , y , z and V_x , V_y , and V_z conditions given above. Assume the fluid properties of the atmosphere are given by the 1976 U.S. Standard Atmosphere and that the winds are zero over the entire trajectory.⁵⁵ The trajectory calculation is terminated when $x = 0.$, i.e., at sea level.

For convenience, detailed missile characteristics were taken to be those of the Improved Hawk missile, since these were readily available.⁵⁶ Missile moments of inertia, center of mass, rocket motor thrust, and mass flow rate of the rocket motor are given in Ref. 57. All of these parameters are functions of time during rocket motor operation but are constant after motor burnout. The rocket motor nominally operates for 24.5 sec., which is about half of the total flight time of the missile. The aerodynamic force and moment coefficients are assumed constant with pitch, yaw, and roll angle of the missile, i.e., linear aerodynamics is assumed. However, the aerodynamic force and moment coefficients are functions of Mach number. The system response presented is the range of the missile, since it captures most of the trajectory characteristics of interest. Complete flight dynamics results are given in Ref. 57.

To illustrate the combined effects of variability, uncertainty, and error in the example, we select 500 values of initial mass through the LHS method. We then compute 500 values of range using combinations of the two mathematical models, the three thrust models, and five selected values of solution error. Our purpose is to study and understand the effects and interactions of these sources of total uncertainty. We do not address in this paper how all possible sources, many of which have been suggested earlier, could be represented and propagated in this flight dynamics example nor do we address the most appropriate way to summarize the effects of these total uncertainty sources for decision makers.

4.1 Effects of Mass Variability

The first source of total modeling and simulation uncertainty examined was the variability of the initial mass of the missile. The mean initial mass was 1378.98 lbs, of which 732 lbs was inert mass and 647 lbs was propellant. As mentioned in Section 3.3, a normal probability distribution for initial mass variability was assumed. The standard deviation, σ_w , was assumed to be 10 lbs. Although it is not important for this example problem, $\sigma_w = 10$ lbs is consistent with actual missile

systems of this size.⁵⁸ We investigated the effects of numerical solution error for both the 6-DOF and 3-DOF models to be certain that this error was not entering into the mass variability results. We computed solutions with per step, relative, truncation error criteria of 10^{-12} , 10^{-9} , and 10^{-6} . Comparing these solutions at the end of the trajectory we found that error criterion of 10^{-12} and 10^{-9} produced the same values of the final range to seven significant digits. As a result, we used 10^{-9} for all remaining calculations when solution error was not of interest. Using this error criterion the computer run time on a SUN Sparc 20 workstation was 49 sec. and 1 sec., respectively, for one 6-DOF and one 3-DOF solution.

Since computer run time was not an issue, we computed 500 Latin Hypercube Sample (LHS) solutions for both the 6-DOF and 3-DOF models. Shown in Fig. 6 is the histogram from the LHS centering the weight at 1379 lbs and using bins of width 5 lbs. As can be seen with this number of samples, the histogram is a good approximation to the assumed normal distribution. Using LHS and 500 samples, the mean value was computed to be 1378.984 lbs, and $\sigma_w = 9.993$ lbs. The 500 samples is roughly a factor of 10 higher than is normally needed. We chose this large number so that we could essentially eliminate any sampling error in the analysis. Since the same random number generator and the same seed were used on both the 6-DOF and 3-DOF models, each model computed trajectories using exactly the same missile weights. Indeed, for all results given in this paper, exactly the same sampled initial missile weights were used.

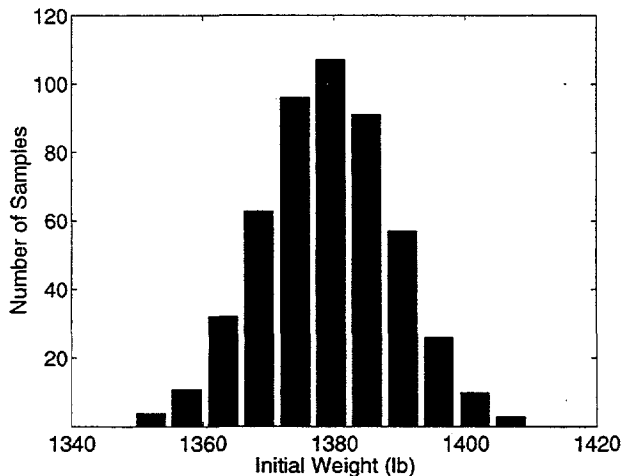


Figure 6
Histogram from LHS for Mass Variability

Figure 7 shows the computed range of the missile as a function of the initial mass for both the 6-DOF and 3-DOF models. The nominal thrust profile for the rocket motor was used. For both models, the missile range is

linear for this small variation in initial weight. However, the lower fidelity model (3-DOF) introduces a bias error of 0.040 nm in range which is constant for all weights sampled. The generation of a bias error in the response of the system is disturbing because it might go undetected if the higher fidelity model results or experimental measurements were not available. One does not, in general, expect this result. Lower fidelity models are used with the hope that the computational results will at least be distributed around the correct answer. For this relatively simple physics system one can easily see how this bias error in range occurs. The arching trajectory of the missile in a plane causes a small positive mean angle of attack during most of the trajectory. Computational results from the 6-DOF trajectory show this value to be about 0.01 to 0.02 deg. after the initial disturbance at launch decays.⁵⁷ This angle of attack causes a lift component on the missile, i.e., a small gliding effect, which results in a slightly longer trajectory. The lower fidelity model does not account for this physics, and as a result, the prediction of range is consistently shorter. From this understanding of the physics, one can then see that the magnitude of the bias will depend on a host of additional parameters that were not investigated, e.g., initial launch altitude, initial launch angle, and the aerodynamic lift coefficient.

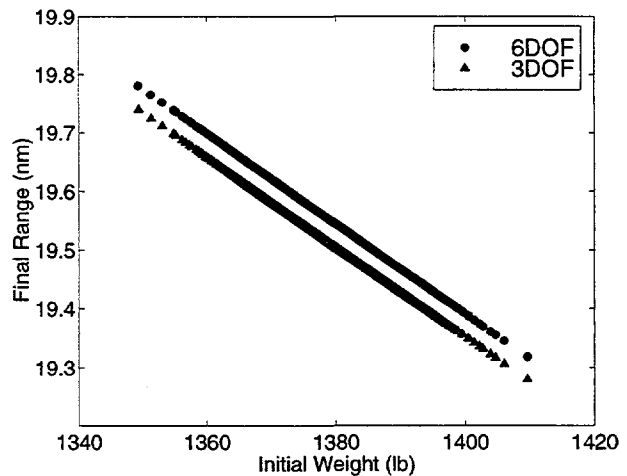


Figure 7
Variability in Range due to Variability in Initial Weight

Figure 8 shows frequency data of the LHS samples as a function of range off-set from the mean value range for the 6-DOF trajectory: 19.552 nm. That is, the range computed for the mean weight of 1379.05 lbs for the 6-DOF trajectory is defined to have zero offset. In this figure the bias error in range of 0.040 nm of the 3-DOF model is seen as a shift of distribution to the left, i.e., shorter range. The frequency plot shows the distribution produced by each model is remarkably similar, as might

be expected from the results of Fig. 7. For the 6-DOF model the standard deviation in range is computed to be 0.0770, whereas for the 3-DOF model, $\sigma_R = 0.0766$.

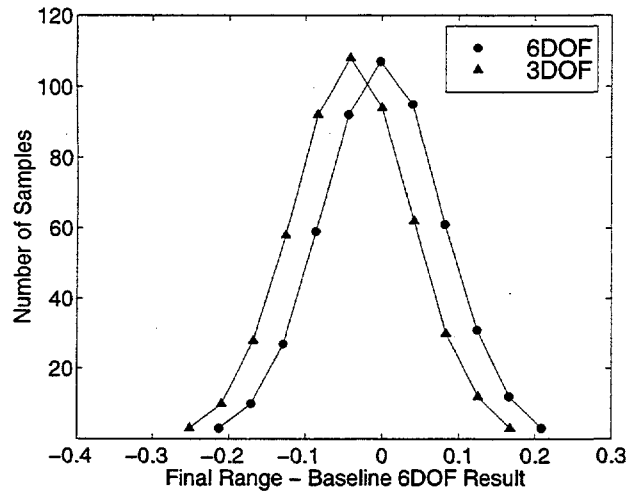


Figure 8
Frequency Data from LHS for Range Offset
Due to Initial Weight

When nearly identical frequency data are computed for different fidelity models and computer resources restrict making all computations using the high fidelity model, the following design of computer experiment strategy is commonly used. Runs are initiated with the same random number seed with each model, and the distributions of the ranges of each model are plotted. These runs are compared to determine if a bias shift in the mean of the distributions has occurred. If a bias does occur, then the lower fidelity model is "calibrated" to eliminate the bias and then used for the hundreds or thousands of runs typically needed for estimation of total uncertainty. This same calibration strategy is used in computational simulations of complex processes when experimental measurements are used for the benchmark.

4.2 Effects of Thrust Uncertainty

As we discussed in Sections 3.3 and 3.4, our approach to determining the uncertainty in the trajectories due to uncertain temperature of the rocket motor is to compute bounding trajectories using three thrust profiles: a nominal profile, the highest profile resulting from the highest temperature allowed by the manufacturer, and the lowest profile resulting from the lowest temperature allowed by the manufacturer. To be representative of thrust uncertainty in actual motors, we chose the changes in performance that have been experimentally measured for the Standard Hawk motor.⁵⁹ At the highest allowed temperature of 120° F, the total impulse of the motor is 2% above the nominal performance, but the burn time is

decreased by 7%. At the lowest allowed temperature of -20° F, the total impulse of the motor is 2% below the nominal performance, and the burn time is increased by 7%. Stated qualitatively, the high temperature motor has a higher net performance over a shorter burn time, and the cold motor has a lower net performance over a longer burn time.

Figure 9 shows the 6-DOF computed range of the missile for each of the three temperature conditions of the motor as a function of initial weight variability. It can be seen from Fig. 9 that, as expected, the motor temperature uncertainty produces a shift in range: the high temperature motor flying 0.625 nm further than the nominal motor temperature, and the cold motor flying 0.616 nm shorter than the nominal motor. The linearity of the missile range as a function of weight continues to hold for both the high and low motor temperature cases. It is also seen that the uncertainty in range due to motor temperature uncertainty is significantly larger than that observed due to weight variability. The uncertainty in range due to uncertain rocket motor temperature is 1.24 nm. The uncertainty in range due to mass variability can be calculated as $4\sigma_R = 4 \times 0.077 = 0.308$ nm, which is only 25% of the uncertainty due to thrust.

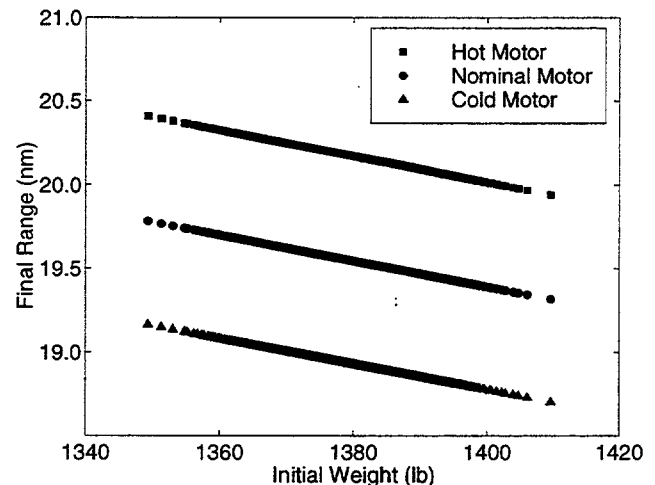


Figure 9
Uncertainty in Range due to Thrust Uncertainty and Mass
Variability for 6-DOF Model

Figure 10 shows the frequency data from the LHS for the 6-DOF model as a function of missile range for each of the three motor temperatures. The mean range for the cold motor is shifted 0.62 nm toward shorter range, whereas the hot motor is shifted the same amount toward longer range. The standard deviation in range for the hot and cold motors are nearly identical: $(\sigma_R)_{hot} = 0.0773$ nm and $(\sigma_R)_{cold} = 0.0762$ nm. Recall these are essentially the same as the value of the nominal motor, $(\sigma_R)_{nom} = 0.0770$ nm. The results for the hot and cold

motors using the 3-DOF model are very similar to the 6-DOF results presented in this section. The only difference is that the 3-DOF results show the 0.040 nm bias in range, as discussed in the previous section.

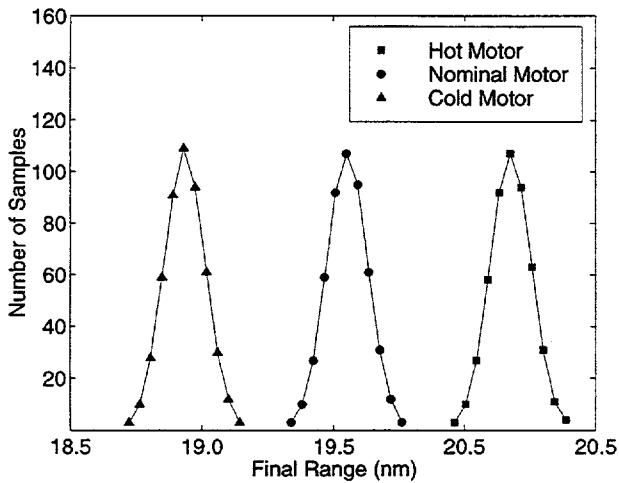


Figure 10
Frequency Data from LHS for Range Uncertainty due to Thrust Uncertainty for 6-DOF Model

We argue that the source of the potentially large uncertainty in missile performance due to motor temperature uncertainty should be characterized as lack of knowledge. Some would argue that the motor temperature uncertainty could be characterized as a variability instead of an uncertainty. The argument is that a probability distribution could be constructed based on experimentally measuring motor temperatures for a large number of actual missile deployments. The variability of motor temperature could then be represented by a probability distribution with some mean and standard deviation. Although this is a reasonable approach, we argue that the variability approach could lead to misleading estimates of system performance for certain deployment situations. For example, if the deployment was in Alaska during the winter versus Saudi Arabia during the summer, the average range of the missile would be of little value. Additional knowledge of the type of deployment could change the representation. A deployment at a permanent installation with significant environmental controlled space would be quite different than a make-shift battlefield deployment. As more and different kinds of knowledge are introduced into the analysis, representations other than probability distributions may be more appropriate, e.g., fuzzy sets, belief functions, and possibility theory. Guidance on developing these representations based on available information, is not as well developed as probability theory.

4.3 Effects of Solution Error

As discussed in Sections 3.4 and 3.6, we are able to precisely control the numerical solution error at each step of the numerical integration of the ODEs. The per step, relative, truncation error is estimated using the Runge-Kutta-Fehlberg 4(5) method, and the time step is adjusted at each step so that the truncation error is less than the specified error criterion. Figure 11 shows the computed range of the missile for the 6-DOF model using the nominal thrust profile as function of the mass variability for five different per step, relative error criteria. There is no effect on calculated range even though the error criterion is varied over eight orders of magnitude: up to 10% error per step. This was not expected. Intuition leads us to believe that as the error criterion increased greatly, the accuracy of the solution would degrade. For certain state variables, for example, those that are periodic, the solution accuracy degrades only slightly. Most variables, including output variables that are derived from state variables, like range, do not degrade because the error criterion must be satisfied by all 12 state variables. The state variables that have the highest frequency are those that will restrict the growth of the time step and the resulting growth in solution error. The highest frequency state variables are the pitch rate, q , and the yaw rate, r . Both have a frequency of 1 to 2 Hz. This limits the maximum time step to 0.1 to 0.2 sec. so that this element of physics can be adequately computed. All lower frequency state variables are computed much more accurately than required by the error criterion.

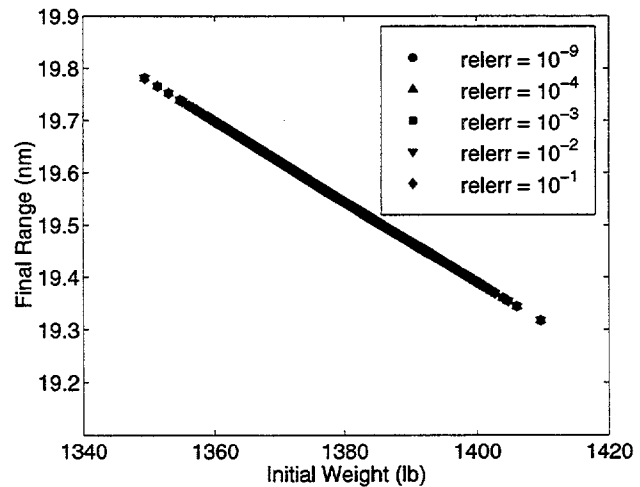


Figure 11
Uncertainty in Range due to Solution Error and Mass Variability for 6-DOF Model

Figure 12 shows the 3-DOF computed range using the nominal thrust profile as a function of mass variability for five values of the per step, relative, truncation error criterion. These five error criteria are the

same as used in the 6-DOF calculation illustrated in Fig. 11. The 3-DOF model has a completely different sensitivity to numerical solution error as compared to the high fidelity model. For a relative error of 10^{-4} slight roughness in the range as a function of weight can be seen. For a 10^{-3} error, the amplitude in roughness of range increases to 0.035 nm. This variation in amplitude can occur over a very small change in weight. For example, near the mean weight of 1379 lbs, a jump of 0.035 nm can be seen over a change in weight of less than one-tenth of a pound. This type of predicted system response roughness due to solution error has been seen by many investigators, particularly those using first order response surface methods and those using optimization methods that rely on numerical differentiation of the system response.

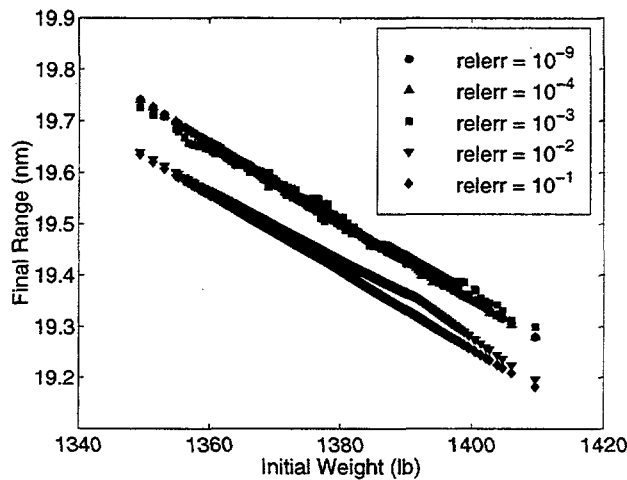


Figure 12
Uncertainty in Range due to Solution Error
and Mass Variability for 3-DOF Model

As the numerical error is increased further, to 10^{-2} and 10^{-1} , Fig. 12 shows that a drop in the predicted range occurs. This introduces a bias error in range similar to that observed in the earlier comparison of the 3-DOF and 6-DOF models. The bias error varies slightly with weight for 10^{-2} error but becomes constant at a value of 0.10 nm for 10^{-1} error. In addition, the range becomes an extraordinarily smooth function of weight: the same characteristic occurring at errors of 10^{-5} and smaller. To understand these unusual characteristics due to solution error, one must examine how the integration step size is changing to control the per step error in the state variables of the 3-DOF model. Contrary to the 6-DOF model, there are *no* periodic state variables in the 3-DOF system. As a result, the step size can increase rapidly from the fixed initial value of 0.1 sec., i.e., all solutions presented in this paper attempt to use $\Delta t = 0.1$ sec. in stepping from $t = 0$. If the step size results in an

estimated truncation error that satisfies the error criterion, then the step is taken. If the estimated error is 0.6 of the error criterion, then the time step is increased for the next step. If it does not meet the error criterion, then the time step is decreased until the error criterion is met. For the 3-DOF model, the time step increases rapidly because all of the state variables are extremely smooth as a function of time, relative to the 6-DOF model. When the error criterion is changed from 10^{-3} to 10^{-4} , Fig. 12, there is a rapid loss in accuracy of the major physical characteristic of the 3-DOF trajectory: the motor thrust profile. From the initial condition until 4.5 sec. the motor thrust is roughly 19,000 lbs. Then it rapidly drops to a sustained thrust value of about 3,600 lbs. for 20 sec., after which thrust terminates. For error criteria less than 10^{-5} , the numerical solution very accurately captures these two rapid drops in thrust. As the error criteria increases up to 10^{-3} , the numerical error becomes more erratic, depending on how the time steps fall with regard to the two rapid drops in thrust. For error criteria of 10^{-2} up to 10^{-1} , the error requirement becomes so loose that the time steps jump across the rapid drops in thrust with little notice.

4.4 Effects of Variability, Uncertainty, and Error

This section discusses the computational results for the combination of the mass variability, thrust uncertainty, and solution error for both the 6-DOF and 3-DOF models. Shown in Fig. 13 is the 6-DOF computed range as a function of mass variability, for all three thrust profiles, for the complete range of numerical solution error. As was seen in Figs. 9 and 11, the dominant characteristic is the very smooth variation in range as a function of initial weight for all three thrust profiles, regardless of the numerical solution error. The high temperature and low temperature motor cases are just as insensitive to solution error as the nominal motor temperature case shown in Fig. 11. The frequency plots, although not shown here, also show essentially no effect of solution error. For example, for the cold motor for a 10^{-9} and 10^{-1} relative error the mean range and standard deviation are, respectively: $R = 20.1771$ and $\sigma_R = 0.0773$ and $R = 20.1766$ and $\sigma_R = 0.0774$. As discussed earlier, the higher fidelity model is remarkably insensitive to solution error because of the temporal fine scale structure controlling the time step.

Figure 14 shows the 3-DOF computed range as a function of mass variability for a hot motor for the complete range of solution errors. As the solution error increases, the range calculation becomes even more erratic than that seen earlier for the 3-DOF model in Fig. 12. For an error criterion of 10^{-3} and for weight samples in the range of 1349 to 1355 lbs, a nearly constant bias

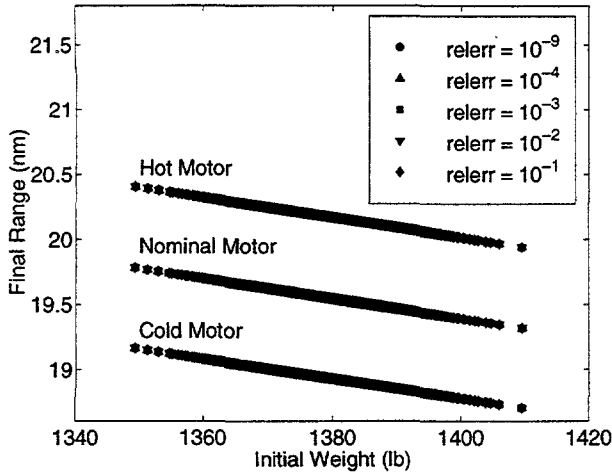


Figure 13
Uncertainty in Range due to Mass Variability, Thrust
Uncertainty, and Solution Error for 6-DOF Model

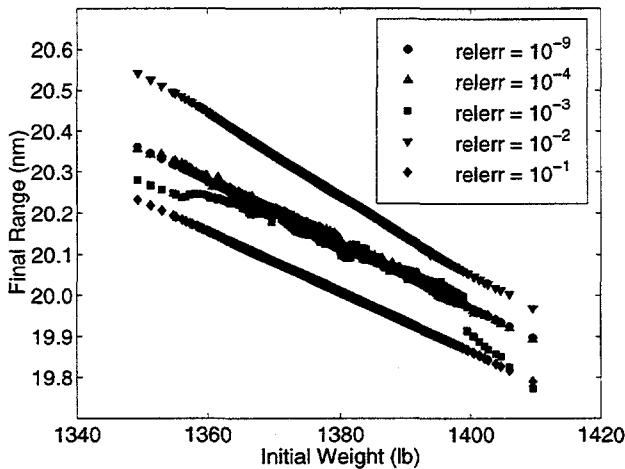


Figure 14
Uncertainty in Range due to Mass Variability and
Solution Error for the High Temperature Motor for 3-
DOF Model

of 0.07 nm toward shorter range occurs. For weights higher than 1355 lbs, the computed range wanders back to near the correct value. At a weight of 1399 lbs, a discontinuous drop of 0.08 nm in range occurs. When the error criterion increases to 10^{-2} , the bias switches to longer ranges for all weights. When the error criterion increases to 10^{-1} , the bias error switches back to shorter ranges for all weights. The frequency plots for this hot motor case, although not shown here, also show erratic behavior for error criteria greater than 10^{-5} . That is, the frequency plots for range show even more sensitivity to solution error than the plot of range as a function of

weight. The computed characteristics of the cold motor case are similar to those for the hot motor case.

4.5 Final Comments on the Results

Probably the most surprising computational results obtained in this example problem are those related to the aggregation and interaction of numerical solution error with variability and uncertainty. The counter intuitive result that the higher fidelity model is much less sensitive to solution error than the lower fidelity model needs further comment. The discussion given earlier for the controlling factor in solution error for each model provides an explanation to why this surprising result occurs. These results have implications for the effect of numerical solution error on uncertainty analyses when the mathematical model equations are given by PDEs. The per-step numerical solution error in the present work was precisely controlled by the adaptive step-size control of the ODE integrator. This level of solution error control and robustness does not presently exist in the numerical solution of PDEs. Even if one only considers elliptic boundary value problems, robust adaptive grid generation for the control of local spatial discretization error does not presently exist. For certain special cases, such as linear boundary value problems or problems with no large gradients, reliable methods for adaptive grid control do exist.

It is our view that the present results for the widely different sensitivity of each mathematical model to solution error would only apply to the numerical solution of PDEs with robust, adaptive, grid generation methods. If one were to use non-adaptive grid generation methods for the solution of the PDEs, very different sensitivities would occur than those observed here. Non-adaptive grid methods would be analogous to a constant time step method in the solution of ODEs. For the present example problem we computed numerical solutions using a constant time step over the length of the trajectory for the 6-DOF and 3-DOF models. Table 1 shows the numerical error in range for various constant time steps for both models using the nominal weight and nominal thrust. As the time step increases, the numerical error for both models increases, but the 6-DOF model error increases more rapidly. When the time step becomes roughly half of the period of the finest scale structure in the 6-DOF model, the error increases exponentially. For a time step of 0.1, the error in the 6-DOF solution has become so large that the trajectory is no longer computable. For the 3-DOF model, the same time steps cause a gradual increase in the solution error. This table shows the opposite sensitivity to numerical error as compared to the adaptive time step method.

Time Step	0.001	0.01	0.07	0.09	0.1
6-DOF	0.0000	0.004	0.038	0.397	∞
3-DOF	0.0000	0.004	0.040	0.052	0.058

Table 1
Error in Range for 6-DOF and 3-DOF for Constant Time Steps

5. SUMMARY AND CONCLUSIONS

We have presented a new framework for modeling and simulation that combines three viewpoints; the systems view from operations research, methods for propagation of uncertainty, and the numerical solution of partial differential equations. The activities that are conducted in each of the phases of modeling and simulation were discussed in detail. We have carefully defined the meaning of and distinguished among variability, uncertainty, and error. In each of the activities in each phase of modeling and simulation we discuss which type of source (variability, uncertainty, or error) typically dominates the activity. Particular emphasis is given to those features of the framework that deal with the nondeterministic features of the analysis, such as representations of variability and uncertainty, and propagation methods. Our framework applies regardless of whether the discretization procedure for solving the partial differential equations is based on finite elements, finite volumes, or finite differences.

This framework was applied to a missile flight example problem. The analysis began at a high-level system view and culminated with the computation of the flight dynamics of the missile. Many conceptual model alternatives were discussed so that the multitude of scenario and analysis options for a complex system can be better recognized. Although this example resulted in the solution of ordinary differential equations rather than PDEs, most of the features of the new framework could be demonstrated. The example showed how alternate mathematical models of the physical phenomena might be compared and utilized. As is common in complex system analyses, a hierarchy of mathematical models can be identified and employed. However, constraints on computer resources commonly require that the higher fidelity model be used to calibrate the lower fidelity model, similar to the use of experimental data in the calibration of complex simulations. Research is needed in the future to develop guidelines concerning how the use of multiple mathematical models can be optimized, given the constraints of varying model fidelity and widely different computer requirements needed for solution. This example computation yielded unexpected results for the

sensitivity of alternate mathematical models to numerical solution error. There can be widely differing sensitivities to numerical error depending on whether solution error is controlled with adaptive mesh refinement versus fixed mesh size. We suggest that adaptive mesh refinement will also produce much more robust results for response surface methods, sensitivity analyses, and optimization methods.

ACKNOWLEDGEMENTS

We thank David Salguero of Sandia National Laboratories for providing generous assistance and advice on the computer code TAOS. Larry Rollstin, also of Sandia, provided the Improved Hawk missile characteristics along with helpful advice on rocket systems. We also thank Rob Easterling and Tim Trucano of Sandia for their comments and suggestions in reviewing an earlier version of this paper.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U. S. Department of Energy under Contract DE-AC04-94AL85000.

REFERENCES

1. Hora, S. C., and Iman, R. L., "Expert Opinion in Risk Analysis: The NUREG-1150 Methodology," *Nuclear Science and Engineering*, Vol. 102, 1989, pp. 323-331.
2. Morgan, M. G., and Henrion, M., *Uncertainty: A guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, 1st ed., Cambridge University Press, New York, 1990.
3. Boyack, B. E., Catton, I., Duffey, R. B., Griffith, P., Katsma, K. R., Lellouche, G. S., Levy, S., Rohatgi, U. S., Wilson, G. E., Wulff, W., and Zuber, N., "Quantifying Reactor Safety Margins Part 1: An Overview of the Code Scaling, Applicability, and Uncertainty Evaluation Methodology," *Nuclear Engineering and Design*, Vol. 119, 1990, pp. 1-15.
4. Breeding, R. J., Helton, J. C., Gorham, E. D., and Harper, F. T., "Summary Description of the Methods used in the Probabilistic Risk Assessments for NUREG-1150," *Nuclear Engineering and Design*, Vol. 135, 1990, pp. 1-27.
5. Hauptmanns, U., and Werner, W., *Engineering Risks Evaluation and Valuation*, First Edition ed., Springer-Verlag, 1991.
6. Bergeron, K. D., Siezak, S. E., and Leach, C. E., "Proposed Deterministic Severe Accident Criteria for the Heavy Water Reactor-New Production Reactor Containment," *Nuclear Safety*, Vol. 34, No. 1, 1993, pp. 20-32.

7. Modarres, M., *What Every Engineer should know about Reliability and Risk Analysis*, Marcel Dekker, Inc., 1993.
8. Helton, J. C., "Treatment of Uncertainty in Performance Assessments for Complex Systems," *Risk Analysis*, Vol. 14, No. 4, 1994, pp. 483-511.
9. Kumamoto, H., and Henley, E. J., *Probabilistic Risk Assessment and Management for Engineers and Scientists*, IEEE Press, New York, NY, 1996.
10. Ang, A. H. S., and Tang, W. H., *Probability Concepts in Engineering Planning and Design: Vol. I Basic Principles*, 1st ed., John Wiley & Sons, New York, NY, 1975.
11. Ditlevsen, O., *Uncertainty Modeling with Applications to Multidimensional Civil Engineering Systems*, 1st ed., McGraw-Hill, Inc., New York, 1981.
12. Ayyub, B. M., Gupta, M. M., and Kanal, L. N., eds. *Analysis and Management of Uncertainty: Theory and Applications*, 1st ed., North-Holland, New York, 1992.
13. Ayyub, B. M., and Gupta, M. M., eds. *Uncertainty Modelling and Analysis: Theory and Applications*, 1st ed., North-Holland, New York, 1994.
14. Downing, D. J., Gardner, R. H., and Hoffman, F. O., "An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models," *Technometrics*, Vol. 27, No. 2, 1985, pp. 151-163.
15. Beck, M. B., "Water Quality Modeling: A Review of the Analysis of Uncertainty," *Water Resources Research*, Vol. 23, No. 8, 1987, pp. 1393-1442.
16. Bogen, K. T., and Spear, R. C., "Integrating Uncertainty and Interindividual Variability in Environmental Risk Assessment," *Risk Analysis*, Vol. 7, No. 4, 1987, pp. 427-436.
17. Helton, J. C., Garner, J. W., and M. G. Marietta, e. a., "Uncertainty and Sensitivity Analysis Results Obtained in a Preliminary Performance Assessment for the Waste Isolation Pilot Plant," *Nuclear Science and Engineering*, Vol. 114, 1993, pp. 286-331.
18. Ballin, P. R., Aziz, K., Journel, A. G., and Zuccolo, L., "Quantifying the Impact of Geological Uncertainty on Reservoir Performing Forecasts," Society of Petroleum Engineers, *12th Symposium on Reservoir Simulation*, New Orleans, LA, 1993.
19. Draper, D., "Assessment and Propagation of Model Uncertainty," *Journal of the Royal Statistical Society B*, Vol. 57, No. 1, 1995, pp. 45-97.
20. Laskey, K. B., "Model Uncertainty: Theory and Practical Implications," *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, Vol. 26, No. 3, 1996, pp. 340-348.
21. Oberkampf, W. L., Diegert, K. V., Alvin, K. F., and Rutherford, B. M., "Variability, Uncertainty, and Error in Computational Simulations," ASME-HTD-Vol. 357-2, *Proceedings of the AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, Albuquerque, NM, 1998, pp. 259-272.
22. Zeigler, B. P., *Theory of Modelling and Simulation*, 1st ed., John Wiley & Sons, New York, 1976.
23. Zeigler, B. P., *Multifaceted Modelling and Discrete Event Simulation*, 1st ed., Academic Press, Orlando, 1984.
24. Jacoby, S. L. S., and Kowalik, J. S., *Mathematical Modeling with Computers*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1986.
25. Bossel, H., *Modeling and Simulation*, 1st ed., A. K. Peters, Ltd., Wellesley, MA, 1994.
26. Schlesinger, S., "Terminology for Model Credibility," *Simulation*, Vol. 32, No. 3, 1979, pp. 103-104.
27. Sargent, R. G., "Simulation Model Validation," in *Simulation and Model-Based Methodologies: An Integrative View*, T.I. Oren, B.P. Zeigler, and M.S. Elzas Eds., Springer-Verlag Berlin Heidelberg, Syracuse, 1984, pp. 537-555.
28. Sargent, R. G., "An Expository on Verification and Validation of Simulation Models," *1985 Winter Simulation Conference*, Sacramento, CA, 1985.
29. Nance, R. E., "Model Representation in Discrete Event Simulation: The Conical Methodology," Virginia Polytechnic Inst. and State University, Dept. of Computer Science, Technical Rept. CS81003-R.
30. Balci, O., "Guidelines for Successful Simulation Studies," *Proceedings of the 1990 Winter Simulation Conf.*, New Orleans, LA, 1990.
31. Ayyub, B. M., "The Nature of Uncertainty in Structural Engineering," in *Uncertainty Modelling and Analysis: Theory and Applications*, B.M. Ayyub and M.M. Gupta Eds., Elsevier, New York, 1994, pp. 195-210.
32. Klir, G. J., and Folger, T. A., *Fuzzy Sets, Uncertainty, and Information*, 1st ed., Prentice Hall, Englewood Cliffs, NJ, 1988.
33. Ayyub, B. M., ed. *Uncertainty Modeling and Analysis in Civil Engineering*, CRC Press, Boca Raton, FL, 1998.
34. Vamos, T., "Epistemic Background Problems of Uncertainty," IEEE Computer Society Press, *First International Symposium on Uncertainty Modeling and Analysis*, College Park, MD, 1990.
35. Cox, E., *The Fuzzy Systems Handbook*, AP Professional, New York, NY, 1998.
36. Oberkampf, W. L., and Blotner, F. G., "Issues in Computational Fluid Dynamics Code Verification and Validation," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 687-695.
37. Lewis, R. O., *Independent Verification and Validation*, 1st ed., John Wiley & Sons, Inc., New

- York, 1992.
38. Knepell, P. L., and Arangno, D. C., *Simulation Validation: A Confidence Assessment*, 1st ed., IEEE Computer Society Press, Washington, 1993.
 39. Neelamkavil, F., *Computer Simulation and Modelling*, 1st ed., John Wiley & Sons, New York, 1987.
 40. Tenner, E., *Why Things Bite Back*, Alfred A. Knopf, New York, NY, 1996.
 41. Yee, H. C., and Sweby, P. K., "Global Asymptotic Behavior of Iterative Implicit Schemes," *International Journal of Bifurcation and Chaos*, Vol. 4, No. 6, 1994, pp. 1579-1611.
 42. Yee, H. C., and Sweby, P. K., "Aspects of Numerical Uncertainties in Time Marching to Steady-State Numerical Solutions," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 712-724.
 43. Kloeden, P. E., and Platen, E., *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, New York, 1992.
 44. Ang, A. H. S., and Tang, W. H., *Probability Concepts in Engineering Planning and Design: Vol. II Decision, Risk, and Reliability*, John Wiley & Sons, New York, NY, 1984.
 45. McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239-245.
 46. Iman, R. L., and Conover, W. J., "A Distribution-Free Approach to Introducing Rank Correlation Among Input Variables," *Communications in Statistics, Simulation and Computation*, Vol. 11, No. 3, 1982, pp. 311-334.
 47. Box, E. P., and Draper, N. R., *Empirical Model-Building and Response Surfaces*, Wiley and Sons, New York, NY, 1987.
 48. Klir, G. J., "Probabilistic versus possibilistic conceptualization of uncertainty," in *Analysis and Management of Uncertainty: Theory and Applications*, B.M. Ayyub, M.M. Gupta, and L.N. Kanal Eds., North-Holland, New York, 1992, pp. 13-26.
 49. Bier, V. M., "Fuzzy set theory, probability theory, and truth functionality," in *Analysis and Management of Uncertainty: Theory and Applications*, B.M. Ayyub, M.M. Gupta, and L.N. Kanal Eds., North-Holland, New York, 1992, pp. 65-78.
 50. Muhanna, R. L., and Mullen, R. L., "Development of Interval Based Methods for Fuzziness in Continuum Mechanics," IEEE Computer Society, *The Third International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society*, College Park, MD, 1995.
 51. Almond, R. G., *Graphical Belief Modeling*, 1st ed., Chapman & Hall, London, U. K., 1995.
 52. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in FORTRAN*, Cambridge University Press, New York, NY, 1992.
 53. Cash, J. R., and Karp, A. H., "A Variable Order Runge-Kutta Method for Initial-Value Problems with Rapidly Varying Right-Hand Sides," *A C M Transactions on Mathematical Software*, Vol. 16, No. 3, 1990, pp. 201-222.
 54. Salguero, D. E., "Trajectory Analysis and Optimization Software (TAOS)," Sandia National Labs., Albuquerque, NM, in preparation.
 55. NOAA, NASA, and USAF, "U. S. Standard Atmosphere, 1976," U. S. Government Printing Office, Washington, DC, 1976.
 56. Rollstin, L. R., *personal communication*, 1998.
 57. Oberkampf, W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., and Alvin, K. F., "A New Methodology for the Estimation of Total Uncertainty in Computational Simulation," Sandia National Laboratories, Albuquerque, NM, in preparation.
 58. Rollstin, L. R., and Fellerhoff, R. D., "Aeroballistic and Mechanical Design and Development of the Talos-Terrier-Recurit (Tater) Rocket System with Flight Test Results," Sandia National Laboratories, SAND74-0440, Albuquerque, NM, Feb. 1976.
 59. CPIA, "Rocket Motor Manual," Chemical Propulsion Information Agency, CPIA/M1, Laurel, MD.

Intentionally Left Blank

APPENDIX D:

**USE OF METAMODELS AND
QUALITATIVE INPUTS FOR ASSESSMENT
AND PROPAGATION OF ERRORS DUE TO
UNCERTAINTY IN MODEL FORM**

Intentionally Left Blank

November 5, 1997

**USE OF METAMODELS AND QUALITATIVE INPUTS FOR
ASSESSMENT AND PROPAGATION OF ERRORS DUE TO
UNCERTAINTY IN MODEL FORM**

PREPARED FOR

SANDIA NATIONAL LABORATORIES
ALBUQUERQUE, NEW MEXICO 87185

PREPARED BY

DANIEL C. KAMMER
ASSOCIATE PROFESSOR

DEPARTMENT OF ENGINEERING PHYSICS
UNIVERSITY OF WISCONSIN-MADISON
1500 JOHNSON STREET
MADISON, WI 53706

EXECUTIVE SUMMARY

A technique is presented for studying the effects of model-form uncertainty and model parameter uncertainty on the results of numerical simulations. A generalized metamodeling approach is used in which model-form is included as a qualitative input and quantitative inputs can be either global to the metamodel or nested within a specific model-form. A method is also presented for the systematic development of a corresponding experiment design based on the D-optimality criterion and Effective Independence. A straightforward least squares approach is used to fit the metamodel using the developed experiment design. Multiple simulation outputs are easily handled. The "goodness" of the computed metamodel is quantified using 12 criteria which are described. The approach is applied to a simple axially forced cantilever beam structure represented by three different models. Both root mean square and peak displacement of the end of the beam are considered as outputs. Quadratic polynomial based metamodels are fit and their goodness is examined using the criteria suggested. The metamodels are applied to quantify the effects of uncertainty on the simulation outputs using a decoupled Monte Carlo analysis in which the inputs are assumed to be normally distributed and prior probabilities are assumed for the model forms. The extreme results and output probability distributions compare favorably with the results of a much more computationally intensive Monte Carlo analysis. The proposed generalized metamodeling approach provides a valuable tool to economically investigate the propagation of both model-form and model parameter uncertainties in both deterministic and nondeterministic analyses which would otherwise be computationally prohibitive.

TABLE OF CONTENTS

	Page
1. Introduction	1
2. Development of Metamodels with Model-Form Inputs	3
3. Numerical Experiment Design	7
4. Quantification of Metamodel Accuracy	10
4.1 Coefficient of Multiple Determination - R^2	10
4.2 t-Test Statistic for Metamodel Coefficients	11
4.3 Variance Inflation Factor	12
4.4 Condition Number	12
4.5 Maximum Residuals	12
4.6 Hat-Diagonal - Influential Observations	13
4.7 Cook's Distance - Influential Observations	13
4.8 Durbin-Watson D Statistic	14
4.9 Residual Plots vs. Predicted Values	15
4.10 Residual Plots vs. Independent Variables	15
4.11 Test for Constant Variance	15
4.12 Shapiro-Wilk Test for Normality	15
5. Numerical Example	16
6. Conclusions	22
7. References	23
Appendix	33

1. Introduction

Numerical simulations of system response are used throughout the research and industrial communities to help make important system design and performance decisions. These simulations contain errors and uncertainties from many diverse sources. In order to make proper decisions, it is important to understand how errors and uncertainties propagate through computational simulations into computed results. A particularly important source of error is the analytical model of the system to be simulated. Sources of error and uncertainty within analytical models include discretization effects, nonlinearities, parameter uncertainties, improper model form, etc.

Model parameter uncertainty has probably received the most attention in the literature. Sensitivity analyses can be performed in an attempt to quantify uncertainty in the simulation output by adjusting model parameters and running the numerical solution in a repeated fashion. Depending on the size and complexity of the simulation, the computational cost of this type of analysis can be quite high. If the uncertainties in model parameters follow known probability distributions, the ultimate goal of uncertainty quantification would be to determine the corresponding probability distributions of the simulation outputs of interest. Accurate prediction of these distributions requires Monte Carlo analysis incorporating many thousands of simulation runs. In most cases, the computational cost is prohibitive.

In many situations, the computational cost associated with running the full simulation during deterministic sensitivity analyses, optimization analyses, and nondeterministic analyses can be greatly reduced by generating what is commonly referred to as a *Response Surface* or *Metamodel*. An appropriate functional form is assumed for the simulation response of interest and then fit to the output produced by the simulation of a set of numerical experiments. In contrast with the full numerical simulation, evaluation of the metamodel is very fast. Assuming that the underlying system is itself deterministic, the prediction of the response can be decoupled from the probabilistic Monte Carlo sampling using the metamodel. This is often referred to as "decoupled" Monte Carlo analysis.

The metamodel approach has recently received attention by the structural dynamics community. Applications include structural optimization [1, 2] and probabilistic design [3-5]. Romero and Bankston, from Sandia, are currently investigating the use of a response surface based on finite element / lattice sampling in decoupled Monte Carlo analysis [6]. All of this work focuses on model uncertainty due to uncertainty in model parameters. An equally important source of uncertainty is due to uncertain model form. While a seemingly small amount of work has been per-

formed considering model form uncertainty from the standpoint of Bayesian analysis [7-10], none has been found within the context of structural dynamics and the use of metamodels in decoupled Monte Carlo.

This report details work that has been performed to include model form as a qualitative variable within a metamodel that also includes quantitative variables representing model parameters. This provides a general metamodel approach that can be used to investigate the effects of both model form and model parameter uncertainty in both deterministic and nondeterministic analysis. The report presents the generalized metamodeling approach, an approach for experiment design, quantification of metamodel accuracy, and an application to a simple example.

2. Development of Metamodels with Model-Form Inputs

The inputs to an analytical model can be categorized as either quantitative or qualitative. Quantitative variables are those that vary smoothly and continuously between their lower and upper bounds like mass, spring stiffnesses, elastic modulus, etc. On the other hand, qualitative input variables vary discretely and their value or level has little if no physical meaning. In this analysis, model form will be represented by a qualitative variable z that has as many levels as models that are to be considered. For example, if there are three different models, z would take on the values of $z_1 = -1$, $z_2 = 0$, and $z_3 = 1$, corresponding to models 1, 2, and 3, respectively.

In metamodel analysis, the quantitative variables are usually coded such that they vary continuously between -1.0 and 1.0 using the relation

$$x_i = \frac{X_i - X_{i0}}{\Delta_i} \quad \Delta_i = \frac{1}{2}(X_{i\max} - X_{i\min}) \quad (1)$$

where x_i is the i th coded input, X_i is the corresponding uncoded input, X_{i0} is the nominal value of the i th input, and $X_{i\min}$ and $X_{i\max}$ represent the lower and upper bounds on the range of the i th uncoded input. The nominal value of the coded input variable always corresponds to 0.0. Coding is performed to offset large possible differences in units between various model inputs which can cause numerical error during fitting of the metamodel.

An important characteristic of quantitative variables that is peculiar to the application of model-form is the fact that a particular input variable may only have physical meaning within a particular model. For example, suppose there are two models being considered for an axial beam, one is a continuous representation while the second is a lumped mass and spring model. Elastic modulus of the beam material would be a meaningful input variable for the first model while values of spring stiffnesses would be meaningful inputs for the second. Elastic modulus would not have any influence on the second model, while spring stiffnesses would not affect the first. These quantitative inputs are said to be *nested* within the corresponding models. Therefore, quantitative variables come in two classes, *global variables* x_i that are common to all models, such as an external forcing frequency, and *nested variables* $w_{i,j}$ representing the j th variable nested to the i th model. Note that $w_{i,j}$ could also represent a nested qualitative variable. The model form variable z is a global variable. Some quantitative or qualitative variables could be common to subsets of models. This third class of variables will not be distinguished in this work but can be readily handled by the approach that is presented.

In most metamodel analysis, quadratic surfaces are assumed in the form of simple polynomials in the input variables. This same approach is used here, but the formulation can be easily generalized to higher order surfaces and more sophisticated functions of the inputs, such as trigonometric functions, inverse functions, etc. Alternate basis function formulations will be the subject of future investigation. Initially assuming a single output y for the simulation, such as an rms displacement, the general form of the quadratic metamodel is given by

$$y = \beta_0 + \sum_{i=1}^l \beta_i x_i + \sum_{i=1}^l \sum_{j=i}^l \beta_{ij} x_i x_j + \sum_{i=1}^m f_i \left[\sum_{j=1}^{n_i} \alpha_{i,j} w_{i,j} + \sum_{j=1}^{n_i} \sum_{k=j}^{n_i} \alpha_{i,jk} w_{i,j} w_{i,k} + \sum_{j=1}^{n_i} \sum_{q=1}^l \varepsilon_{i,jq} w_{i,j} x_q \right] + \sum_{i=1}^{m-1} \gamma_i z^i + \sum_{i=1}^{m-1} z^i \left[\sum_{j=1}^l \delta_{i,j} x_j + \sum_{j=1}^l \sum_{k=j}^l \delta_{i,jk} x_j x_k \right] \quad (2)$$

in which:

- l = number of global input variables
- m = number of models
- n_i = number of nested input variables in model i
- β_i, β_{ij} = regression coefficients for global variables
- $\alpha_{i,j}, \alpha_{i,jk}$ = regression coefficients for variables nested to model i
- $\varepsilon_{i,jq}$ = regression coefficients for coupling between variables nested to model i and global variables
- γ_i = regression coefficients for model form variable
- $\delta_{i,j}, \delta_{i,jk}$ = regression coefficients for coupling between model form and global variables

Note that the global/nested formulation given by Eq. (2) contains the same number of terms as would be obtained by nesting the global terms within each model. However, many of the terms will later be found to be statistically insignificant and dropped from the metamodel. Due to this fact, the global/nested formulation will in general produce a final metamodel with fewer terms.

It is also of note that the sums on the second line of Eq. (2) can produce terms in the metamodel that are higher than second order in the input variables even though the metamodel is supposedly restricted to quadratic terms. For example, suppose there are four models with four levels for z and assume that there are no variations of any other global or nested inputs. The corresponding metamodel from Eq. (2) evaluated at all four model levels produces the set of equations

$$\begin{aligned}
\beta_0 + \gamma_1 z_1 + \gamma_2 z_1^2 + \gamma_3 z_1^3 &= y_1 \\
\beta_0 + \gamma_1 z_2 + \gamma_2 z_2^2 + \gamma_3 z_2^3 &= y_2 \\
\beta_0 + \gamma_1 z_3 + \gamma_2 z_3^2 + \gamma_3 z_3^3 &= y_3 \\
\beta_0 + \gamma_1 z_4 + \gamma_2 z_4^2 + \gamma_3 z_4^3 &= y_4
\end{aligned} \tag{3}$$

where y_i is the response for the i th model. It is apparent that, depending on the values of y_i , the metamodel could require up to four constants $(\beta_0, \gamma_1, \gamma_2, \gamma_3)$, and thus terms through z^3 , to accurately predict the response. In general, the metamodel should contain coupling terms between the global terms in the quadratic representation and terms in z up to the power of $m-1$.

The nesting of variables in Eq. (2) is handled by the introduction of the functions f_i which have the general form

$$f_i = \prod_{\substack{j=1 \\ j \neq i}}^m \frac{z - z_j}{z_i - z_j} \tag{4}$$

When the model form variable z takes on the value corresponding to the i th model, $f_i = 1$, otherwise, $f_i = 0$. In this manner, depending on the current value of z , the appropriate terms corresponding to the current model form are either retained or excluded in Eq. (2).

In order to calculate the metamodel coefficients in Eq. (2), a sequence of simulations, or numerical experiments, is performed using predetermined settings for the input variables. The data for the complete set of experiments is then combined into a matrix equation

$$X\theta = Y \tag{5}$$

in which X is the $n_e \times p$ experiment design matrix, θ is a column vector containing the model coefficients, Y is a column vector containing the responses from the numerical experiments, n_e is the number of experiments performed, and p is the number of terms in the metamodel. The coefficient vector is then estimated in the usual manner using least-squares

$$\hat{\theta} = (X^T X)^{-1} X^T Y \tag{6}$$

Each row of X corresponds to one of the prescribed experiments and each column corresponds a

term within the assumed metamodel. Initially, the full quadratic model in Eq. (2) is assumed. Analysis of variance (ANOVA) is performed leading to statistical tests that can determine the significance of each metamodel term [11]. Columns corresponding to insignificant terms are then dropped from the experiment design matrix X and the coefficients for the reduced metamodel are recomputed and checked again for significance. If additional terms are found to be insignificant, they are dropped and the process repeats. This process corresponds to a backwards stepwise regression.

If multiple outputs of the simulation are of interest, they can be handled simultaneously by initially assuming the same full order experiment design matrix for both and again forming matrix equation (2) where now for k different outputs

$$Y = [Y_1 \ Y_2 \ \dots \ Y_k] \quad \theta = [\theta_1 \ \theta_2 \ \dots \ \theta_k] \quad (7)$$

in which Y_i is a column vector containing the i th response for the n_e experiments and θ_i is a column containing the corresponding regression coefficients. Equation (7) can also be solved using the least-squares approach in Eq. (6). The regression coefficients in each column of θ are checked for significance and unimportant terms are dropped. Re-estimation and analysis of coefficients in the backward stepwise manner then proceeds on an individual response type basis. After final estimation, the models can be recombined into the form of Eq. (7) by adding zeros to the individual columns of coefficients at the appropriate locations to expand back to the original assumed metamodel size.

The proposed process essentially provides a procedure by which metamodels for individual model forms can be combined. This combined metamodel offers several advantages. Only one regression model needs to be fitted. By pooling the data from all the models, more degrees of freedom for error are obtained. The significance of terms containing the qualitative model form variable can be determined using statistical tests which sheds light on the significance of a particular model. The effect of model form on the response uncertainty can be determined by assuming some type of probability for each of the models in a decoupled Monte Carlo analysis.

3. Numerical Experiment Design

In order to achieve the best possible metamodel representation of a simulation, the combined parameter/model-form space must be sampled appropriately. The optimal points are selected based on statistical experiment design. There is a vast literature in this area proposing a number of standard designs such as Central Composite [12] and Box-Behnken [13], etc. These designs are usually implemented for the estimation of relatively low order metamodels containing a relatively small number of input variables. As the number of variables increases, coupled with the inclusion model-form with nested and global parameters, the standard designs become costly and difficult, if not impossible to generate. Therefore, this investigation is based upon computer generated experiment designs.

The best known and one of the most widely used approaches is the D-optimality design criterion [11] which attempts to minimize a measure of the variance associated with the estimates of the metamodel coefficients which is related to the moment matrix

$$M = \frac{1}{n_e} X^T X \quad (8)$$

An important norm on the moment matrix is given by its determinant

$$|M| = \frac{1}{n_e^p} |X^T X| \quad (9)$$

Under the assumption of independent normally distributed model errors with constant variance, the determinant of $X^T X$ is inversely proportional to the square of the volume of the confidence region corresponding to the regression coefficients. A large determinant implies a small confidence region which implies good estimates of the coefficients. A D-optimal design therefore maximizes $|M|$ over the design space.

There are several approaches available to determine the D-optimal design. Some, such as the genetic algorithm, can be quite time consuming. The approach used here is a suboptimal approach called Effective Independence (EfI) [14]. Initially, a candidate set of experiments D is produced. Each column of D represents one of the metamodel inputs in coded form. Columns corresponding to quantitative inputs are given by

$$D_i = \frac{2\pi_i - (N+1)}{N} \quad (10)$$

in which N is the number of candidate experiments selected by the analyst, and π_i is a uniform random permutation of the integers between 1 and N . This produces a variant of the Latin Hypercube [15] with centered values of the coded inputs between -1 and 1. The column of D corresponding to model-form is constructed as series of random permutations of the m values of z . In this work, N is selected to be evenly divisible by m so that each model is represented in the candidate set of experiments an equal number of times. Once the candidate numerical experiments are produced, the corresponding experiment design matrix X is generated using the general form assumed for the metamodel in Eq. (2).

The EfI method determines the importance of each candidate experiment in D by examining X and computing the N dimensional vector

$$E_D = [X\Psi]^{\wedge 2} \lambda^{-1} \{1\}_p \quad (11)$$

where Ψ is a matrix containing the eigenvectors of the information matrix $Q = X^T X$, \wedge^2 denotes a term-by-term square of the resulting matrix in the brackets and $\{1\}_p$ is a p dimensional column vector of 1's corresponding to the number of terms in the metamodel. It has been shown in Refs. [16] and [17] that the Effective Independence of the i th candidate experiment, E_{D_i} , is related to the determinant of the information matrix by the expression

$$E_{D_i} = \frac{|Q| - |Q_i|}{|Q|} \quad (12)$$

in which Q_i represents the information matrix with the i th candidate experiment deleted from X . Therefore, E_{D_i} corresponds to the fractional change in the determinant of the information matrix if the i th candidate experiment is deleted. The values of E_{D_i} thus satisfy the relation

$$0 \leq E_{D_i} \leq 1.0 \quad (13)$$

where a value of 0.0 indicates that the corresponding candidate experiment will contribute nothing to the identification of the metamodel coefficients, while a value of 1.0 indicates that the experiment is absolutely vital to the identification. The experiment design process proceeds by sorting

the entries in E_D and deleting the lowest ranked experiment. The remaining experiments are then reranked. In an iterative manner the large candidate set of experiments can be quickly reduced. As each experiment is deleted, the determinant of the information matrix becomes smaller. At each iteration, according to Eq. (12), the deletion of the lowest ranked experiment will produce the smallest change in the information matrix determinant. Therefore, the method tends to produce an experiment design that maximizes the determinant of the information matrix, which is also the goal of a D-optimal design. The number of required experiments in the reduced design can be determined by iteratively eliminating experiments to the point where only p are left, corresponding to the minimum number required for identification. At each iteration, the determinant of the moment matrix can be computed using Eq. (9). The design which produces the maximum value is used to identify the metamodel coefficients.

Variance criteria such as D-optimality assume that all the error associated with the metamodel is random and do not consider systematic error or bias resulting from an insufficient metamodel. It has been noted that D-optimal designs tend to concentrate sample points at the fringes of the design space while designs that spread the samples throughout design space provide greater protection against bias. It is believed that a weighting matrix can be used in conjunction with EfI to penalize candidate experiments that are "far" from the center of design space. This could be used to offset the tendency of the D-optimality criterion to place experiments at the corners of design space. Future work will examine this issue in more detail.

4. Quantification of Metamodel Accuracy

In order for the metamodel to be a useful tool, its accuracy needs to be quantified. Fox [4] suggests twelve different tests that can be used to determine the goodness of the computed metamodel. Each is briefly described in the following paragraphs.

4.1 Coefficient of Multiple Determination - R^2

This statistic gives a measure of the amount of variability in the simulation data about the mean value that is explained by the metamodel. A value of 1.0 implies that all variability in the data is explained by the metamodel. In the backward stepwise regression process mentioned previously, the process of dropping insignificant terms from the metamodel can continue until there is a non-trivial change in the value of the statistic. The statistic is computed using the expression

$$R^2 = \frac{SSR}{SST} \times 100 \quad (14)$$

in which SST is the total sum of squares and SSR is the sum of squares due to regression given by

$$SST = \sum_{i=1}^{n_e} (Y_i - \bar{Y})^2 \quad (15)$$

$$SSR = \sum_{i=1}^{n_e} (\hat{Y}_i - \bar{Y})^2 \quad (16)$$

where \bar{Y} is the mean value of the simulation data Y_i and \hat{Y}_i is the corresponding value computed by the metamodel. This statistic is not always a good measure of metamodel goodness because it depends on the relative difference between the number of experiments n_e and the number of terms in the metamodel. For example, if $n_e = p$, R^2 is always 1.0.

The R^2 statistic can be adjusted to account for the relative difference between experiments and coefficients by computing the adjusted statistic R_A^2

$$R_A^2 = 1 - \left(\frac{n_e - 1}{n_e - p} \right) (1 - R^2) \quad (17)$$

While R^2 always decreases as terms are dropped from the metamodel, R_A^2 can actually increase if an unnecessary term is deleted. If there is a significant difference between R^2 and R_A^2 , there is a good chance that insignificant terms are included in the metamodel. Values close to 1.0 are obviously good. More details can be found in [11].

4.2 t-Test Statistic for Metamodel Coefficients

This statistic tests the null hypothesis for each of the estimated coefficients \hat{b}_i in the metamodel. It is computed using the expression

$$t_o = \frac{\hat{b}_i}{se(\hat{b}_i)} \quad (18)$$

in which $se(\hat{b}_i)$ is the estimated value of the standard error of \hat{b}_i given by

$$se(\hat{b}_i) = [c_{ii}s^2]^{1/2} \quad (19)$$

where c_{ii} is the i th diagonal element from covariance matrix Q^{-1} and s^2 is the estimate error variance

$$s^2 = \frac{1}{n_e - p} (SST - SSR) \quad (20)$$

Statistic t_o is compared against the corresponding value from a t distribution table. Assuming a two-sided test, if $|t_o| < t(\frac{\alpha}{2}, n_e - p)$, where α is the level of significance and $n_e - p$ is the number of degrees of freedom in the test, the null hypothesis is accepted and the corresponding coefficient is dropped from the metamodel. If $|t_o| > t(\frac{\alpha}{2}, n_e - p)$, the corresponding coefficient is found to be statistically significant and retained in the metamodel. It is important to remember that this test is a marginal one in that it tests for the significance of the i th coefficient assuming that the others in the estimated model are all present. This test is usually performed at the 95% confidence level or a value of $\alpha = 0.05$ and is equivalent to the p-value tests suggested by Fox.

4.3 Variance Inflation Factor

This statistic provides a measure of whether there is multicollinearity among the independent variables in the metamodel. A rule of thumb is that any $VIF_i > 10.0$ indicates that multicollinearity may be present [11]. At least two of the independent variables are highly correlated and only one of the two variables is needed in the metamodel. The statistic measures how much the variances of the estimated metamodel coefficients are inflated as compared to when the input variables are not linearly related. This is essentially a test for the independence of the columns in the experiment design matrix X . If $VIF_i = 1.0$, the i th column of X is not linearly dependent on the remaining columns of X . Otherwise, $VIF_i > 1.0$ implies an inflated variance for the corresponding metamodel coefficient. The statistic is computed by normalizing the columns of X to unit length and eliminating the column of 1's corresponding to the zero intercept coefficient β_0 , forming the matrix X_T . The i th variance inflation factor is computed as the matrix diagonal term

$$VIF_i = \left[(X_T^T X_T)^{-1} \right]_{ii} \quad (21)$$

4.4 Condition Number

Computing the condition number of the information matrix $Q = X^T X$ gives a measure of its ill-conditioning. Condition numbers greater than 100.0 indicate that some of the estimated metamodel coefficients may have a fair amount of numerical error. This test is basically another test for linear dependence of the columns of the experiment design matrix. It can be computed as the ratio of the largest to smallest eigenvalues of Q .

4.5 Maximum Residuals

The i th residual error is calculated as

$$e_i = Y_i - \hat{Y}_i \quad (22)$$

For the metamodel to be acceptable, the largest residual error e_{max} must be acceptably low for the case of interest. An estimated metamodel can satisfy all the listed goodness criteria, but if it produces errors that are unacceptable, it is of no use to the analyst.

4.6 Hat-Diagonal - Influential Observations

This statistic is computed for each data point that is used to estimate the metamodel. It searches for data points or experiments that have profound influence on the estimated model and a small residual. This implies that if the data point was removed from the estimation analysis, the resulting metamodel would be vastly different. The metamodel or response surface tries to bend dramatically to fit through the influential data point. The best metamodels are those that fit a trend to the data rather than twisting and turning too much. This statistic is based on the Hat matrix

$$H = X(X^T X)^{-1} X^T \quad (23)$$

The Hat matrix is an orthogonal projector onto the column space of X . It is therefore idempotent, meaning that its trace is equal to its rank, which in this case is p . The average value of the diagonal terms h_{ii} is p/n_e . The rule of thumb is that if

$$h_{ii} > \frac{2p}{n_e} \quad (24)$$

which is the average value, then the i th experiment is said to be a high leverage point and should perhaps be deleted from the experiment design matrix [11].

4.7 Cook's Distance - Influential Observations

This statistic also checks for influential data points in the experiment design, but unlike the Hat-Diagonal, it looks for outliers, i.e. points with large residuals. Often, h_{ii} and D_i give analogous results, but sometimes they will identify different points, so both statistics should be computed. The i th experiment is said to be influential if the corresponding Cook's distance D_i is greater than the 50th percentile of an F distribution with numerator degrees of freedom given by p and denominator degrees of freedom given by $n - p$. Practically, a value of 1.0 is usually used as a cutoff [11]. Therefore, if $D_i > 1.0$, the i th experiment is influential and should be dropped from the experiment design. The statistic is computed using the relation

$$D_i = \frac{r_i^2}{p} \frac{h_{ii}}{(1 - h_{ii})} \quad (25)$$

where r_i is the i th studentized residual given by

$$r_i = \frac{e_i}{[s^2(1-h_{ii})]^{1/2}} \quad (26)$$

4.8 Durbin-Watson D Statistic

This measure of goodness checks if the error terms in the metamodel are independent. Error terms in a good metamodel have nothing to do with each other. Fox [4] states that in general, a value of D close to 2.0 indicates that the errors are not correlated. The statistic basically tests for a correlated error term within the metamodel

$$y_i = \beta_0 + \beta_1 x_{1i} + \dots + \varepsilon_i \quad (27)$$

where y_i is the response from the i th experiment and the error term is

$$\varepsilon_i = \rho\varepsilon_{i-1} + u_i \quad (28)$$

This is an autoregressive error model where ρ is the autocorrelation parameter. If $\rho = 0.0$, $\varepsilon_i = u_i = N(0, \sigma^2)$ and the errors are not correlated. If $\rho < 0.0$, there is negative correlation and if $\rho > 0.0$ there is positive correlation between the errors. The usual hypotheses are tested

$$H_0 : \rho = 0 \quad H_a : \rho > 0$$

The D statistic is computed as

$$D = \frac{\sum_{i=2}^{n_e} (e_i - e_{i-1})^2}{\sum_{i=1}^{n_e} e_i^2} \quad (29)$$

Upper and lower bounds, d_L and d_U , respectively, are given in tables in many texts such Neter et al [18] to be used with the following decision rules.

If $D > d_U$: conclude H_0

If $D < d_L$: conclude H_a
If $d_L \leq D \leq d_U$: test is inconclusive

If a test for negative correlation is required, compute $D^- = 4 - D$ and apply the same decision process as with D . To use a two-sided test

$$H_0 : \rho = 0 \quad H_a : \rho \neq 0$$

employ both one-sided tests separately but at a confidence level of $\frac{\alpha}{2}$.

4.9 Residual Plots vs. Predicted Values

This plot should show the residuals randomly scattered about zero. Otherwise, there may need to be some transformation of the response variable. There should be no systematic tendencies and the residuals should lie within a constant band about the origin indicating a constant variance. More can be found in Neter, et al, pp. 111-123 [18].

4.10 Residual Plots vs. Independent Variables

These plots should have the same characteristics of random scatter as the residual plots in Section 4.9. If not, this may indicate that the input variable may have to be transformed. This type of plot can be directly applied to global inputs but it seems to only be appropriate within the corresponding model for nested inputs.

4.11 Test for Constant Variance

The main assumptions in fitting a metamodel are that the errors or residuals follow a normal distribution with a mean of zero and a constant variance. This test checks for constant variance and is left for study in future work.

4.12 Shapiro-Wilk Test for Normality

This test checks to see if the residuals follow the desired normal distribution. If the errors are not normally distributed, an underlying assumption of the metamodel process has been violated and the corresponding model should not be used. This statistic is also left for future work.

5. Numerical Example

A simple numerical example is presented to illustrate the application of the metamodeling process to model-form uncertainty. The structure to be simulated, pictured in Fig. 1a., is a simple axial cantilevered beam with two sections of equal length but one section has twice the cross sectional area and thus twice the mass and stiffness. The total mass of the beam is 2.0 while the resultant stiffness with respect to the right end is 100.0. The beam is sinusoidally forced as shown with a frequency of 19.0 rad/sec and an amplitude of 300.0. The numerical simulation outputs of interest are the root mean square (rms) displacement and the peak displacement at the force application point.

Three different models are assumed for the axial beam. Model 1, shown in Fig. 1b. is simply a single degree of freedom oscillator in which half the total mass has been lumped in M_2 and the spring constant has been selected consistent with the overall beam stiffness. Model 2 is more refined by having two degrees of freedom. However both mass and stiffness are still evenly distributed within the model as illustrated in Fig. 1c. Model 3 is the most sophisticated of the three with the mass and stiffness distribution reflecting the actual beam. Values for the Model 3 components are shown in Fig. 1c. In this analysis, the model-form variable z , has the values -1, 0, and 1, for models 1, 2, and 3, respectively.

All model parameters, the forcing frequency, and the model-form will be considered as input variables for a metamodeling analysis. The D matrix representing a candidate set of experiments must first be generated. The i th row of D has 10 entries representing the 10 coded input variables in the order

$$D^i = \begin{bmatrix} x_1^i & w_{1,1}^i & w_{1,2}^i & w_{2,1}^i & w_{2,2}^i & w_{3,1}^i & w_{3,2}^i & w_{3,3}^i & w_{3,4}^i & z^i \end{bmatrix} \quad (30)$$

$$\omega \quad M1 \quad K1 \quad M2 \quad K2 \quad m1 \quad m2 \quad k1 \quad k2 \quad z$$

The corresponding physical input variables are listed just below. The Latin Hypercube variant given in Eq. (10) was used to generate 1,002 candidate experiments via a MATLAB function called "randesign" listed in Table A-1. The candidate set of experiments was then used to generate the candidate experiment design matrix X for the fully quadratic metamodel using Eq. (2) and another MATLAB function called "mod3buildX" listed in Table A-2.. The same metamodel form is initially assumed for both outputs. The full metamodel contains 41 terms and is given by the expression

$$\begin{bmatrix} y_{rms} & y_{peak} \end{bmatrix} = x \begin{bmatrix} \theta_{rms} & \theta_{peak} \end{bmatrix} \quad (31)$$

in which row vector x is given by

$$\begin{aligned} x = & [1 \quad x_1 \quad x_1^2 \quad f_1 w_{1,1} \quad f_1 w_{1,2} \quad f_1 w_{1,1}^2 \quad f_1 w_{1,2}^2 \quad f_1 w_{1,1} w_{1,2} \quad f_1 w_{1,1} x_1 \quad f_1 w_{1,2} x_1 L \\ & f_2 w_{2,1} \quad f_2 w_{2,2} \quad f_2 w_{2,1}^2 \quad f_2 w_{2,2}^2 \quad f_2 w_{2,1} w_{2,2} \quad f_2 w_{2,1} x_1 \quad f_2 w_{2,2} x_1 \quad f_3 w_{3,1} L \\ & f_3 w_{3,2} \quad f_3 w_{3,3} \quad f_3 w_{3,4} \quad f_3 w_{3,1}^2 \quad f_3 w_{3,2}^2 \quad f_3 w_{3,3}^2 \quad f_3 w_{3,4}^2 \quad f_3 w_{3,1} w_{3,2} L \\ & f_3 w_{3,1} w_{3,3} \quad f_3 w_{3,1} w_{3,4} \quad f_3 w_{3,2} w_{3,3} \quad f_3 w_{3,2} w_{3,4} \quad f_3 w_{3,3} w_{3,4} \quad f_3 w_{3,1} x_1 L \\ & f_3 w_{3,2} x_1 \quad f_3 w_{3,3} x_1 \quad f_3 w_{3,4} x_1 \quad z \quad z^2 \quad z x_1 \quad z^2 x_1 \quad z x_1^2 \quad z^2 x_1^2] \end{aligned} \quad (32)$$

and both θ_{rms} and θ_{peak} have the form

$$\begin{aligned} \theta = & [\beta_0 \quad \beta_1 \quad \beta_{11} \quad \alpha_{1,1} \quad \alpha_{1,2} \quad \alpha_{1,11} \quad \alpha_{1,22} \quad \alpha_{1,12} \quad \epsilon_{1,11} \quad \epsilon_{1,21} L \\ & \alpha_{2,1} \quad \alpha_{2,2} \quad \alpha_{2,11} \quad \alpha_{2,22} \quad \alpha_{2,12} \quad \epsilon_{2,11} \quad \epsilon_{2,21} \quad \alpha_{3,1} \quad \alpha_{3,2} \quad \alpha_{3,3} L \\ & \alpha_{3,4} \quad \alpha_{3,11} \quad \alpha_{3,22} \quad \alpha_{3,33} \quad \alpha_{3,44} \quad \alpha_{3,12} \quad \alpha_{3,13} \quad \alpha_{3,14} \quad \alpha_{3,23} \quad \alpha_{3,24} L \\ & \alpha_{3,34} \quad \epsilon_{3,11} \quad \epsilon_{3,21} \quad \epsilon_{3,31} \quad \epsilon_{3,41} \quad \gamma_1 \quad \gamma_2 \quad \delta_{1,1} \quad \delta_{2,1} \quad \delta_{1,11} \quad \delta_{2,11}]^T \end{aligned} \quad (33)$$

The candidate experiment matrix was then reduced to a suboptimal design using Efl and the MATLAB function "xexpdes" listed in Table A-3. The D-optimal design parameter given in Eq. (9) was computed and examined for the design that produced the maximum value. Fig. 2. illustrates the experiment design parameter versus the number of experiments retained. The maximum is attained at 86 experiments. Of the 86 selected, 21 corresponded to model 1, 22 corresponded to model 2, and 43 corresponded to model 3. This roughly corresponds the relative number of meta-model terms nested to each mode-form. Figure 3. illustrates a scatter plot showing the selected experiment points in the M1 vs. ω design space. Even though Efl is a variance reduction based algorithm, the selected points are still are fairly well spread throughout the design space.

All three models are statically equivalent and at a forcing frequency of $\omega = 2.0$ rad/sec. the rms responses for nominal models 1, 2, and 3, of 2.27, 2.23, and 2.19, respectively, are in rough agreement. However, in other forcing frequency regimes, they markedly disagree. The nominal forcing frequency of $\omega = 19.0$ rad/sec. was selected from such a region. It is above the fundamental natural frequency for all three models but is below the second frequency of models 2 and 3. It is important to remember that there is no damping in any of the models, therefore the corresponding response surfaces are not smooth. It is therefore believed that this example is a very stern test of

the proposed metamodling process.

In this example, a maximum variation of 10% was assumed for all quantitative inputs. This would correspond to a level of ± 1.0 in the coded variables. The 86 simulations, each with 1,000 time steps of data, were run and the metamodel fit using the MATLAB function "mod3afit" listed in Table A-4. Once the model was fitted, the coefficients for each output were checked for significance at the 95% confidence level. The response and corresponding metamodels are now treated separately. Each is reduced to only the significant terms computed in the first analysis and recomputed. After a couple iterations, 24 terms were found to be significant for the rms displacement and 30 terms were found to be significant for the peak displacement.

The *rms displacement metamodel* is given by

$$x_{rms} = [1 \quad x_1 \quad x_1^2 \quad f_1 w_{1,1} \quad f_2 w_{2,1} \quad f_2 w_{2,1} w_{1,2} \quad f_2 w_{2,1} x_1 \quad f_2 w_{2,2} x_1 \quad f_3 w_{3,1} L \\ f_3 w_{3,2} \quad f_3 w_{3,3} \quad f_3 w_{3,4} \quad f_3 w_{3,3}^2 \quad f_3 w_{3,1} w_{3,2} \quad f_3 w_{3,1} w_{3,3} \quad f_3 w_{3,2} w_{3,3} L \\ f_3 w_{3,1} x_1 \quad f_3 w_{3,2} x_1 \quad f_3 w_{3,3} x_1 \quad z \quad z^2 \quad z x_1 \quad z^2 x_1 \quad z x_1^2] \quad (34)$$

$$\theta_{rms} = [1.5791 \quad -0.2631 \quad 0.2627 \quad -0.1907 \quad -0.1211 \quad -0.0782 \quad 0.1534 \quad -0.1270 \quad -0.3314 \quad . \\ -0.3530 \quad 0.4293 \quad -0.0962 \quad 0.1160 \quad 0.1057 \quad -0.1307 \quad -0.1134 \quad 0.2508 \quad 0.3112 \quad . \\ -0.4023 \quad 0.5335 \quad 0.7158 \quad -0.4104 \quad -0.5158 \quad 0.2213]^T \quad (35)$$

The *peak displacement metamodel* has the form

$$x_{peak} = [1 \quad x_1 \quad x_1^2 \quad f_1 w_{1,1} \quad f_1 w_{1,1} x_1 \quad f_2 w_{2,1} \quad f_2 w_{2,2} \quad f_2 w_{2,2}^2 \quad f_2 w_{2,1} w_{1,2} L \\ f_2 w_{2,1} x_1 \quad f_2 w_{2,2} x_1 \quad f_3 w_{3,1} \quad f_3 w_{3,2} \quad f_3 w_{3,3} \quad f_3 w_{3,4} \quad f_3 w_{3,1}^2 L \\ f_3 w_{3,2}^2 \quad f_3 w_{3,3}^2 \quad f_3 w_{3,4}^2 \quad f_3 w_{3,1} w_{3,2} \quad f_3 w_{3,1} w_{3,3} \quad f_3 w_{3,2} w_{3,3} L \\ f_3 w_{3,1} x_1 \quad f_3 w_{3,2} x_1 \quad f_3 w_{3,3} x_1 \quad z \quad z^2 \quad z x_1 \quad z^2 x_1 \quad z x_1^2] \quad (36)$$

$$\theta_{peak} = [3.0135 \quad -0.4216 \quad 0.5745 \quad -0.3778 \quad 0.1256 \quad -0.1822 \quad -0.2175 \quad 0.1967 \quad -0.2633 \\ 0.5184 \quad -0.4162 \quad -0.6586 \quad -0.6669 \quad 0.8906 \quad -0.2509 \quad 0.1688 \quad 0.1859 \quad 0.2897 \\ 0.2230 \quad 0.2212 \quad -0.2902 \quad -0.2833 \quad 0.5170 \quad 0.6994 \quad -0.8174 \quad 1.0512 \quad 1.3884 \\ -0.8760 \quad -1.1979 \quad 0.4655]^T \quad (37)$$

The goodness of the metamodels must now be assessed. Figures 4. and 5. present comparisons between numerical simulation responses and corresponding responses predicted by the metamodels for rms displacement and peak displacement, respectively, for the 86 experiments used to fit the metamodels. The experiments have been sorted by model; experiments 1 through 21 correspond to model 1, experiments 22 through 43 correspond to model 2, and experiments 44 through 86 correspond to model 3. The metamodels appear to do a good job of representing the simulation responses over all three model-forms. The maximum absolute residual for the rms metamodel was 0.379 for an experiment corresponding to model 3. The maximum absolute percentage error for rms displacement was 8.04% for an experiment corresponding to model 2. In the case of peak response, the maximum absolute residual was 0.427 for an experiment corresponding to model 3. The maximum absolute percentage error for peak response was 6.92% for an experiment corresponding to model 2. For many applications, these errors would probably be considered as too large, indicating that the computed metamodels may not be satisfactorily accurate. As discussed previously, this example appears to be a very stern test of the method due to the existence of discontinuities or ridges in the response surfaces at resonances. Some experiments in the parameter space lie close to these ridges producing poor metamodel predictions.

Besides residuals, other goodness criteria discussed in Section 4. were applied to the two metamodels. Table 1 presents the results for both rms and peak displacement metamodels. The Coefficient of Multiple Determination R-square and the adjusted R-square values are close to 100.0 for both metamodels indicating that a large portion of the variation in the data is explained by the metamodels and there are no insignificant terms left in either. The maximum variance inflation factors are either less than or approximately equal to the rule of thumb maximum value of 10.0 indicating good estimates of the metamodel coefficients. The condition number of the information matrix for the peak displacement metamodel is a bit larger than the allowed value of 100.0 indicating that the estimates for the peak displacement metamodel coefficients might not be as accurate as those of the rms displacement metamodel. This is consistent with the results of the variance inflation factor analysis.

The maximum hat-diagonal values for both metamodels are below their allowable values indicating that there are no profoundly influential points with small residuals in the experiment designs. Both metamodels have a Cook's distance less than 1.0 indicating that there are no profoundly influential points in the experiment design with large residuals. Finally, the Durbin-Watson D statistic for each metamodel is close to 2.0, indicating in a practical sense that the residuals are not correlated. Residual plots were also examined. Figure 8. illustrates the residuals versus the values predicted by the metamodel for peak displacement. The residuals are randomly scattered

about zero with no systematic tendencies. Figure 9 presents the same residuals versus the forcing frequency. It also shows a random scatter about zero with no systematic tendencies. Analogous results were found for the rms displacement. Both metamodels seem to satisfy these other goodness criteria.

To examine the predictive capability of the metamodels, the rms and peak displacements were estimated for all 1,002 candidate experiments used in the suboptimal experiment design. Figures 6 and 7 plot the estimated rms and peak responses versus actual rms and peak responses for all 1,002 candidate experiments. The maximum absolute percentage errors for the rms and peak displacements were 8.73% and 13.29%, respectively. The maximum error in rms displacement corresponds to an experiment from model 3, while the maximum for peak displacement corresponds to model 1. Ultimately, the usefulness of a metamodel for a particular application is determined by the size of the maximum residuals. Certainly, the maximum percentage predictive residuals for the computed metamodels are quite large. However, over much of the parameter space, the metamodels do a relatively good job of prediction, reflected in the root mean square errors of 2.49% and 3.49% for rms displacement and peak displacement, respectively.

Finally, Monte Carlo and decoupled Monte Carlo analyses were performed in which normal distributions with 5.0% standard deviations were assumed for all quantitative inputs. The metamodels were fit assuming 10.0% or $\pm 2\sigma$ ($\pm 1/40$ extremes) variation on the inputs. This is classified by Fox [5] as moderately extreme. Ten thousand numerical simulations were performed in which the quantitative inputs were selected from the described normal distributions and the model-form was selected based upon user provided prior probabilities. The meaning of these prior model-form probabilities $P(M_i)$ is currently the subject of great debate. For example, three different possible definitions are: i) the probability that model i is correct, ii) the probability that model i is "best", the probability that model i is "good enough" [19]. The first definition is not very useful because all models are approximations of reality. Definitions ii) and iii) are very subjective and require definitions of "best" and "good enough". Definition ii) may be the most practical where "best" is defined with respect to a "goodness" index relative to a candidate set of models. This is a subject of future research and for the purposes of this example, the probabilities are assumed given with respect to the candidate set of three models.

The Monte Carlo analysis was performed using the MATLAB function "mod3arand" presented in Table A-5. One of the output matrices, D , contains the coded values of the inputs for each randomly selected experiment. The corresponding experiment design matrix X can then be generated using the MATLAB function "mod3buildX". A decoupled Monte Carlo analysis was then per-

formed using the metamodels calculated previously. Two cases were considered for this example. Case I assumed an equal probability for each model, while Case II assumed that model 3 was twice as likely as model 2 which was twice as likely as model 1. Table 2. presents the $\pm 1/100$ and $\pm 1/40$ extremes predicted by the Monte Carlo analyses and the metamodels. In both cases, the values of the $-1/100$ and $-1/40$ extremes are accurately predicted by the metamodels. However, the $+1/100$ extremes are not accurately predicted which is consistent with the fact that the metamodels had difficulty predicting extreme values within the experiment design used to generate them. The $+1/40$ extremes are very accurately predicted for the peak displacement and somewhat less accurately predicted for rms displacement.

Probability distributions for the outputs can be generated using frequency data from the Monte Carlo analyses. Figure 10 illustrates the frequency distribution for the rms displacement from the Monte Carlo analysis for Case I in which all models have equal probability. Figure 11. presents the corresponding distribution for the decoupled Monte Carlo analysis. The metamodel results do a good job of qualitatively describing the Monte Carlo distribution. The quantitative accuracy is not as good but still seems to be representative. Figures 12 and 13 illustrate the analogous results for Case II in which model 3 is twice as likely as model 2 which is twice as likely as model 1. The more complex nature of the distributions is apparent. It is obvious that model form plays an important role. Once again, the metamodel results qualitatively capture this additional complexity with somewhat less quantitative accuracy. Similar results were obtained for peak displacements. It is apparent that the decoupled Monte Carlo analysis using the metamodel produces useful results concerning the impact of model-form uncertainty on probability distributions for simulation outputs.

6. Conclusions

A technique has been presented for studying the effects of model-form uncertainty and model parameter uncertainty on the results of numerical simulations. A generalized metamodeling approach is used in which model-form is included as a qualitative input and quantitative inputs can be either global to the metamodel or nested within a specific model-form. A method is also presented for the systematic development of a corresponding experiment design based on the D-optimality criterion and Effective Independence. A straightforward least squares approach is used to fit the metamodel using the developed experiment design. Multiple simulation outputs can be easily handled. The goodness of the computed metamodel can be quantified using 12 criteria which have been described and implemented. The validated metamodel can then be used as an economical tool to investigate the effects of uncertainties in both deterministic and nondeterministic analyses such as decoupled Monte Carlo.

The proposed approach was applied to a simple axially forced cantilever beam structure represented by three different models. Both root mean square and peak displacement of the end of the beam were considered as outputs. Quadratic polynomial based metamodels were fit and their goodness was examined using the criteria suggested. In many cases, the metamodels were found to be good, except that they produced large errors at extreme values of the outputs found within the experiment design space. It is believed that the example chosen is a very stern test of the method due to the existence of discontinuities or ridges in the response surfaces at resonances. Some experiments in the parameter space lie close to these ridges producing poor metamodel predictions. The metamodels were then applied to quantify the effects of uncertainty on the simulation outputs using a decoupled Monte Carlo analysis in which the inputs were assumed to be normally distributed and prior probabilities was assumed for the model forms. The extreme results and output probability distributions compared favorably with the results of a much more computationally intensive Monte Carlo analysis.

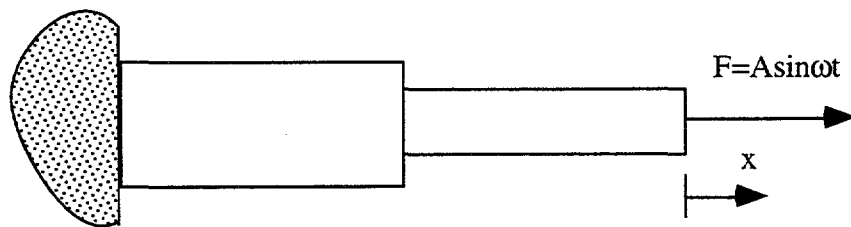
It is believed that the proposed generalized metamodeling approach can provide a valuable tool to economically investigate the propagation of both model-form and model parameter uncertainties in both deterministic and nondeterministic analyses which would otherwise be computationally prohibitive. Future research will concentrate on the use of advanced basis functions such as trigonometric or inverse functions to improve accuracy, further quantification of metamodel accuracy, the use of weighting in the experiment design to guard against bias, the relation between the statistical significance of model-form terms within the metamodel and the significance of individual models, how to assign meaningful prior probabilities to the model forms, and finally, the develop-

ment and investigation of ways to use metamodels to study the effects of model-form uncertainty.

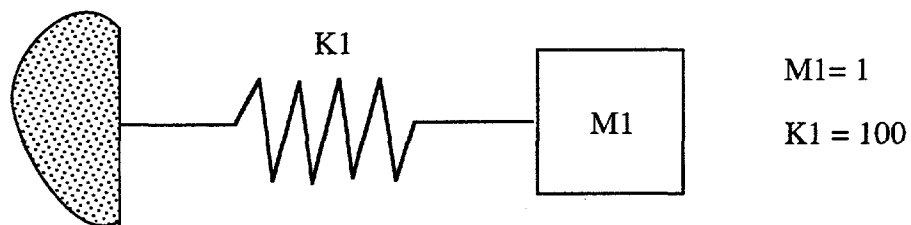
7. References

- [1] Kaufman, M. and e. al., "Variable-Complexity Response Surface Approximations for Wing Structural Weight in HSCT Design," *Computational Mechanics*, **18**, 1996, pp. 112-126.
- [2] Venter, G. and R. T. Haftka, "Minimum-Bias Based Experiment Design for Constructing Response Surfaces in Structural Optimization," *38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, FL, 1997, pp. 1225-1238.
- [3] Fox, E. P., "Methods of Integrating Probabilistic Design within an Organization's Design System using Box-Behnken Matrices," *34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, La Jolla, CA, 1993, pp. 714-723.
- [4] Fox, E. P., "The Pratt & Whitney Probabilistic Design System," *35th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Hilton Head, SC, 1994, pp. 1075-1085.
- [5] Fox, E. P., "Issues in Utilizing Response Surface Methodologies for Accurate Probabilistic Design," *AIAA Structures, Structural Dynamics, and Materials Conference*, Salt Lake City, UT, 1996, pp. 1615-1622.
- [6] Romero, D. R. and S. D. Bankston, "Application of Decoupled Monte Carlo Analysis with Finite Element / Lattice Sampling Response Surface for Multimodal Test Problem," Sandia National Laboratories, 1997.
- [7] Sacks, J. S., S. B. Schiller and W. J. Welch, "Designs for Computer Experiments," *Technometrics*, **31**(1), 1989, pp. 41-47.
- [8] Sacks, J., W. J. Welch, T. J. Mitchell and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, **4**(4), 1989, pp. 409-435.
- [9] Currin, C., T. Mitchell, M. Morris and D. Ylvisaker, "Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, **86**(416), 1991, pp. 953-963.

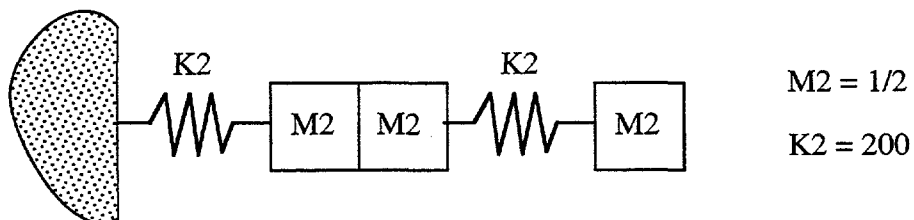
- [10] Draper, D., "Assessment and Propagation of Model Uncertainty," *Journal of the Royal Statistical Society*, **57**(1), 1995, pp. 45-97.
- [11] Myers, R. H. and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York, 1995.
- [12] Box, G. E. P. and K. B. Wilson, "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society, Series B*, **213**, 1951, pp. 1-45.
- [13] Box, G. E. P. and D. W. Behnken, "Some New Three-Level Designs for the Study of Quantitative Variables," *Technometrics*, **30**, 1960, pp. 1-40.
- [14] Kammer, D. C., "Sensor Placement for On-Orbit Modal Identification and Correlation of Large Space Structures," **14**(2), 1991, pp. 251-259.
- [15] Koehler, J. R. and A. B. Owen, "Computer Experiments," *Handbook of Statistics*, edited by S. Ghosh and C. R. Rao, Vol. **13**, Elsevier Science, 1996, pp. 261-308.
- [16] Poston, W. L. and R. H. Tolson, "Maximizing the Determinant of the Information Matrix with the Effective Independence Method," *Journal of Guidance, Control, and Dynamics*, **15**(6), 1992, pp. 1513-1514.
- [17] Kammer, D. C. and L. Yao, "Enhancement of On-Orbit Modal Identification of Large Space Structures through Sensor Placement," *Journal of Sound and Vibration*, **171**(1), March 1994, pp. 119-140.
- [18] Neter, J., W. Wasserman and M. H. Kutner, *Applied Linear Statistical Models*, Richard D. Irwin, Inc., Homewood, IL, 1985.
- [19] Atwood, C., V. Bier, D. Kleinmuntz, B. Sagar, A. Shlyakhter, N. Siu and R. Winkler, "Discussion Group III: Quantification of Model Uncertainty," *Model Uncertainty: Its Characterization and Quantification*, edited by A. Mosleh, 1993, pp. 247-251.



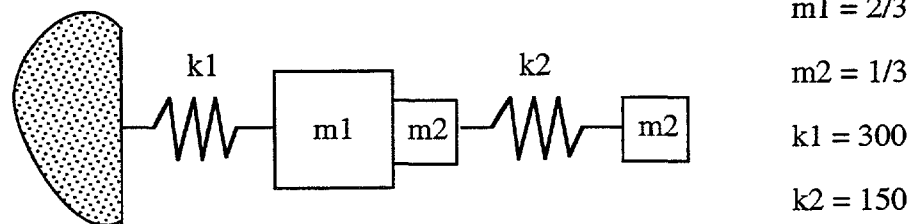
(a) System



(b) Model 1



(c) Model 2



(d) Model 3

Fig. 1 System to be simulated and discrete model representations

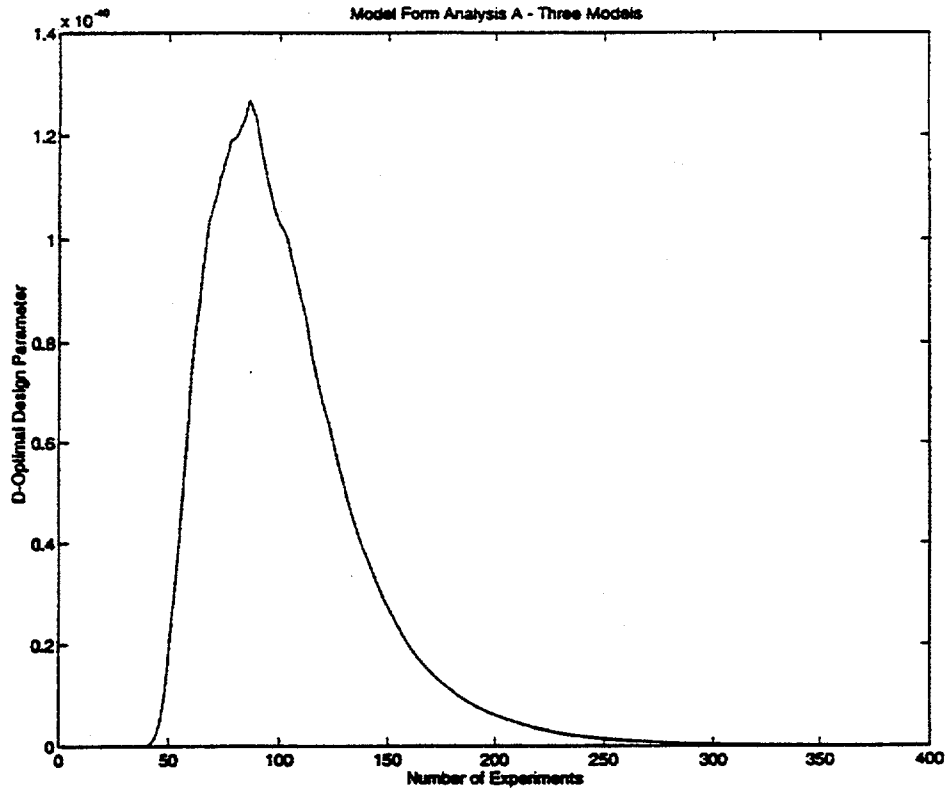


Fig. 2. D-Optimal design parameter.

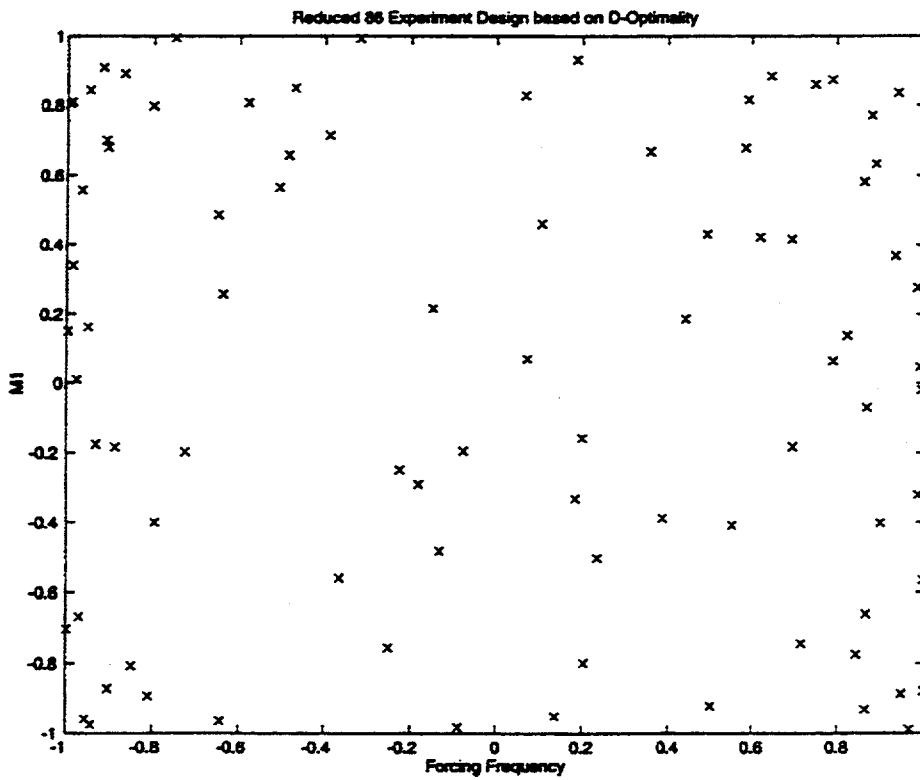


Fig. 3. Selected experiment design points.

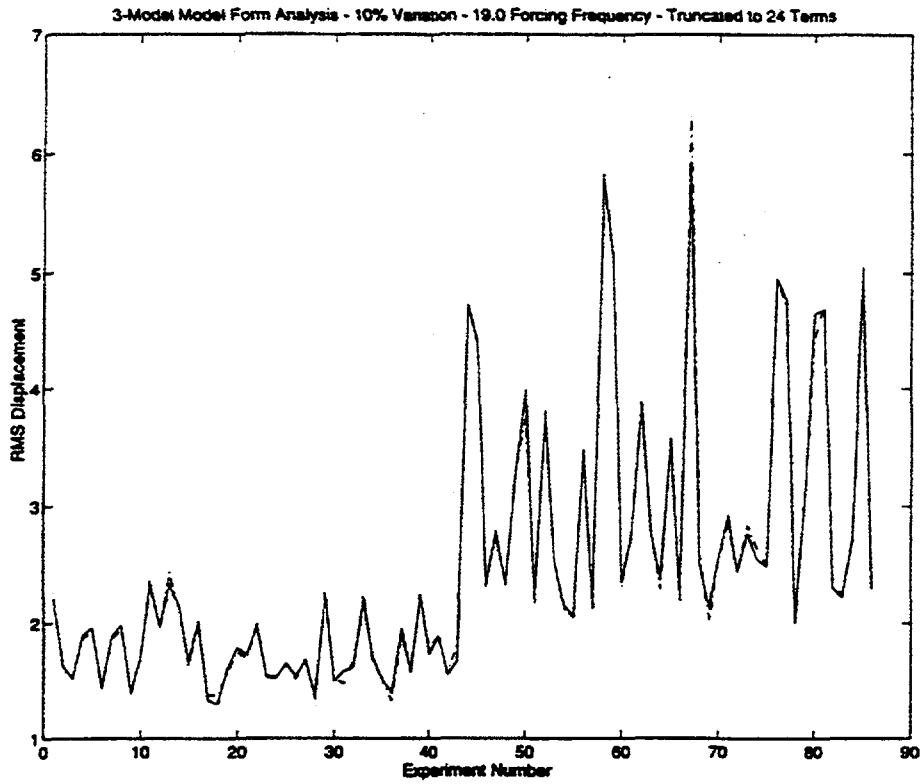


Fig. 4. RMS displacement for 86 point experiment design sorted by model.
 - - - exact, ___ estimated

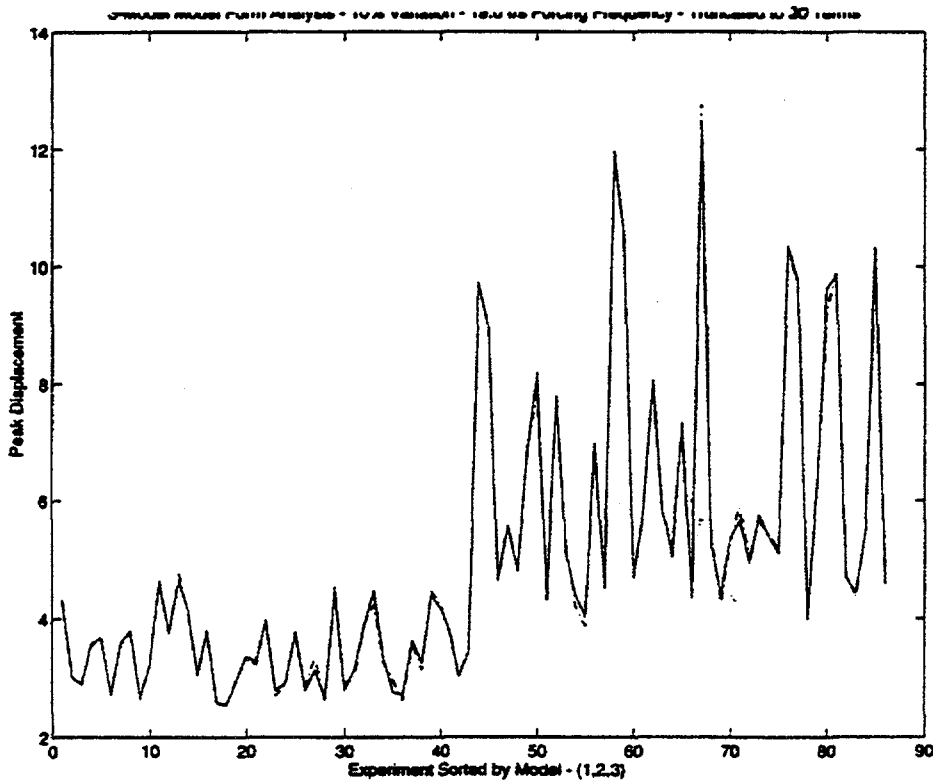


Fig. 5. Peak displacement for 86 point experiment design sorted by model.
 - - - exact, ___ estimated

24 Term Metamodel Prediction of 1002 Candidate Experiments - 19 r/s Forcing Frequency

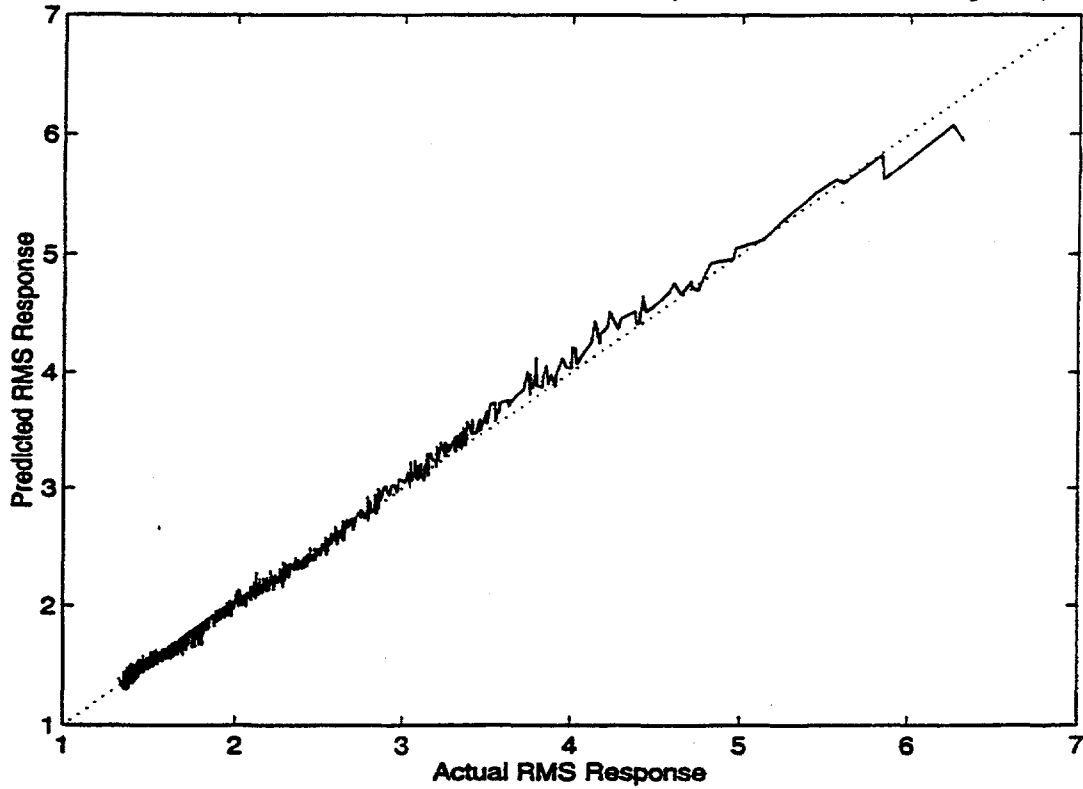


Fig. 6. RMS displacement for 1,002 candidate experiments.

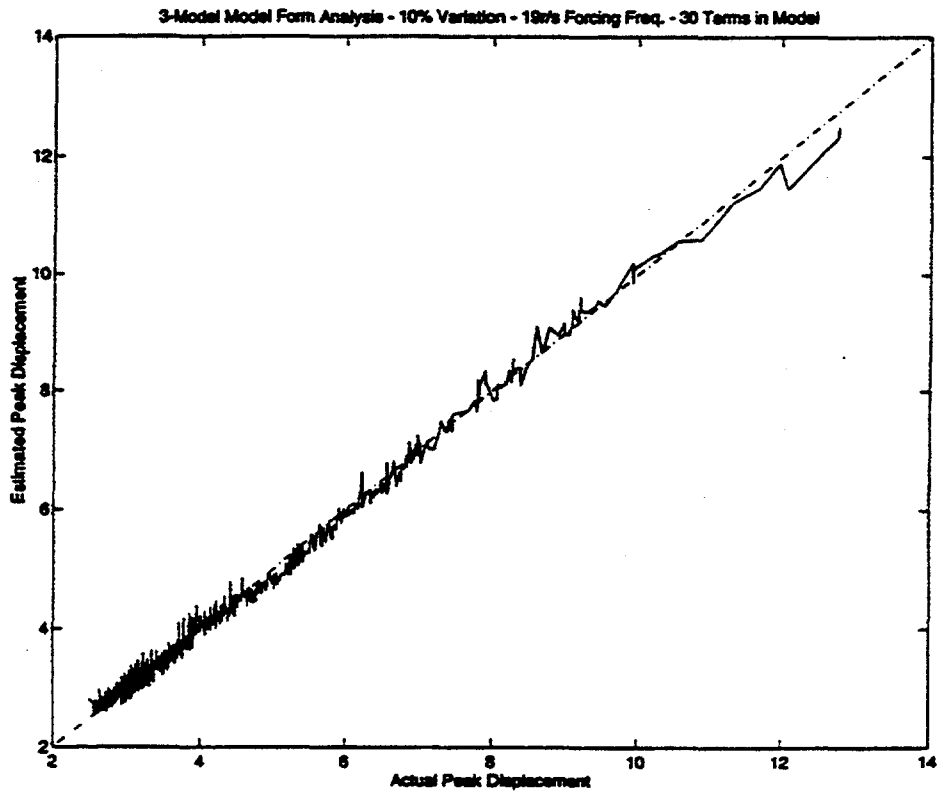


Fig. 7. Peak displacement for 1,002 candidate experiments.

Table 1. Goodness criteria for rms and peak displacement metamodels.

	RMS Displacement	Peak Displacement
R^2	99.57	99.76
R^2_A	99.41	99.63
VIF_{\max}	4.82	10.46
$\text{cond}(X^T X)$	53.76	142.66
e_{\max}	0.379	0.427
h_{\max} (allowable)	0.495 (0.558)	0.618 (0.698)
Cook's D	0.881	0.373
Durbin-Watson D	1.93	2.10

Table 2. Extreme values for random analyses.

Case I

	$\pm 1/100$ Extremes						$\pm 1/40$ Extremes					
	Lower			Upper			Lower			Upper		
	<u>MC</u>	<u>MM</u>	<u>%Err</u>	<u>MC</u>	<u>MM</u>	<u>%Err</u>	<u>MC</u>	<u>MM</u>	<u>%Err</u>	<u>MC</u>	<u>MM</u>	<u>%Err</u>
rms	1.39	1.40	0.70	6.66	6.03	-9.46	1.44	1.45	0.69	4.01	4.13	2.99
peak	2.63	2.67	1.52	13.74	12.33	-10.26	2.71	2.75	1.48	8.30	8.36	0.72

Case II

	$\pm 1/100$ Extremes						$\pm 1/40$ Extremes					
	Lower			Upper			Lower			Upper		
	<u>MC</u>	<u>MM</u>	<u>%Err</u>	<u>MC</u>	<u>MM</u>	<u>%Err</u>	<u>MC</u>	<u>MM</u>	<u>%Err</u>	<u>MC</u>	<u>MM</u>	<u>%Err</u>
rms	1.41	1.42	0.71	7.36	6.38	-13.32	1.45	1.47	1.38	4.35	4.45	2.30
peak	2.66	2.69	1.13	14.98	13.09	-12.62	2.74	2.77	1.09	9.00	9.04	0.44

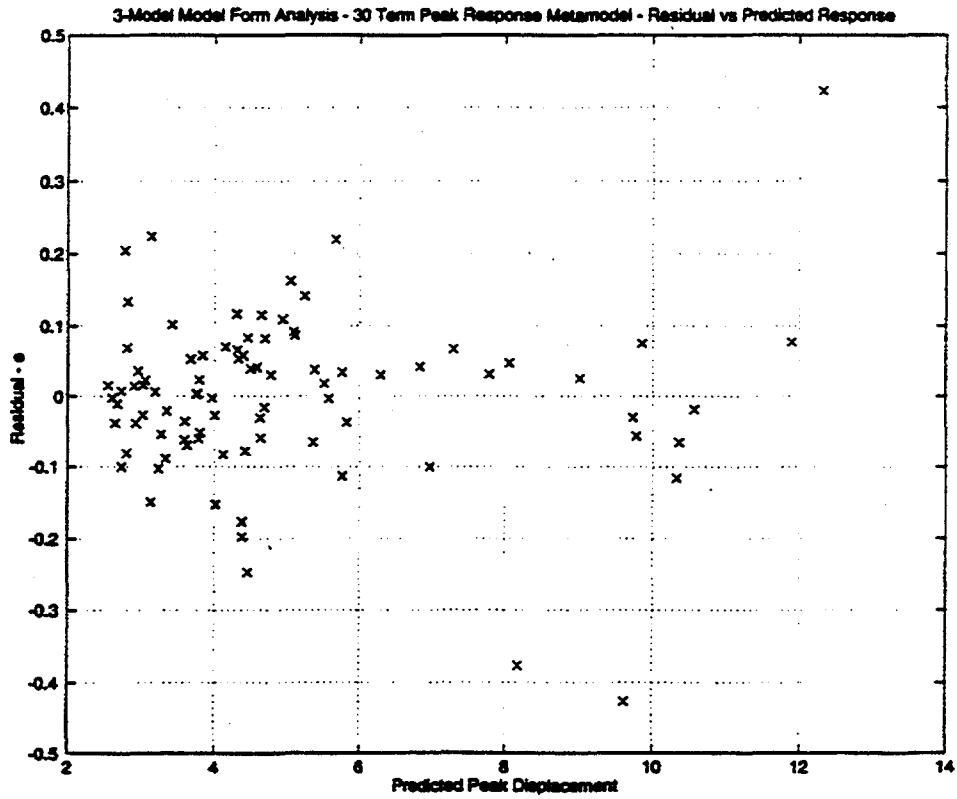


Fig. 8. Peak displacement residual vs. metamodel prediction.

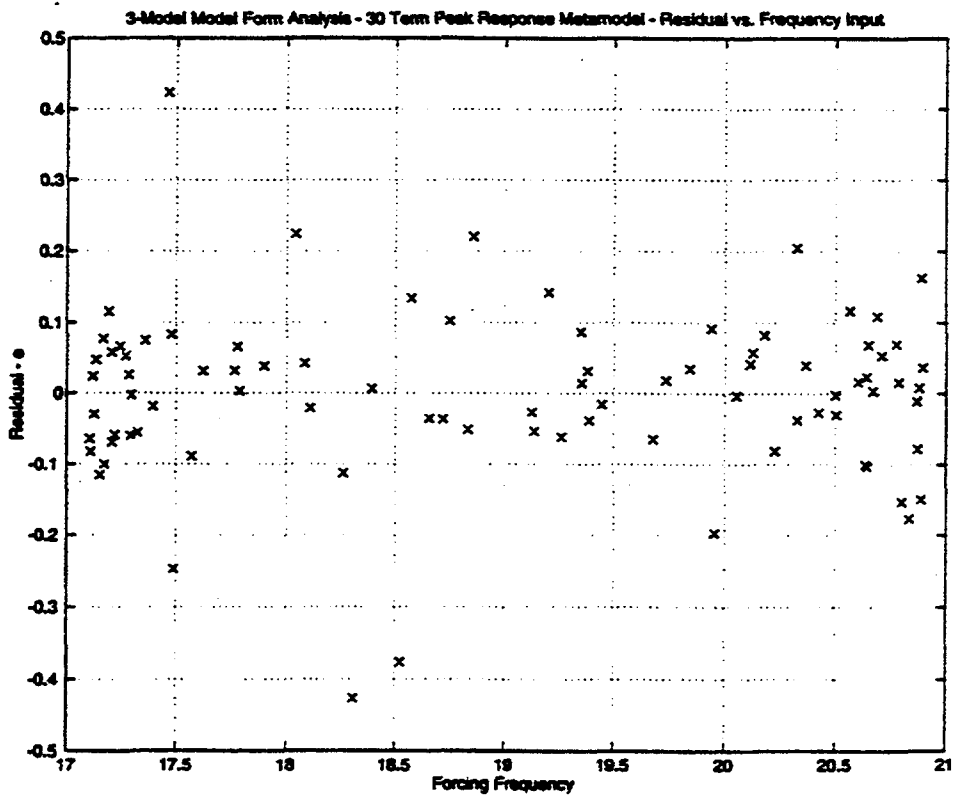


Fig. 9. Peak displacement residual vs. forcing frequency.

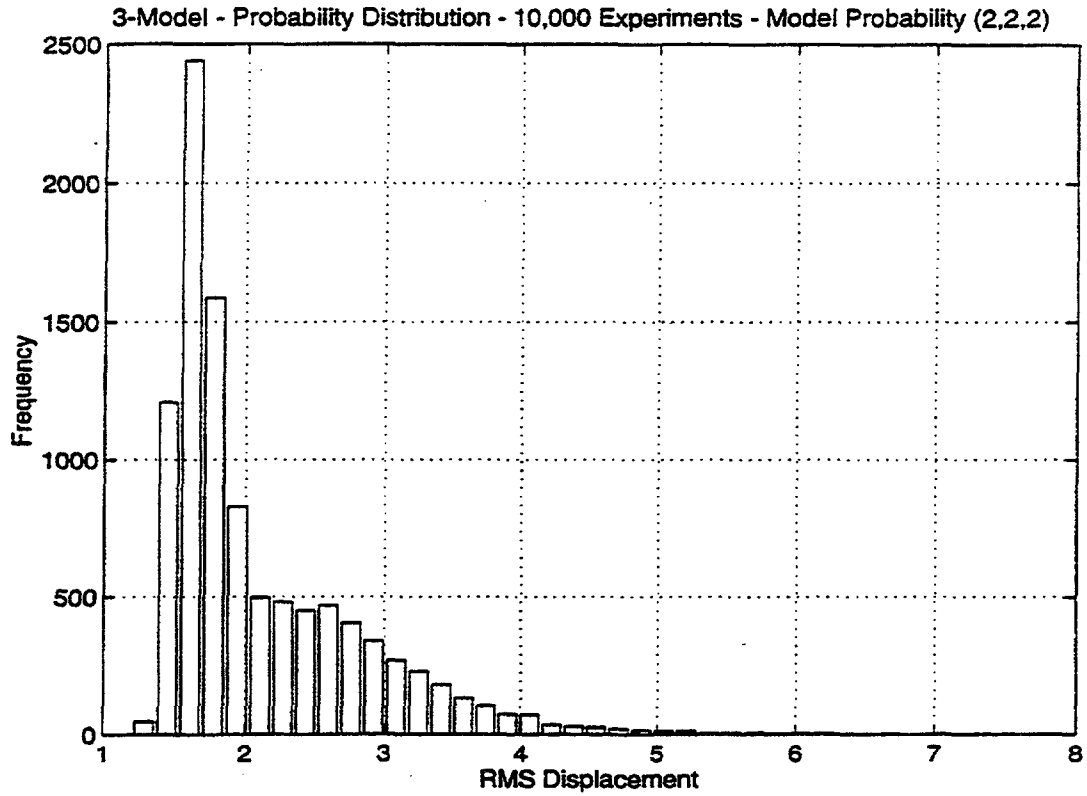


Fig. 10. Frequency distribution for rms displacement - Case I.

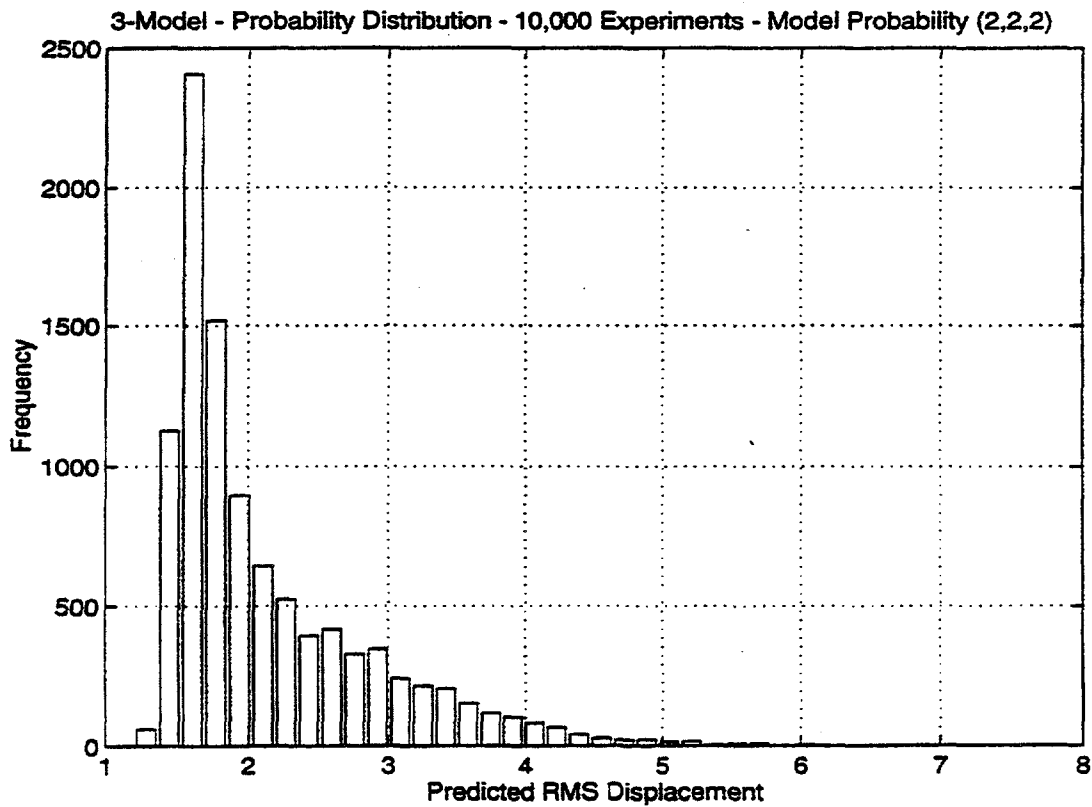


Fig. 11. Frequency distribution for predicted rms displacement - Case I.

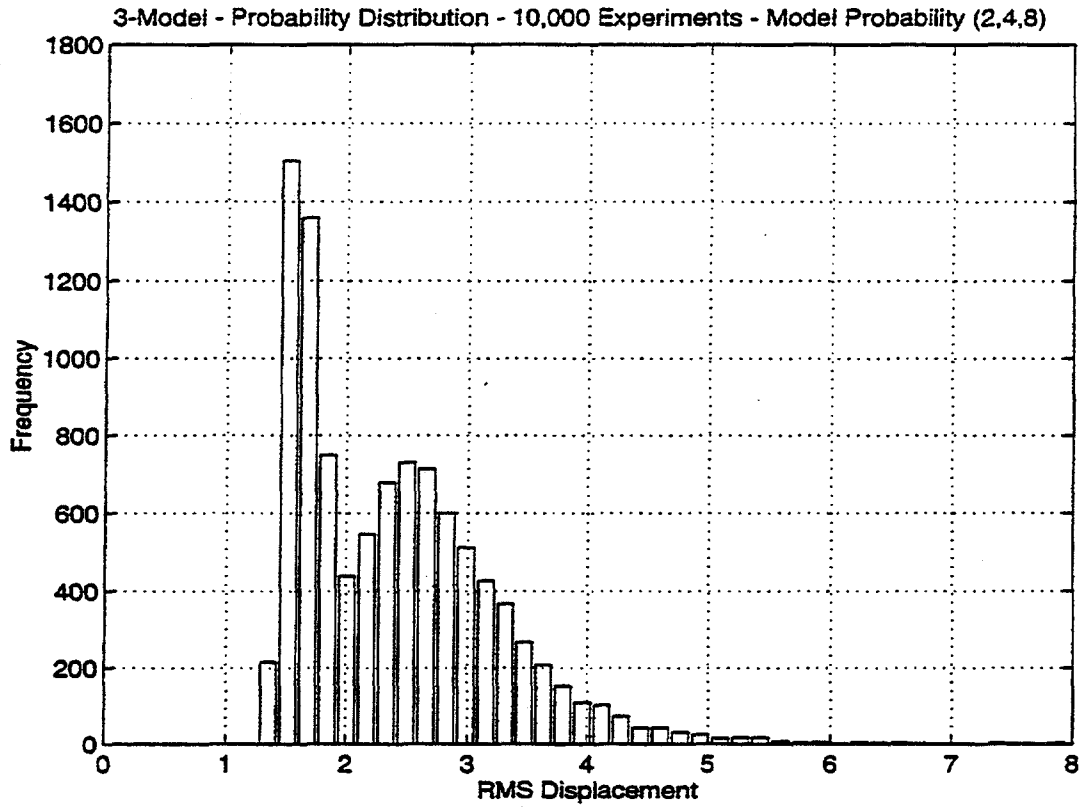


Fig. 12. Frequency distribution for rms displacement - Case II.

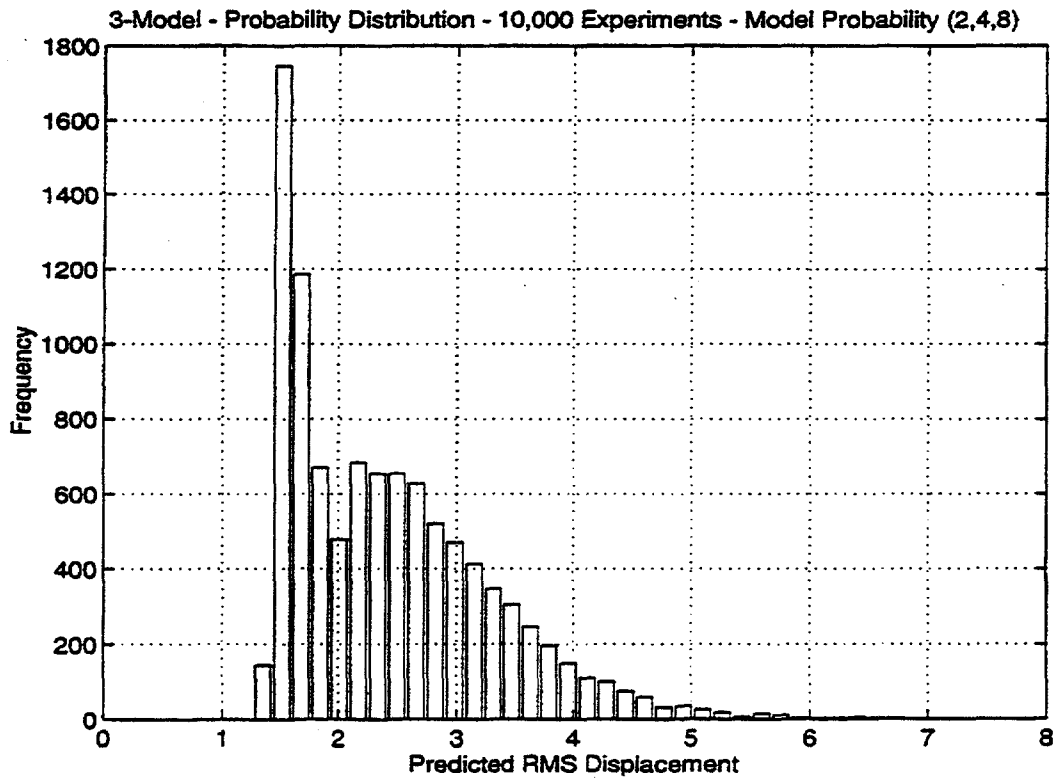


Fig.13. Frequency distribution for predicted rms displacement - Case II.

APPENDIX

Table A-1. MATLAB function for generating candidate experiment design.

```

function [D]=randesign(dum);
%
%   Created by:      Daniel C. Kammer
%                   Associate Professor
%                   Dept of Engineering Mechanics
%                   University of Wisconsin
%                   Madison, WI 53706
%                   (608) 262-5724 / 262-3990
%
%   Generates a coded random candidate design for the study of model form
%   using metamodels.  The last column of the design variable is always
%   assumed to be the qualitative variable representing model form.  The
%   remaining columns are assumed to be quantitative inputs with coded
%   values between -1 and 1.  This program assumes that there are 3 model
%   form levels.  The number of candidate designs must be evenly
%   divisible by the number of model form levels, 3.
%
%
%   HISTORY
%   =====
%
%   Created 08-04-97
%   -----
%
%   Use:  [D]=randesign(dum);
%
%   OUTPUT
%   =====
%   D      =  Experiment design matrix containing coded variables for
%
%             [w m1 k1 m2 k2 ... z]
%
%=====
%
Date=date
%
%   Input number of quantitative variables
%   -----
nv = input('Number of Quantitative Variables? = ')
%
%   Input number of levels for model form
%   -----
%nl = input('Number of Model Form Levels? = ')
%
%   Input of desired candidate experiments
%   -----
ne = input('Number of Desired Experiments? = ')
%
nl=3;                               % number of model form levels
m = ne/nl;
%
t1=clock;                            % start clock
%
D=((2*randperm(ne)')-(ne+1))/ne;      % generate first column
%
for i=1:nv-1                          % loop over quantitative variables
%
D=[D ((2*randperm(ne)')-(ne+1))/ne];% build design matrix for quan. var.
%
end
%
Z = randperm(3);
%

```



```

for i=1:m-1
Z = [Z randperm(3)];
end
%
Z=Z';
%
for i=1:ne
if Z(i)==1;
    Z(i)=-1.0;
elseif Z(i)==2;
    Z(i)=0;
else;
    Z(i)=1.0;
end
end
%
D = [D Z];
%
t2=clock;
Etime=etime(t2,t1)
% clock time duration
%
end

```

Table A-2. MATLAB function for generating experiment design matrix X.

```

function [X]=mod3buildX(d);
%
%   Created by:      Daniel C. Kammer
%                   Associate Professor
%                   Dept of Engineering Mechanics
%                   University of Wisconsin
%                   Madison, WI 53706
%                   (608) 262-5724 / 262-3990
%
%   Builds experiment matrix for 3-Model model form problem
%   -----
%
%   Input
%   =====
%   d   = Experiment design - (nexp x 10)
%
%           [w M1 K1 M2 K2 m1 m2 k1 k2 z]
%
%   Output
%   =====
%   X   = Corresponding experiment coefficient matrix for use with metamodel
%         (nexp x 41)
%
[nexp,m]=size(d);
fr1=.5*(d(:,10)-ones(nexp,1)).*d(:,10);
fr2=ones(nexp,1)-(d(:,10).^2);
fr3=.5*(d(:,10)+ones(nexp,1)).*d(:,10);
%
X=[ones(nexp,1) d(:,1) d(:,1).^2 fr1.*d(:,2) fr1.*d(:,3)...
  fr1.*(d(:,2).^2) fr1.*(d(:,3).^2) fr1.*d(:,2).*d(:,3) fr1.*d(:,2).*d(:,1)...
  fr1.*d(:,3).*d(:,1) fr2.*d(:,4) fr2.*d(:,5) fr2.*(d(:,4).^2)...
  fr2.*(d(:,5).^2) fr2.*d(:,4).*d(:,5) fr2.*d(:,4).*d(:,1)
  fr2.*d(:,5).*d(:,1)...
  fr3.*d(:,6) fr3.*d(:,7) fr3.*d(:,8) fr3.*d(:,9) fr3.*(d(:,6).^2)...
  fr3.*(d(:,7).^2) fr3.*(d(:,8).^2) fr3.*(d(:,9).^2) fr3.*d(:,6).*d(:,7)...
  fr3.*d(:,6).*d(:,8) fr3.*d(:,6).*d(:,9) fr3.*d(:,7).*d(:,8)...
  fr3.*d(:,7).*d(:,9) fr3.*d(:,8).*d(:,9) fr3.*d(:,6).*d(:,1)...
  fr3.*d(:,7).*d(:,1) fr3.*d(:,8).*d(:,1) fr3.*d(:,9).*d(:,1) d(:,10)...
  d(:,10).^2 d(:,10).*d(:,1) (d(:,10).^2).*d(:,1) d(:,10).*d(:,1).^2)...
  (d(:,10).^2).*(d(:,1).^2)];
%
end

```

Table A-3. MATLAB function for reducing experiment design matrix X to suboptimal size.

```

function [DETER,COND,FEXP,DEXP,EFI,FX,M]=xexpdes(X);
%
%   Created by:      Daniel C. Kammer           07-10-97
%                   Associate Professor
%                   Dept of Engineering Physics
%                   University of Wisconsin
%                   Madison, WI 53706
%                   (608) 262-5724 / 262-3990
%
%   Iteratively Generate a Computer Experiment Design using Effective
%   Independence and D-Optimal Design Parameter
%   =====
%
%   HISTORY
%   =====
%
%   =====
%
%   INPUT
%   =====
%   X      = Input initial candidate experiment
%
%   OUTPUT
%   =====
%   DETER = Determinant of information matrix
%   COND  = Condition Number of information matrix
%   FEXP  = Final list of selected experiments
%   DEXP  = List of deleted experiments
%   EFI   = Final Effective independence ranking
%   FX    = Final experiment design
%   M     = Computed D-Optimal design parameter
%
%   Use:   [DETER,COND,FEXP,DEXP,EFI,FX,M]=xexpdes(X);
%   =====
Date=date
%
%   Input number of iterations
%   -----
nit = input('Number of Desired Iterations = ')
%
%   Input number of experiments to be truncated per iteration
%   -----
ntk = input('Number of Experiments to be Truncated Per Iteration = ')
%
%   Calculate Initial Matrix Measures and Ranking
%   -----
t1=clock;
A = X'*X;
FX = X;
[m,p]=size(FX);
nexp=(m:-ntk:m-nit*ntk)';           % number of exps at each iteration
FEXP = (1:m)';                     % Id vector for experiemnts
DETER(1) = det(A);
COND(1) = cond(A);
[P,L]=eig(A);
E=FX*P;
E2=E.^2;
EFD=E2*(inv(L));
ADD=ones(p,1);
EFI=EFD*ADD;
[Y,I]=sort(-EFI);
%

```

```

% Loop over iterations
% -----
for i = 1:nit
    j = i+1;
    format short e
    ONE=ones(m-ntk,1);           % Column of 1's
    i1=cumsum(ONE);            % Running sum
    id=I(i1);                  % Retained dof in ranked order
    iext=sort(id);             % Transform to numerical sort
%
    ONED=ones(ntk,1);          % Column vector of 1's for deleted exps
    i1d=cumsum(ONED) + m-ntk;  % Vector of deleted rows in ORDER
    idd=I(i1d);                % Vector of deleted rows in EFI
    DEXP=[DEXP' FEXP(idd)'];   % Vector of deleted exp id numbers
%
    FX = FX(iext,:);
    FEXP = FEXP(iext);
    A = FX'*FX;
%
    DETER(j) = det(A);
    COND(j) = cond(A);
    [P,L]=eig(A);
    E=FX*P;
    E2=E.^2;
    EFD=E2*inv(L);
    EFI=EFD*ADD;
    [Y,I]=sort(-EFI);
    m=m-ntk;
%
end
%
% Compute D-Optimal Design Parameter
% -----
M = DETER'./(nexp.^p);
%
format bank
t2=clock;
Etime=etime(t2,t1)
%
% Print Results
% -----
format short e
DETER=DETER';
COND=COND';
%
ORDER=-Y;
[YY,II]=sort(I);
format bank
LIST=[YY FEXP(I) ORDER]
format short e
end

```

Table A-4. MATLAB function for fitting metamodel.

```

function [e,Y,Yhat,b,M1,K1,M2,K2,m1,m2,k1,k2,W,t]=mod3afit(D,X);
%
% Created by: Daniel C. Kammer
% Associate Professor
% Dept of Engineering Mechanics
% University of Wisconsin
% Madison, WI 53706
% (608) 262-5724 / 262-3990
%
% Studies uncertainty due to model form for an axial beam with two
% different cross sections. Fits a metamodel to three models of the
% axial beam which is harmonically forced at the endpoint. Model 1
% is a single dof oscillator while model 2 contains two dof with the
% mass and stiffness evenly distributed over both dof. Model 3 has
% two dof with the actual mass and stiffness distributions of the two
% different cross sections of the beam. The metamodel is fully quadratic
% and contains 41 parameters, 86 experiments were chosen from an initial
% set of 1002 using EfI and the D-Optimality criterion.
%
% Updated to a consistent mass lumping representation for all the models
%
% Input Variables
% -----
%
% Global
% -----
% X1 = w - forcing frequency
% z = -1 - model 1
% = 0 - model 2
% = 1 - model 3
%
% Internal to Model 1
% -----
% W1,1 = m1 - oscillator mass
% W1,2 = k1 - oscillator stiffness
%
% Internal to Model 2
% -----
% W2,1 = m2 - oscillator masses
% W2,2 = k2 - oscillator stiffnesses
%
% Internal to Model 3
% -----
% W3,1 = m1 - mass of oscillator 1
% W3,2 = m2 - mass of oscillator 2
% W3,3 = k1 - stiffness of oscillator 1
% W3,4 = k2 - stiffness of oscillator 2
%
% Output Variable
% -----
% Y(:,1) = rms displacement at force input location
% Y(:,2) = peak displacement at force input location
%
% HISTORY
% =====
%
% Created 08-08-97
% -----
%
% Consistent mass lumping added - 8-22-97
% Added peak displacement output - 9-19-97
%
% Use: [e,Y,Yhat,b,M1,K1,M2,K2,m1,m2,k1,k2,W,t]=mod3afit(D,X);

```

```

%
% INPUT
% =====
% D      = Experiment design matrix containing coded variables for
%
%          [w M1 K1 M2 K2 m1 m2 k1 k2 z]
%
% X      = Expanded experiment in coded variables for least squares solution
%
% OUTPUT
% =====
% Y      = Rms displacement predicted by simulation
% Yhat   = Estimated rms displacement using fitted model
% b      = Estimated regression parameters
% M1     = Mass values used during experiment for model 1
% K1     = Stiffness values used during experiment for model 1
% M2     = Mass values used during experiment for model 2
% K2     = Stiffness values used during experiment for model 2
% m1     = Mass value used during experiment for mass 1, model 3
% m2     = Mass value used during experiment for mass 2, model 3
% k1     = Stiffness value used during experiment for spring 1, model 3
% k2     = Stiffness value used during experiment for spring 2, model 3
% W      = Frequency values used during experiment
% e      = Residuals
% =====
%
% Date=date
%
% Input variation of mass in model 1
% -----
dM1 = input('% to Vary Mass in Model 1? = ')
dM1=dM1/100;
%
% Input variation of stiffness in model 1
% -----
dK1 = input('% to Vary Stiffness in Model 1? = ')
dK1=dK1/100;
%
% Input variation of mass in model 2
% -----
dM2 = input('% to Vary Mass in Model 2? = ')
dM2=dM2/100;
%
% Input variation of stiffness in model 2
% -----
dK2 = input('% to Vary Stiffness in Model 2? = ')
dK2=dK2/100;
%
% Input variation of mass 1 in model 3
% -----
dm1 = input('% to Vary Mass 1 in Model 3? = ')
dm1=dm1/100;
%
% Input variation of mass 2 in model 3
% -----
dm2 = input('% to Vary Mass 2 in Model 3? = ')
dm2=dm2/100;
%
% Input variation of spring 1 in model 3
% -----
dk1 = input('% to Vary Spring 1 in Model 3? = ')
dk1=dk1/100;
%
% Input variation of spring 2 in model 3

```

```

% -----
dk2 = input('% to Vary Spring 2 in Model 3? = ')
dk2=dk2/100;
%
% Input variation of forcing frequency
% -----
dw = input('% to Vary Forcing Frequency? = ')
dw=dw/100;
%
% Input number of data points
% -----
nt = input('Number of Data Points in each Experiment = ')
%
t1=clock; % start clock
%
dt=.02; % time step
t=[0:dt:(nt-1)*dt]'; % time vector
%f=inv(dt)*(0:nt-1)'/nt; % frequency vector in Hertz
%
[ne,np]=size(X); % No. of experiments and parameters
%
% Nominal Input Variables
% -----
M10=1; % mass for model 1
K10=100; % stiffness for model 1
M20=0.5; % mass for model 2
K20=200; % stiffness for model 2
m10=2/3; % mass 1 in model 3
m20=1/3; % mass 2 in model 3
k10=300; % spring 1 in model 3
k20=150; % spring 2 in model 3
w0=19; % forcing frequency
%w0=2.
%
dM1=dM1*M10; % delta mass in model 1
dK1=dK1*K10; % delta stiffness in model 1
dM2=dM2*M20; % delta mass in model 2
dK2=dK2*K20; % delta stiffness in model 2
dm1=dm1*m10; % delta mass 1 in model 3
dk1=dk1*k10; % delta spring 1 in model 3
dm2=dm2*m20; % delta mass 2 in model 3
dk2=dk2*k20; % delta spring 2 in model 3
dw=dw*w0; % delta frequency
%
W=w0*ones(ne,1)+dw*D(:,1); % design frequency values
M1=M10*ones(ne,1)+dM1*D(:,2); % design mass values for model 1
K1=K10*ones(ne,1)+dK1*D(:,3); % design stiffness values for model 1
M2=M20*ones(ne,1)+dM2*D(:,4); % design mass values for model 2
K2=K20*ones(ne,1)+dK2*D(:,5); % design stiffness values for model 2
m1=m10*ones(ne,1)+dm1*D(:,6); % design mass 1 values for model 3
k1=k10*ones(ne,1)+dk1*D(:,8); % design spring 1 values for model 3
m2=m20*ones(ne,1)+dm2*D(:,7); % design mass 2 values for model 3
k2=k20*ones(ne,1)+dk2*D(:,9); % design spring 2 values for model 3
Z=D(:,10); % model form
%
for i=1:ne % loop over experiments
%
F=300*sin(W(i)*t); % Input force
%
if Z(i)==-1; % check for model 1
M=M1(i);
K=K1(i);
A=[0 1; -inv(M)*K 0]; % model 1 system matrices
B=[0; inv(M)];

```

```

CC=[1 0];
DD=[0];
x0=[0;0];           % initial conditions for model 1
%
elseif Z(i)==0;     % check for model 2
M=[2*M2(i) 0;0 M2(i)]; % model 2 system matrix
K=[2*K2(i) -K2(i); -K2(i) K2(i)];
A=[zeros(2,2) eye(2); -inv(M)*K zeros(2,2)];
B=[0;0;0;inv(M(2,2))];
CC=[0 1 0 0];
DD=[0];
x0=[0;0;0;0];
%
else;               % model 3
M=[m1(i)+m2(i) 0;0 m2(i)]; % model 3 system matrices
K=[(k1(i)+k2(i)) -k2(i); -k2(i) k2(i)];
A=[zeros(2,2) eye(2); -inv(M)*K zeros(2,2)];
B=[0;0;0;inv(M(2,2))];
CC=[0 1 0 0];
DD=[0];
x0=[0;0;0;0];
end
%
y=lsim(A,B,CC,DD,F,t,x0); % simulate ith experiment
%
rms=sqrt(sum(y.^2)/nt); % rms value
[peak,imax]=max(abs(y(:,1))); % peak response
%
Y(i,1)=rms(:,1); % rms displacement
Y(i,2)=peak; % peak displacement
%
end
%Y=Y';
%
plot(t,y(:,1)) % nominal displacement response
title('Typical Response')
xlabel('Seconds')
ylabel('Displacement')
%
% Estimate Model Parameters using Coded Variables
% -----
[N,p]=size(X);
Condition=cond(X'*X) % information matrix condition number
b=inv(X'*X)*X'*Y; % estimate of regression parameters
Yhat=X*b; % estimate response using fitted model
e=Y-Yhat; % residual
%
% Calculate Statistics
% -----
SSE=diag(e'*e) % sum of squared residuals
s2=SSE/(N-p) % variance of errors
SST=diag(Y'*Y)'-(sum(Y).^2)/N % total sum of squares
SSR=diag(Yhat'*Y)'-(sum(Y).^2)/N % sum of squares due to regression
Fstat=(SSR/(p-1))./(SSE/(N-p)) % F stat for significance of fitted regression
R2=(SSR./SST)*100 % % of variation due to fitted regression
R2a=(1-((1-R2/100)*(N-1)/(N-p)))*100 % adjusted R2 statistic
seb=sqrt(diag(inv(X'*X))*s2); % estimated standard error for parameters
tstat=b./seb; % t stat for null hypothesis of parameters
seb=[(1:np)' seb]
tstat=[(1:np)' tstat]
bhat=[(1:np)' b] % estimated parameters
%
% Compute Variance Inflation Factor

```



```

% -----
g=sqrt(diag(X'*X)).^(-1);
g=diag(g);
x=X*g; % normalize columns of X
[VIF,num]=max(diag(inv(x'*x))) % variance inflation factor
[VIFnob0,num]=...
max(diag(inv(x(:,2:p) '*x(:,2:p)))) % VIF with no B0 term
%
% Maximum Residuals
% -----
[Maxe,num]=max(abs(e)) % max residuals
[Maxepercent,num]=max(100*abs(e./Y)) % max % residuals
%
% Hat Diagonal Statistic
% -----
h=diag(X*inv(X'*X)*X');
hallowable=2*p/N
[hmax,num]=max(h)
%
% Cook's Distance
% -----
r=e./sqrt((1-h)*s2); % studentized residuals
[CDmax,num]=max((h./(1-h))*[1 1]).*(r.^2)/p)
%
% Durbin-Watson D Statistic
% -----
dd1=sum((e(2:N,:)-e(1:(N-1),:)).^2);
dd2=sum(e.^2);
DurbinD=dd1./dd2
%
t2=clock;
Etime=etime(t2,t1) % clock time duration
end

```

Table A-5. MATLAB function for random analysis.

```

function [Y,t,D]=mod3arand(d);
%
%   Created by:      Daniel C. Kammer
%                   Associate Professor
%                   Dept of Engineering Mechanics
%                   University of Wisconsin
%                   Madison, WI 53706
%                   (608) 262-5724 / 262-3990
%
%   Simulates random response for three models for the axial beam with two
%   different cross sections harmonically forced at the endpoint for
%   comparison with results produced by a metamodel. Model 1 is a single dof
%   oscillator while
%   model 2 contains two dof with the mass and stiffness evenly distributed
%   over both dof. Model 3 has two dof with the actual mass and stiffness
%   distributions of the two different cross sections of the beam. The
%   quantitative inputs are assumed to normally distributed. The model form is
%   based upon a probability vector containing P1 entries -1 for model 1, P2
%   entries 0 for model 2 and P3 entries +1 for model 3. P1 through P3 are
%   specified by the user to reflect the relative confidence among the three
%   models.
%
%   Updated to a consistent mass lumping representation for all the models
%
%   Input Variables
%   -----
%
%   Global
%   -----
%   X1 = w      - forcing frequency
%   z = -1      - model 1
%   = 0         - model 2
%   = +1        - model 3
%
%   Internal to Model 1
%   -----
%   W1,1 = m1   - oscillator mass
%   W1,2 = k1   - oscillator stiffness
%
%   Internal to Model 2
%   -----
%   W2,1 = m2   - oscillator masses
%   W2,2 = k2   - oscillator stiffnesses
%
%   Internal to Model 3
%   -----
%   W3,1 = m1   - mass of oscillator 1
%   W3,2 = m2   - mass of oscillator 2
%   W3,3 = k1   - stiffness of oscillator 1
%   W3,4 = k2   - stiffness of oscillator 2
%
%   Output Variable
%   -----
%   Y(:,1) = rms displacement at force input location
%   Y(:,2) = peak displacement at force input location
%
%   HISTORY
%   =====
%
%   Created 10-21-97
%   -----
%
%   Consistent mass lumping added - 8-22-97

```

```

%      Added peak displacement output - 9-19-97
%
%      Use:  [Y,t,D]=mod3arand(d);
%
%
%      INPUT
%      =====
%      d      =  Fractional variation of each input variable used
%                in previous metamodel.
%
%      [dw dM1 dK1 dM2 dK2 dm1 dm2 dk1 dk2]
%
%      OUTPUT
%      =====
%      Y      =  Rms displacement and peak displacement
%      M1     =  Mass values used during experiment for model 1
%      K1     =  Stiffness values used during experiment for model 1
%      M2     =  Mass values used during experiment for model 2
%      K2     =  Stiffness values used during experiment for model 2
%      m1     =  Mass value used during experiment for mass 1, model 3
%      m2     =  Mass value used during experiment for mass 2, model 3
%      k1     =  Stiffness value used during experiment for spring 1, model 3
%      k2     =  Stiffness value used during experiment for spring 2, model 3
%      W      =  Frequency values used during experiment
%      t      =  Time vector for a single experiment
%      D      =  Resulting experiment design matrix containing coded variables for
%
%                [w M1 K1 M2 K2 m1 m2 k1 k2 z]
%      =====
%
Date=date
%
%      Input standard deviation of mass in model 1
%      -----
dM1 = input('% SD of Mass in Model 1? = ')
dM1=dM1/100;
%
%      Input standard deviation of stiffness in model 1
%      -----
dK1 = input('% SD of Stiffness in Model 1? = ')
dK1=dK1/100;
%
%      Input standard deviation of mass in model 2
%      -----
dM2 = input('% SD of Mass in Model 2? = ')
dM2=dM2/100;
%
%      Input standard deviation of stiffness in model 2
%      -----
dK2 = input('% SD of Stiffness in Model 2? = ')
dK2=dK2/100;
%
%      Input standard deviation of mass 1 in model 3
%      -----
dm1 = input('% SD of Mass 1 in Model 3? = ')
dm1=dm1/100;
%
%      Input standard deviation of mass 2 in model 3
%      -----
dm2 = input('% SD of Mass 2 in Model 3? = ')
dm2=dm2/100;
%
%      Input standard deviation of spring 1 in model 3
%      -----

```

```

dk1 = input('% SD of Spring 1 in Model 3? = ')
dk1=dk1/100;
%
% Input standard deviation of spring 2 in model 3
% -----
dk2 = input('% SD of Spring 2 in Model 3? = ')
dk2=dk2/100;
%
% Input standard deviation of forcing frequency
% -----
dw = input('% SD of Forcing Frequency? = ')
dw=dw/100;
%
% Input number of possibilities for model 1
% -----
P1 = input('Number Chances for Model 1? = ')
%
% Input number of possibilities for model 2
% -----
P2 = input('Number Chances for Model 2? = ')
%
% Input number of possibilities for model 3
% -----
P3 = input('Number Chances for Model 3? = ')
%
'Total number in model form pool'
Ptotal = P1+P2+P3
%
'Percentage probability for model 1'
(P1/Ptotal)*100
%
'Percentage probability for model 2'
(P2/Ptotal)*100
%
'Percentage probability for model 3'
(P3/Ptotal)*100
%
% Input number of experiments to run
% -----
ne = input('Number of Experiments to Run = ')
%
% Input number of data points
% -----
nt = input('Number of Data Points in each Experiment = ')
%
t1=clock; % start clock
%
dt=.02; % time step
t=[0:dt:(nt-1)*dt]'; % time vector
%
% Generate Model Form Probability Vector
% -----
PZ=[-1*ones(P1,1); zeros(P2,1); ones(P3,1)];
%
% Nominal Input Variables
% -----
M10=1; % mass for model 1
K10=100; % stiffness for model 1
M20=0.5; % mass for model 2
K20=200; % stiffness for model 2
m10=2/3; % mass 1 in model 3
m20=1/3; % mass 2 in model 3
k10=300; % spring 1 in model 3
k20=150; % spring 2 in model 3

```

```

w0=19; % forcing frequency
%
dM1=dM1*M10; % delta mass in model 1
dK1=dK1*K10; % delta stiffness in model 1
dM2=dM2*M20; % delta mass in model 2
dK2=dK2*K20; % delta stiffness in model 2
dm1=dm1*m10; % delta mass 1 in model 3
dk1=dk1*k10; % delta spring 1 in model 3
dm2=dm2*m20; % delta mass 2 in model 3
dk2=dk2*k20; % delta spring 2 in model 3
dw=dw*w0; % delta frequency
%
for i=1:ne % loop over experiments
%
W=w0+dw*randn(1); % experiment frequency value
M1=M10+dM1*randn(1); % experiment mass values for model 1
K1=K10+dK1*randn(1); % experiment stiffness values for model 1
M2=M20+dM2*randn(1); % experiment mass values for model 2
K2=K20+dK2*randn(1); % experiment stiffness values for model 2
m1=m10+dm1*randn(1); % experiment mass 1 values for model 3
k1=k10+dk1*randn(1); % experiment spring 1 values for model 3
m2=m20+dm2*randn(1); % experiment mass 2 values for model 3
k2=k20+dk2*randn(1); % experiment spring 2 values for model 3
pp=randperm(Ptotal);
Z=PZ(pp(1)); % experiment model form
MM(i)=M1;
%
F=300*sin(W*t); % Input force
%
if Z==-1; % check for model 1
M=M1;
K=K1;
A=[0 1; -inv(M)*K 0]; % model 1 system matrices
B=[0; inv(M)];
CC=[1 0];
DD=[0];
x0=[0;0]; % initial conditions for model 1
%
elseif Z==0; % check for model 2
M=[2*M2 0;0 M2]; % model 2 system matrix
K=[2*K2 -K2; -K2 K2];
A=[zeros(2,2) eye(2); -inv(M)*K zeros(2,2)];
B=[0;0;0;inv(M(2,2))];
CC=[0 1 0 0];
DD=[0];
x0=[0;0;0;0];
%
else; % model 3
M=[m1+m2 0;0 m2]; % model 3 system matrices
K=[(k1+k2) -k2; -k2 k2];
A=[zeros(2,2) eye(2); -inv(M)*K zeros(2,2)];
B=[0;0;0;inv(M(2,2))];
CC=[0 1 0 0];
DD=[0];
x0=[0;0;0;0];
end
%
y=lsim(A,B,CC,DD,F,t,x0); % simulate ith experiment
%
rms=sqrt(sum(y.^2)/nt); % rms value
[peak,imax]=max(abs(y(:,1))); % peak response
%
Y(i,1)=rms(:,1); % rms displacement
Y(i,2)=peak; % peak displacement

```

```

%
% Transform Random Inputs to Coded Variables
% -----
D(i,1)=(W-w0)/(d(1)*w0);           % experiment frequency value
D(i,2)=(M1-M10)/(d(2)*M10);       % experiment mass values for model 1
D(i,3)=(K1-K10)/(d(3)*K10);       % experiment stiffness values for model 1
D(i,4)=(M2-M20)/(d(4)*M20);       % experiment mass values for model 2
D(i,5)=(K2-K20)/(d(5)*K20);       % experiment stiffness values for model 2
D(i,6)=(m1-m10)/(d(6)*m10);       % experiment mass 1 values for model 3
D(i,7)=(m2-m20)/(d(7)*m20);       % experiment mass 2 values for model 3
D(i,8)=(k1-k10)/(d(8)*k10);       % experiment spring 1 values for model 3
D(i,9)=(k2-k20)/(d(9)*k20);       % experiment spring 2 values for model 3
D(i,10)=Z;
%
end
%
plot(t,y(:,1))                     % nominal displacement response
title('Typical Response')
xlabel('Seconds')
ylabel('Displacement')
%
%
t2=clock;
Etime=etime(t2,t1)                 % clock time duration
%
end

```

Intentionally Left Blank

APPENDIX E:

Combining Metamodels with Rational Function Representations of Discretization Error for Uncertainty Quantification

**submitted to International Journal for Numerical
Methods in Engineering**

Intentionally Left Blank

Combining Metamodels with Rational Function Representations of Discretization Error for Uncertainty Quantification

Daniel C. Kammer
Department of Engineering Physics
University of Wisconsin
1500 Engineering Dr., Madison, WI 53706

Kenneth F. Alvin
Structural Dynamics and Vibration Control
Sandia National Laboratories

David S. Malkus
Department of Engineering Physics
University of Wisconsin

Abstract

A new method is presented for extending metamodeling techniques to include the effects of finite element model mesh discretization errors. The method employs a rational function representation of the discretization error rather than the power series representation used by methods such as Richardson extrapolation. Examples dealing with simple function estimation and estimation of the vibrational frequency of a one dimensional bar showed that when extrapolated to the continuum, the rational function model gave more accurate estimates using fewer terms than the Richardson extrapolation technique. This is an important consideration for computational reliability assessment of high consequence systems, as small biases in solutions can significantly affect the accuracy of small magnitude probability estimates. In general, the rational function form of the discretization error produces a nonlinear model requiring an iterative nonlinear least-squares solution technique. However, all the examples studied in this work proved to be close-to-linear, meaning that the linear least-squares estimate of the model coefficients could not be improved. In subsequent nondeterministic analyses, the rational function based metamodel also produced more accurate estimates of failure probabilities using fewer terms than the Richardson extrapolation method under very severe extrapolation conditions. Rational function representations of discretization error offer greater flexibility by allowing a user to accurately extrapolate to a continuum representation from numerical experiments performed outside the asymptotic region where the usual power series representation is not converging. This allows the use of coarser meshes in the numerical experiments, saving a significant amount of time and effort.

1. Introduction

Numerical simulations of system response are used throughout the research and industrial communities to help make critical system design and performance decisions. These simulations contain errors due to known simplifications, variabilities due to random inputs with known statistics, and uncertainties due to lack of knowledge [1]. In order to make proper decisions, it is important to understand how all three of these types of deviations propagate through simulations into computed results. A particularly important source of error, variability, and uncertainty is the analytical model of the system being simulated.

Model parameter variability has received the most attention in the literature. Assuming that the variabilities in model parameters follow known probability distributions, the ultimate goal of the quantification of their effects is to determine the corresponding probability distributions of the simulation outputs of interest. Accurate prediction of these distributions requires Monte Carlo analysis incorporating hundreds of thousands of simulation runs. In most cases, the computational cost is prohibitive. In many situations, the computational effort can be greatly reduced by generating what is commonly referred to as a *Response Surface* or *Metamodel*. An appropriate functional form is assumed for the simulation response of interest and then fit to the output produced by the simulation of a set of carefully selected numerical experiments. The functional form is usually taken as a quadratic polynomial in the input variables.

In contrast with the full numerical simulation, evaluation of the metamodel is very fast. Assuming that the underlying system is itself deterministic, the prediction of the response can be decoupled from the probabilistic Monte Carlo sampling using the metamodel. This technique has been referred to as "decoupled" Monte Carlo analysis [2]. The metamodel approach has long been used in the biological, physical, and sociological sciences [3], but has more recently received attention from the structural dynamics community. Applications include structural optimization [4, 5] and probabilistic design [6-8]. Response surfaces based on finite element / lattice sampling have been investigated by Romero and Bankston [2].

While most of the work mentioned has focused on the effects of model parameter variability, Alvin [9] recently incorporated mesh discretization error into the metamodel formulation in an effort to quantify its effect on nondeterministic analysis. The approach is based upon the classical Richardson extrapolation [10] technique used in finite element modeling to quantify the error due to spatial discretization inherent in a mesh. Discretization error is modeled as a polynomial in parameter h , which could, for example, represent a characteristic element length. The method strongly resembles the metamodel approach itself, making it very attractive to combine the two into a single formulation. In general, a metamodel is used to predict desired responses by interpolating among selected experiment design points. In contrast, the idea behind including the discretization parameter is to eventually extrapolate to the value of $h = 0$, obtaining a metamodel representing the

continuous system. The discretization error is then theoretically removed from the subsequent decoupled Monte Carlo analysis. In this way, the effects of bias error due to discretization on the nondeterministic analysis can be quantified. Alvin demonstrated the efficacy of this approach for a simple bar example using three different mesh refinements [9]. One conclusion drawn from Alvin's work is that small failure probability estimates, which are typical in high consequence systems such as nuclear reactors, can be very sensitive to small bias errors in computational solutions. Hence, estimating the potential effects of discretization error is important in computational reliability analysis.

In order for Richardson extrapolation to produce accurate results, the numerical experiments must be performed within the asymptotic region of the power series being used to model the discretization error. This simply means that the term or terms retained from the power series dominate the terms that are omitted. The implication is that even the coarse meshes used in the numerical experiments must be sufficiently refined. The work presented in this paper considers an alternative representation of the discretization error in the form of a rational function or ratio of polynomials in the parameter h . In general, rational functions are much more flexible than polynomials for approximating mathematical functions. They are capable of representing a wide variety of ascending and descending curves, including curves with minima and maxima, and curves that approach asymptotes. Due to this greater flexibility, it has been found that the use of rational functions in general response surface analysis can produce metamodels that are more accurate and require fewer coefficients than the corresponding polynomial representations [11]. Minimizing the number of coefficients is important to avoid systematic error. The increased flexibility of rational functions over power series is also reflected in the observed superiority of rational functions in extrapolation applications [12]. A rational function approximation of an analytic function can remain accurate even after all the terms in a power series representation in h have similar magnitudes making the idea of "method order" meaningless [13]. It is believed that this attribute will allow the use of coarser meshes in the numerical experiments, which would save a significant amount of time and effort.

2. Theory

In metamodel analyses, parsimony is usually preferred over generality, therefore, quadratic metamodels are assumed in the form of simple polynomials in the input variables. This same approach is used here, but the formulation can be easily generalized to higher order surfaces and more sophisticated basis functions if there is a need. Assuming a single output y for the simulation, such as a peak displacement or stress, the general form of the quadratic metamodel is given by

$$y(x, \beta) = \beta_0 + \sum_{i=1}^l \beta_i x_i + \sum_{i=1}^l \sum_{j=i}^l \beta_{ij} x_i x_j \quad (1)$$

in which x_i are usually quantitative input variables, like mass, spring stiffness, elastic modulus, etc., and β_i are constant coefficients to be determined. Quantitative variables are usually coded such that they vary continuously between -1.0 and 1.0 using the relation

$$x_i = \frac{X_i - X_{io}}{\Delta_i} \quad \Delta_i = \frac{1}{2}(X_{i\max} - X_{i\min}) \quad (2)$$

where x_i is the i th coded input, X_i is the corresponding uncoded input, X_{io} is the nominal value of the i th input, and $X_{i\min}$ and $X_{i\max}$ represent the lower and upper bounds on the range of the i th uncoded input. The nominal value of the coded input variable always corresponds to 0.0. Coding is performed to offset large possible differences in units between various model inputs, which can cause numerical error during fitting of the metamodel.

Spatial discretization error is fundamental to all finite element models as well as other types of analytical model representations [14]. Richardson extrapolation is a technique that attempts to account for this by modeling the error as a low order polynomial in the discretization parameter h of the form

$$y_h = y + \alpha h^q + O(h^{q+1}) \quad (3)$$

where y_h is the numerical solution for the current value of h , y is the continuum solution, α is an unknown coefficient, and q is defined as the formal order of the method. In its classic application, h is assumed to be sufficiently small, i.e. in the asymptotic range, such that the leading term in Eq. (3) will dominate and the higher order terms can be omitted. More general polynomial extrapolation methods often use an even asymptotic expansion in h for the error

$$y_h - y = \alpha_1 h^2 + \alpha_0 h^4 + \dots + \alpha_m h^{2m} \quad (4)$$

It can be shown that for the same number of terms, an even powered asymptotic expansion will converge twice as quickly in extrapolation applications as a series with powers increasing by one [12].

2.1 Combined Metamodel Forms

Assuming a multiplicative discretization effect, the metamodel for the continuum approximation, given by Eq. (1), can be combined with the asymptotic expansion of Eq. (4) to produce an extended metamodel that explicitly includes the effects of discretization in the form

$$y_h = y(x, \beta) [1 + \alpha_1 h^2 + \alpha_2 h^4 + \dots + \alpha_m h^{2m}] \quad (5)$$

This form allows coupling between the discretization parameter h and the continuum form of the metamodel $y(x, \beta)$ and is analogous to that used by Alvin [9]. It is believed that improved convergence properties can be obtained by alternatively using a rational function approximation for the discretization error, producing the metamodel form

$$y_h = y(x, \beta) \frac{1 + N(\alpha_i, h)}{1 + D(\gamma_i, h)} = y(x, \beta) \frac{[1 + \alpha_1 h^2 + \alpha_2 h^4 + \dots + \alpha_m h^{2m}]}{[1 + \gamma_1 h^2 + \gamma_2 h^4 + \dots + \gamma_n h^{2n}]} \quad (6)$$

where, in general, $m \leq n$. The corresponding rational function (RF) representation of the discretization error, $y_h - y$, is given by

$$\varepsilon_{hRF} = y(x, \beta) \frac{N(\alpha_i, h) - D(\alpha_i, h)}{1 + D(\gamma_i, h)} \quad (7)$$

A specific example can be considered in order to clarify and compare the two discretization error model forms. Three quantitative inputs are considered, x_1 , x_2 , and x_3 . The fourth model input is the discretization parameter h . The truncated metamodel considered for this application containing the polynomial form of the discretization error is given by

$$y_{hRE} = y(x, \beta) [1 + \alpha_1 h^2 + \alpha_2 h^4] \quad (8)$$

Retaining only the linear coupling between the quantitative variables x_i and h^2 , the expanded form becomes

$$\begin{aligned} y_{hRE} = & \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \\ & \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \alpha_1 h^2 + \alpha_{11} x_1 h^2 + \alpha_{12} x_2 h^2 + \\ & \alpha_{13} x_3 h^2 + \alpha_2 h^4 \end{aligned} \quad (9)$$

Additional terms from the general form in Eq. (5) can be easily included in the analysis. Equation (9) will be called the Richardson extrapolation (RE) metamodel in the sequel. In order to calculate the metamodel coefficients in Eq. (9), a sequence of simulations, or numerical experiments, is performed using predetermined settings for the input variables. The data for the complete set of experiments is then combined into the matrix equation

$$X\theta = Y \quad (10)$$

in which X is the experiment design matrix, θ is a column vector containing the model coefficients, and Y is a column vector containing the responses from the numerical experiments. An estimate of the model coefficients is generated using linear least-squares (LS)

$$\hat{\theta} = [X^T X]^{-1} X^T Y \quad (11)$$

There are many criteria that can be used to optimally design numerical experiments [15]. For example, the well-known Box-Behnken experiment design [8] for the case of four model inputs is presented in Table 1. The coded values of the three quantitative inputs x_i are -1 , 0 , and 1 , representing low, nominal, and high levels, respectively. The nominal value of the inputs corresponds to their expected values, while the high and low levels represent plus and minus two standard deviations. In the case of a finite element analysis, the values -1 , 0 , and 1 corresponding to the discretization variable h represent coarse, nominal, and fine meshes.

The truncated metamodel containing the rational function form of the discretization error considered in this paper is given by

$$y_{hRF} = y(x, \beta) \left[\frac{1 + \alpha_1 h^2}{1 + \gamma_1 h^2 + \gamma_2 h^4} \right] \quad (12)$$

or in its expanded form

$$\begin{aligned} y_{hRF} = & [\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \\ & \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \alpha_1 h^2 + \alpha_{11} x_1 h^2 + \alpha_{12} x_2 h^2 + \\ & \alpha_{13} x_3 h^2] \left[\frac{1}{1 + \gamma_1 h^2 + \gamma_2 h^4} \right] \end{aligned} \quad (13)$$

Only quadratic terms in h have been retained in the numerator of the discretization error model, while a quartic term is included in the denominator. As in the case of the RE metamodel, only

linear coupling between the quantitative variables x_i and h^2 is retained. Additional terms in the numerator and denominator can be easily included if required. The metamodel represented in Eqs. (12) or (13) is nonlinear in the coefficients γ_1 and γ_2 , meaning that an iterative nonlinear least-squares technique, such as the Marquardt method [16], must be used to estimate the coefficients. In the case of a linear model, such as Eq. (9), if the errors are assumed to be independent and normally distributed, with zero mean and constant variance, LS produces estimates of the model parameters which are unbiased, normally distributed, and minimum variance. In practice, least-squares produces the best available estimates. Other desirable properties of linear models include ease of obtaining the LS estimates, i.e. no iteration, straightforward statistical interpretation of the estimates, and the predicted values of the response variable y will be unbiased. In contrast, LS estimates for a nonlinear model achieve these properties only asymptotically. However, if the LS estimator for the parameters of a nonlinear model is only slightly biased, with a distribution that is close to normal, and has variance close to the minimum, the model can be considered as close-to-linear [17].

Fortunately, the nonlinear model form in Eq. (12) is found to be close-to-linear. It is important to understand that this does not mean that the coefficients that appear nonlinearly, γ_0 and γ_1 , are necessarily small, but rather that the rational form will have properties close to those discussed for a linear model. An iterative technique for nonlinear LS requires an initial guess for the model coefficients θ . A good initial guess is provided by the linear LS estimate, where the nonlinear estimation problem associated with the RF metamodel is cast in a linear form

$$X_{RF}\theta_{RF} = Y \quad (14)$$

by multiplying through Eq. (13) by the denominator of the discretization error term and then subtracting $Y\gamma_1h^2 + Y\gamma_2h^4$ from both sides of the resulting expression. The j th row of the corresponding experiment design matrix is given by

$$X_{RFj} = [1 \quad x_1 \quad x_2 \quad x_3 \quad x_1^2 \quad x_2^2 \quad x_3^2 \quad x_1x_2 \quad x_1x_3 \quad x_2x_3 \quad h^2 \quad x_1h^2 \quad x_2h^2 \quad x_3h^2 \quad -Y_jh^2 \quad -Y_jh^4] \quad (15)$$

where the inputs and the response Y_j are evaluated at the j th experiment in Table 1. The corresponding coefficient vector is given by

$$\theta_{RF} = \{\beta_0 \quad \beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_{11} \quad \beta_{22} \quad \beta_{33} \quad \beta_{12} \quad \beta_{13} \quad \beta_{23} \quad \alpha_1 \quad \alpha_{11} \quad \alpha_{12} \quad \alpha_{13} \quad \gamma_1 \quad \gamma_2\}^T \quad (16)$$

If the RF metamodel described by Eq. (12) behaves as though it is close-to-linear, the optimization scheme used by the nonlinear LS method will find the minimum of the sum of squared errors function in one step. In all of the examples considered during this work, when the linear LS estimate was used as a starting point, convergence took just one step and the optimum estimate essentially coincided with the original starting point. This indicates that the RF metamodel behaves linearly and the best estimate is produced by linear LS. In general, every example should be checked for linearity when using RF metamodels.

2.2 Error Analysis

When working with finite sequences to approximate functions, it is desirable to be able to compute a rate of convergence. For a truncated sequence of simple polynomials, the rate of convergence is straightforward to estimate. In the case of the general RE metamodel

$$y_{hRE}^m = y(x, \beta) [1 + \alpha_1 h^2 + \alpha_2 h^4 + \dots + \alpha_m h^{2m}] \quad (17)$$

the rate would be $O(h^{2(m+1)})$ or at least as fast as the term $h^{2(m+1)}$ for small h . In the case of a RF metamodel, the truncated sequence is a little more difficult to identify. Due to the nonzero polynomial in the denominator, Eq. (6) is equivalent to a sum of infinite sequences of simple polynomials given by

$$y_{hRF}^n = y(x, \beta) [1 + N(\alpha_i, h)] \sum_{i=1}^n S_i \quad (18)$$

where

$$S_1 = 1 + \delta_{11} h^2 + \delta_{12} h^4 + \delta_{13} h^6 + \dots$$

$$S_i = \delta_{ii} h^{2i} + \delta_{ii+1} h^{2(i+1)} + \delta_{ii+2} h^{2(i+2)} + \dots, \quad i > 1$$

The order of convergence can be estimated by computing the difference between a rational function representation containing an infinite number of terms in the denominator, y_{hRF}^∞ , and the truncated expression in Eq. (6) containing terms through order h^{2n} . Equation (18) produces the result

$$y_{hRF}^\infty - y_{hRF}^n = y(x, \beta) [1 + N(\alpha_i, h)] [S_{n+1} + S_{n+2} + \dots] \quad (19)$$

Therefore, assuming $n > m$, the representation y_{hRF}^n converges at least as fast as $h^{2(n+1)}$.

In order to have confidence in the accuracy of extrapolated results using either the RE or RF error representations, consistency checks must be computed. For conciseness, only first and second order models in h^2 will be considered, but the following consistency checks can be generalized to models of any order. Let ε_h^1 and ε_h^2 represent the discretization error produced by truncating either the RE or RF representations to first and second order in h^2 , respectively. In the case of RE representations, Malkus and Webster [18] proposed a check of the relative contributions at first and second order given by the measure

$$r_1 = \frac{\varepsilon_h^2 - \varepsilon_h^1}{\varepsilon_h^1} \quad (20)$$

For the RE models proposed in Eqs. (4) and (5), the measure has the form

$$r_{1RE} = \frac{\alpha_2 h^2}{\alpha_1} \quad (21)$$

In the case of RF models of the form given in Eq. (12), ε_h^1 and ε_h^2 are given by

$$\varepsilon_{hRF}^1 = y \frac{(\alpha_1 - \gamma_1)h^2}{1 + \gamma_1 h^2} \quad (22)$$

$$\varepsilon_{hRF}^2 = y \frac{[(\alpha_1 - \gamma_1)h^2 - \gamma_2 h^4]}{1 + \gamma_1 h^2 + \gamma_2 h^4} \quad (23)$$

which produces

$$r_{1RF} = \frac{-\gamma_2 h^2 (1 + \alpha_1 h^2)}{(\alpha_1 - \gamma_1) [1 + \gamma_1 h^2 + \gamma_2 h^4]} \quad (24)$$

A second measure, proposed by Conte and de Boor [19], estimates the dominance of that portion of the series retained in the discretization error model over that portion discarded. Assuming that the desired output value is available at three different discretizations, the measure is computed using the relation

$$r_2 = \frac{y_1 - y_2}{y_2 - y_3} \quad (25)$$

in which y_i denotes the value produced at a discretization parameter value h_i . In a finite element analysis, h_i would correspond to three different mesh refinements. In the following analysis, it is assumed that h_1 corresponds to the coarsest discretization, with $h_2 = h_1/q_2$ and $h_3 = h_1/q_3$ in which q_2 and q_3 are integers. Assuming a second order RE metamodel in the form of Eq. (8), an estimate of measure r_2 can be computed by substituting into Eq. (25)

$$\hat{r}_{2RE}^2 = \frac{\alpha_1 \left(1 - \frac{1}{q_2^2}\right) + \alpha_2 h_1^2 \left(1 - \frac{1}{q_2^4}\right)}{\alpha_1 \left(\frac{1}{q_2^2} - \frac{1}{q_3^2}\right) + \alpha_2 h_1^2 \left(\frac{1}{q_2^4} - \frac{1}{q_3^4}\right)} \quad (26)$$

The corresponding measure for a first order model, \hat{r}_{2RE}^1 , can be found by setting α_2 equal to zero. For the second order RF metamodel in Eq. (12), the corresponding estimate is given by

$$\hat{r}_{2RF}^2 = \frac{\left[\gamma_1 \left(1 - \frac{1}{q_2^2}\right) + \gamma_2 h_1^2 \left(1 - \frac{1}{q_2^4}\right) \right] \left(1 + \gamma_1 \frac{1}{q_3^2} h_1^2 + \gamma_2 \frac{1}{q_3^4} h_1^4\right)}{\left[\gamma_1 \left(\frac{1}{q_2^2} - \frac{1}{q_3^2}\right) + \gamma_2 h_1^2 \left(\frac{1}{q_2^4} - \frac{1}{q_3^4}\right) \right] \left(1 + \gamma_1 h_1^2 + \gamma_2 h_1^4\right)} \quad (27)$$

Setting γ_2 equal to zero produces the measure for the corresponding first order RF model, \hat{r}_{2RF}^1 .

These consistency measures can be used in a fashion analogous to that proposed by Malkus and Webster [18]. It will be considered strong evidence that either of the methods is first order in h^2 and in its asymptotic range if r_1 is small with respect to 1 and $\hat{r}_2^1 \approx r_2$. Conversely, if r_1 is large and $\hat{r}_2^2 \approx r_2$, it will be considered strong evidence that the method is second order in h^2 and in its asymptotic range. Otherwise, it will be concluded that the analysis is indeterminate, meaning that the methods are not in the asymptotic range or possibly of higher order.

3. Numerical Examples

Several numerical examples are considered in this paper. Initially, a very simple problem is presented to illustrate the superior convergence properties of the rational function representation of the discretization error model. The second example deals with the prediction of the natural frequency of the third axial mode of vibration of a one dimensional prismatic bar which is constrained at one end.

3.1 Convergence Study for Simple Example

The function

$$y_h = 3 \cdot \left(\cos \frac{\pi}{2} x \right) e^{h^2} \quad (28)$$

will be considered as a simple representation of the possible effects of discretization on a system response. This might represent the solution produced by a finite element analysis. The extended metamodel formulations discussed in the previous section will be applied to estimate y_h and then extrapolated to their continuum estimates at $h = 0$. Quantitative variable x varies between -1 and 1 while the discretization variable h will take on three different values in the numerical experiments representing different mesh sizes. Thirty numerical experiments were generated by randomly varying x according to a modified Latin Hypercube centered at zero. The corresponding value of h was randomly selected from the three possible values used for the analysis.

The metamodel forms applied in this example are given by

$$y_{hRE} = y(x, \beta) [1 + \alpha_1 h^2 + \alpha_2 h^4] \quad (29)$$

$$y_{hRF} = y(x, \beta) \left[\frac{1}{1 + \gamma_1 h^2 + \gamma_2 h^4} \right]$$

Note that the RF metamodel has no polynomial present in its numerator. This form was selected such that both of the metamodels in (29) have the same number of unknown coefficients in the discretization error, allowing direct comparison of the models. This form of the RF model is sometimes referred to as an Inverse Polynomial [20]. When expanded, the metamodels in Eq. (29) have the forms

$$y_{hRE} = \beta_0 + \beta_1 x + \beta_2 x^2 + \alpha_1 h^2 + \alpha_{11} x h^2 + \alpha_{12} x^2 h^2 + \alpha_2 h^4 \quad (30)$$

$$y_{hRF} = \beta_0 + \beta_1 x + \beta_2 x^2 - \gamma_1 y_h h^2 - \gamma_2 y_h h^4$$

Note that both models are second order in h^2 , but the RE model possesses two more unknown coefficients than the RF metamodel due to the explicit coupling between the quantitative variable x and h^2 . In contrast, the RF model form in Eq. (29) possesses this type of coupling without explicitly introducing new coefficients.

Two cases are considered for this example. In the first, the values of the discretization parameter are selected as $h_1 = 1.2$, $h_2 = 0.6$, $h_3 = 0.3$. This corresponds to the usual practice of doubling the mesh size between refinement analyses. Values of h were selected such that the numerical experiments were not performed in what would be expected as the asymptotic region.

Parameters q_2 and q_3 have values 2 and 4, respectively. Both first and second order metamodels in h^2 are considered. The first order models are derived from Eq. (29) by simply omitting terms proportional to h^4 . Linear LS was used to estimate the RF and RE model coefficients. The estimate of the leading coefficient β_0 , corresponding to the response at nominal x and zero mesh size, serves as a measure of the ability of the associated metamodel to accurately extrapolate to the continuum value of 3.0 at $h = 0$. Analysis results are presented in Table 2.

A first order RF model produces the extrapolated result $\hat{\beta}_0 = 3.23$ possessing an error of 7.67%. In general, the true value of 3.00 is not known, therefore, the consistency measures r_1 and r_2 must be applied to validate the accuracy of the extrapolated results. Note that r_1 cannot be computed until a second order metamodel is generated. However, r_2 can be immediately calculated using Eq. (25) resulting in the value 8.22. An estimate, $\hat{r}_2 = 14.59$, can be computed from the first order RF metamodel using Eq. (27). A lack of agreement between r_2 and its estimate, indicates that the result extrapolated from the first order RF model cannot be assumed to be accurate. A first order RE metamodel produced the extrapolated result $\hat{\beta}_0 = 2.33$ having an error of -22.33%. Equation (26) gives the corresponding estimate $\hat{r}_2 = 4.00$. Measure r_2 indicates that the extrapolated result cannot be accepted as an accurate estimate, which is consistent with the large amount of error that is actually present.

Second order metamodels were then investigated. The RF model produces the estimate $\hat{\beta}_0 = 2.96$. Measure r_1 can now be computed using Eq. (24) resulting in a value of -1.82, which indicates that the additional second order term is almost twice as large in magnitude as the first order term. Measure r_2 now has the value 7.95 compared with the value of 8.22 computed from the actual response data. Due to the facts that r_1 is not small and $\hat{r}_2 \approx r_2$, it can be assumed that the RF method is second order and in its asymptotic range, meaning that the extrapolated results can be accepted as accurate. This is consistent with the actual amount of error, -1.33%, present in the estimate. The RE model produced an estimate of $\hat{\beta}_0 = 2.77$ with $r_1 = 0.74$ and $\hat{r}_2 = 6.26$. The -23.86% error in \hat{r}_2 coupled with the relatively large value of r_1 indicates that the RE method is not second order and not in its asymptotic range. The estimate of $\hat{\beta}_0$ cannot be accepted as accurate, even though the actual error level of -7.67% might be accurate enough for some applications. Note that bad values of r_1 and \hat{r}_2 are necessary, but not sufficient, to conclude that the extrapolated result is inaccurate. Further details can be found in Ref. [18].

The second case considered for this simple example puts even greater demands on the extrapolation capabilities of the methods. The values of the discretization parameter are increased to $h_1 = 2.0$, $h_2 = 1.0$, $h_3 = 0.5$. Only second order models were considered. Table 2 shows a 2.00% error in the estimate of β_0 for the RF model, while there is a -283.67% errors in the corresponding RE estimate. The values listed for r_1 and \hat{r}_2 indicate that there is strong evidence

that the RF method is second order and in its asymptotic range, while there is no such evidence for the RE method. This is consistent with the actual error levels and illustrates the superior extrapolation capability of the RF representation. In all of the cases listed, the RF discretization error model clearly outperforms the RE model in extrapolating to the continuum at $h = 0$. Note that the nonlinear Marquardt LS method was unable to improve on the initial linear LS estimate of the RF model coefficients, indicating that the RF model is close-to-linear in this region of parameter space.

3.2 Finite Element Bar Example

The proposed methods are now applied to a more realistic example. The natural frequency of the third axial mode of a one dimensional continuous prismatic bar, constrained at one end, is to be predicted using results based on finite element models. This example is based on the problem studied by Alvin [9]. The exact continuum solution in Hertz is given by the expression

$$y = \frac{5\sqrt{E/\rho}}{4L} \quad (31)$$

Three quantitative inputs are considered, including elastic modulus (x_1), mass density (x_2), and length of the bar (x_3). Numerical experiments using finite element models of the beam were generated using the Box-Behnken design in Table 1. The standard deviations of the quantitative inputs are taken as 1.0% of their mean values. The coded values of -1 , 0 , and 1 for the three quantitative inputs in Table 1 represent low, nominal, and high levels, respectively. The nominal value of the inputs corresponds to their expected values, while the high and low levels represent plus and minus two standard deviations. In the following computations, the nominal values of the quantitative inputs E , ρ , and L , correspond to 10^7 psi, $0.10 \text{ lb/in}^3/\text{g}$, and 16.0 in, respectively.

The fourth model input is the discretization parameter h . In general, h can be expressed as

$$h = \frac{1}{N^{1/p}} \quad (32)$$

where N is the number of elements in the finite element mesh and p is the spatial order of the elements [14]. In this example, elements are one dimensional, therefore, h is just the inverse of the number of elements in the mesh. In the case of the discretization variable h , values -1 , 0 , and 1 correspond to coarse, nominal, and fine meshes. The actual values of h given by Eq. (32) are used in the design matrix instead of the coded values listed in Table 1. In each of the following cases, a consistent mass distribution was used and the meshes were uniform for simplicity. It is

believed that analogous results would be obtained for nonuniform meshes as considered by Alvin [9].

3.2.1 Case 1

The first case investigated uses a nominal mesh with 20 elements, $h_2=1/20$. Employing the usual convention of halving and doubling the number of elements, the coarse and fine meshes have 10 and 40 elements with discretization parameter values of $h_1=1/10$ and $h_3=1/40$, respectively. This particular case is comparable to the one considered by Alvin [9]. Linear LS was used to fit both the RE model in Eq. (9) and the RF model in Eq. (13) to the response data obtained from the numerical experiments. The estimated model coefficients were tested for statistical significance at the 95% confidence level using the t-Test statistic [3]. All 15 coefficients in the RE model were found to be statistically significant while the four h dependent terms in the numerator of the RF representation were insignificant resulting in the reduced RF metamodel

$$y_{hRF} = [\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3] \left[\frac{1}{1 + \gamma_0 h^2 + \gamma_1 h^4} \right] \quad (33)$$

This form of the discretization error is identical to the inverse polynomial form used in the simple convergence example of Section 3.1 in which α_1 was assumed to be zero.

Two measures of goodness for extrapolation capability are considered. The first is the estimate of the zero-intercept coefficient β_0 which represents the frequency corresponding to the nominal continuum model, 15,351.12 Hz. The results presented in Table 3 show that both the RE and RF metamodels are essentially exact in their predictions. In both cases, measure r_1 is small and $\hat{r}_2 \approx r_2$ indicating there is strong evidence that each of the methods is first order and in its asymptotic range.

One of the main goals of this work is to accurately extrapolate the discrete metamodel form to the continuum metamodel such that an accurate decoupled Monte Carlo analysis can be performed. Therefore, all the h -independent coefficients must be accurately estimated, not just β_0 . One hundred random experiments were selected within the initial design region and applied to the exact relation given in Eq. (31). The estimated RE and RF metamodels were extrapolated to $h = 0$ and applied to the same 100 experiments. The second measure of goodness considered is the maximum percentage residual error for the 100 experiments. Table 3 indicates that the RF model is slightly better, but both models give very accurate estimates within the design region. However, it is important to remember that the RF model is able to achieve this level of accuracy using three less

terms than the RE representation. As suggested earlier, simplicity is desired over generality in response surface analysis.

3.2.2 Case 2

The extrapolation capabilities of the RE and RF metamodels were then tested by using coarser meshes. In Case 2, the nominal mesh has only 6 elements, $h_2 = 1/6$, while the coarse and fine meshes correspond to

y insignificant, resulting in 13 terms in the RE metamodel. All the h dependent terms were retained. Linear LS was also applied to the RF model form. As in Case 1, all the h dependent terms in the numerator of the discretization error model were found to be statistically insignificant producing a reduced RF metamodel with 12 terms. The extrapolation performed in Case 1 was repeated with the results presented in Table 3. Percentage-wise, both metamodels accurately predict the nominal frequency for the continuum, but the RF metamodel is twice as accurate as the RE representation. With respect to the 100 random experiments within the design space, the RF model produced a residual error, 0.11%, that was less than half that of the RE model. In both cases, measure r_1 is not small and $\hat{r}_2 \approx r_2$ indicating there is strong evidence that each of the methods is second order and in its asymptotic range.

3.2.3 Case 3

Case 3 represents the most extreme case of extrapolation possible for the chosen example. The nominal mesh contains only 4 elements, while the coarse and fine meshes only contain 3 and 5 elements, respectively. In this case, the usual convention of halving and doubling the mesh is not followed. In an extreme case, this could be considered a perturbation of the nominal mesh. Linear LS was used to estimate each model and the same terms were found to be significant. The extrapolation results listed in Table 3 indicate that the RF representation is again more than twice as accurate as the RE model. As in Case 2, measure r_1 is not small and $\hat{r}_2 \approx r_2$ indicating strong evidence that each of the methods is second order and in its asymptotic range. An error of -1.57% in the prediction of the nominal continuum frequency and a residual error of 1.57% indicates that, even in this extreme case, the RF metamodel is an accurate predictor on the experiment design space for the continuum. The error measures are consistent with this observation. In an actual application, the estimate of coefficient β_0 can be used to estimate the discretization error present in a finite element mesh. For the refined mesh, the finite element model with nominal inputs predicts the frequency of 16,927.04 Hz. Using the RF estimate of β_0 , the mesh error can be estimated as 12.04%. Note that in all three cases discussed, the Marquardt nonlinear LS method could not improve on the linear LS solution.

3.3 Decoupled Monte Carlo Analysis

The ultimate goal of the extended metamodel approaches presented in this report is to eliminate or at least quantify the effect of mesh discretization error on the results of nondeterministic analyses. Assuming each of the inputs satisfies a normal probability distribution, a Monte Carlo analysis was performed using the exact relation for the continuum frequency given by Eq. (31). The inputs were randomly sampled 100,000 times and the results used to compute the probability of failure, defined as the vibrational frequency being less than 15,000 Hz. In each of the cases discussed, the metamodels were extrapolated to the continuum approximation at $h = 0$ and applied to the 100,000 samples used in the exact formulation.

In Case 1, which used the most refined meshes, both the RE and RF metamodels accurately predict the mean frequency value and the probability of failure as listed in Table 4. The benefit of the RF model is that it uses 3 less terms. Case 2 uses coarser meshes and begins to exercise the extrapolation capabilities of each of the metamodel forms. The results presented in Table 4 indicate that the RF model is more than twice as accurate as the RE representation. But, while the mean frequency value is very accurately estimated, the percentage error in the estimated probability of failure is large, at 21.24%. This result agrees with Alvin's bar example observations that the accuracy of the estimate of the mean value must be extreme to be able to accurately estimate small magnitude failure probabilities. The accuracy of the extrapolated metamodels in Case 2 is illustrated in Fig. 1 in which the percentage error in the probability of failure estimate is plotted against the number of standard deviations that the failure criterion is away from the mean frequency. The RF metamodel is clearly superior as the failure criterion deviates below the mean. Above the mean value, both methods approach zero error. Case 3 uses the most coarse meshes possible (3, 4, and 5 elements) for metamodel estimation. Even in this extreme case, the RF model produced small residual errors over all 100,000 experiments and an accurate estimate of the mean frequency at -1.57% error as listed in Table 4. The RF model, once again, is better than twice as accurate as the RE representation, but the RF probability of failure estimate is very inaccurate, while the RE prediction is grossly in error for this extreme case of extrapolation.

4. Conclusions

A new method has been presented for extending response surface techniques to include the effects of finite element model mesh discretization errors. The method employs a rational function representation of the discretization error in parameter h rather than the power series representation used by the straightforward Richardson extrapolation technique. Consistency measures were introduced to test the method order and whether the method is in its asymptotic range. Examples dealing with simple function estimation and the estimation of the third axial vibration frequency of a one dimensional bar showed that the rational function representation gave more accurate

extrapolated estimates using fewer terms than the Richardson extrapolation technique. In subsequent nondeterministic analyses used to estimate failure probabilities, the rational function based metamodel also produced more accurate estimates than Richardson extrapolation under severe extrapolation conditions. However, it was also shown that accurate estimates of small magnitude failure probabilities are difficult to obtain in cases of extreme extrapolation. Future work will address this problem.

While the rational function form of the discretization error produces a nonlinear model, requiring an iterative nonlinear least-squares solution technique, all the examples studied in this work proved to be close-to-linear, meaning that the linear least-squares estimate of the model coefficients could not be improved. It is believed that the rational function representation of the discretization error offers greater flexibility by allowing a user to accurately extrapolate to a continuum representation from numerical experiments performed outside of the asymptotic region where the usual power series representation is not converging. This allows the use of coarser meshes in the numerical experiments, saving a significant amount of time and effort.

5. References

- [1] Alvin, K. F. and W. L. Oberkampf, "Uncertainty Quantification in Computational Structural Dynamics: A New Paradigm for Model Validation," Sandia National Laboratories, 1997.
- [2] Romero, D. R. and S. D. Bankston, "Application of Decoupled Monte Carlo Analysis with Finite Element / Lattice Sampling Response Surface for Multimodal Test Problem," Sandia National Laboratories, 1997.
- [3] Khuri, A. I. and J. A. Cornell, *Response Surfaces: Designs and Analyses*, Marcel Dekker, Inc., New York, 1996.
- [4] Kaufman, M. and e. al., "Variable-Complexity Response Surface Approximations for Wing Structural Weight in HSCT Design," *Computational Mechanics*, Vol. 18, 1996, pp. 112-126.
- [5] Venter, G. and R. T. Haftka, "Minimum-Bias Based Experiment Design for Constructing Response Surfaces in Structural Optimization," *38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, FL, 1997, pp. 1225-1238.
- [6] Fox, E. P., "Methods of Integrating Probabilistic Design within an Organization's Design System using Box-Behnken Matrices," *34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, La Jolla, CA, 1993, pp. 714-723.
- [7] Fox, E. P., "The Pratt & Whitney Probabilistic Design System," *35th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Hilton Head, SC, 1994, pp. 1075-1085.
- [8] Fox, E. P., "Issues in Utilizing Response Surface Methodologies for Accurate Probabilistic Design," *AIAA Structures, Structural Dynamics, and Materials Conference*, Salt Lake City, UT, 1996, pp. 1615-1622.

Table 1. Box-Behnken design for four input variables.

<u>Runs</u>	<u>x1</u>	<u>x2</u>	<u>x3</u>	<u>x4</u>
1-4	±1	±1	0	0
5-8	0	±1	±1	0
9-12	0	0	±1	±1
13-16	±1	0	±1	0
17-20	0	±1	0	±1
21-24	±1	0	0	±1
25	0	0	0	0

Table 2. Metamodel extrapolation results for simple function example.

Case 1. $h_1 = 1.2, h_2 = 0.6, h_3 = 0.3$

<u>RF</u>	<u>$\hat{\beta}_0$</u>	<u>% Error</u>	<u>r_1</u>	<u>\hat{r}_2</u>	<u>r_2</u>	<u>% Error</u>
First Order	3.23	7.67	----	14.59	8.22	77.49
Second Order	2.96	-1.33	-1.82	7.95	8.22	-3.28
<u>RE</u>						
First Order	2.33	-22.33	----	4.00	8.22	-51.34
Second Order	2.77	-7.67	0.74	6.26	8.22	-23.84

Case 2. $h_1 = 2.0, h_2 = 1.0, h_3 = 0.5$

<u>RF</u>						
Second Order	3.06	2.00	-35.19	35.17	36.17	-2.76
<u>RE</u>						
Second Order	-5.51	-283.67	3.79	10.51	36.17	-70.94

Table 3. Simple beam extrapolation results.

Case 1. $h_1 = 1/10, h_2 = 1/20, h_3 = 1/40$						Max. Residual
	$\hat{\beta}_0$	% Error	r_1	\hat{r}_2	r_2	% Error
RF	15351.15	2.00e-4	-1.98e-2	4.02	4.02	6.00e-4
RE	15351.13	6.50e-5	5.97e-3	4.00	4.02	7.40e-4
Case 2. $h_1 = 1/3, h_2 = 1/6, h_3 = 1/12$						
RF	15334.93	-0.11	-0.55	2.38	2.38	0.11
RE	15319.67	-0.20	-0.38	2.38	2.38	0.27
Case 3. $h_1 = 1/3, h_2 = 1/4, h_3 = 1/5$						
RF	15110.19	-1.57	-0.62	0.92	0.92	1.57
RE	14841.28	-3.32	-0.49	0.92	0.92	3.63

Table 4. Nondeterministic beam analysis results.

Case 1. $h_1 = 1/10, h_2 = 1/20, h_3 = 1/40$					
Method	Max. Residual	Mean	% Probability		
	% Error	Value	% Error	of Failure	% Error
Continuum	0.00	15352.40	0.00	2.93	0.00
RF	8.2e-3	15352.43	2.0e-4	2.94	0.48
RE	1.0e-2	15352.43	2.0e-4	2.94	0.48
Case 2. $h_1 = 1/3, h_2 = 1/6, h_3 = 1/12$					
Continuum	0.00	15354.22	0.00	2.84	0.00
RF	0.11	15338.04	-0.11	3.45	21.24
RE	0.31	15323.46	-0.20	4.44	56.11
Case 3. $h_1 = 1/3, h_2 = 1/4, h_3 = 1/5$					
Continuum	0.00	15353.51	0.00	2.86	0.00
RF	1.58	15112.55	-1.57	27.36	856.30
RE	3.84	14844.51	-3.32	78.16	4886.10

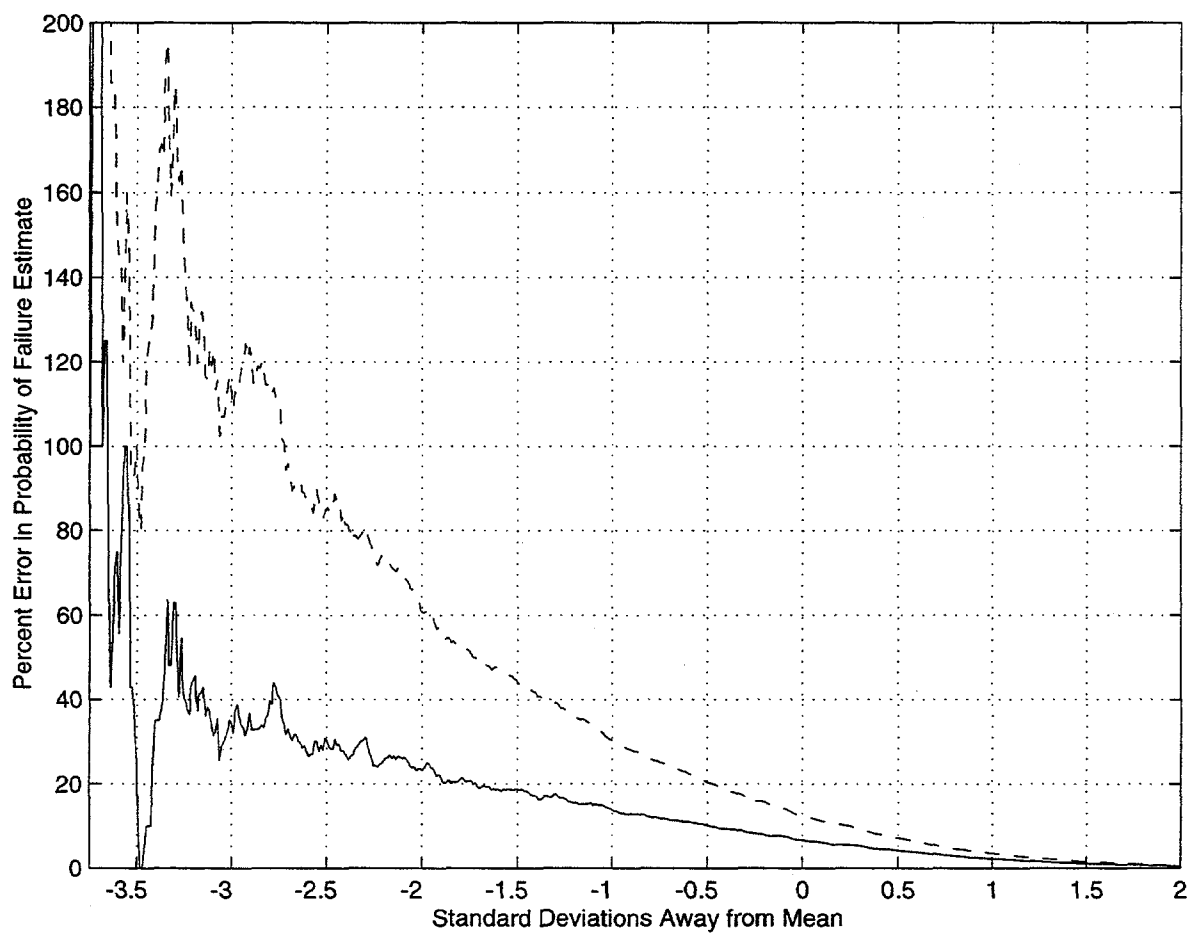


Figure 1. Case 2 probability of failure estimate errors. ----- RE metamodel, ----- RF metamodel

Intentionally Left Blank

DISTRIBUTION:

Anderson, M.
ACTA
2790 Skypark Dr., Suite 310
Torrance, CA 90505-5345

Balci, O.
Dept. of Computer Science
Virginia Tech.
Blacksburg, VA 24061

Cafeo, J. A.
General Motors R&D Center
Mail Code 480-106-256
30500 Mound Road
Box 9055
Warren, MI 48090-9055

Chen, C.-H.
Dept. of Systems Engineering
University of Pennsylvania
220 South 33rd St.
Philadelphia, PA 19104-6315

Coleman, H.
Dept. of Mech. and Aero. Engr.
Univ. of Alabama/Huntsville
Huntsville, AL 35899

Farhat, C.
Department of Aerospace Engineering
University of Colorado at Boulder
Campus Box 429
Boulder, CO 80309-0429

Flatau, A. F.
The National Science Foundation
4201 Wilson Blvd.
Arlington, VA, 22230

Ayyub, B.
Dept. of Civil Engineering
University of Maryland
College Park, MD 20742

Berman, A.
75 Kenwood Circle
Bloomfield, CT 06002-3435

Cavendish, J. C.
General Motors R&D Center
Mail Code 480-106-359
30500 Mound Road
Box 9055
Warren, MI 48090-9055

Craig, R. R., Jr.
University Of Texas At Austin
Aerospace Engineering
Mail Code C0600
Dept. Of Aerospace Engineering
Austin, TX 78712

Elishakoff, I.
Dept. of Mechanical Engineering
Florida Atlantic Univ.
777 Glades Rd.
Boca Raton, FL 33431-0991

Felippa, C. A.
Department of Aerospace Engineering
University of Colorado at Boulder
Campus Box 429
Boulder, CO 80309-0429

Ghanem, R.
Department of Civil Engineering
The Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218-2686

Glimm, J. G.
Dept. of Applied Math and Statistics
Math, P138A
State University of New York at Stony
Brook
Stony Brook, NY 11794-3600

Hills, R. G.
Department of Mechanical Engineering
New Mexico State University
Las Cruces, NM 88001

Inman, D. J.
Department of Engineering Science and
Mechanics
Virginia Tech.
Blacksburg, VA 24061-0219

Kammer, D. C.
539 Engineering Research Building
1500 Engineering Drive
Madison, WI 53706-1687

Mahadevan, S.
Dept of Civil and Environmental Engr.
Vanderbilt University
Box 6077, Station B
Nashville, TN 37235

Malkus, D. S.
533 Engineering Research Building
1500 Engineering Drive
Madison, WI 53706-1687

Juan Meza
DP 51
Dept. of Energy
1000 Independence Ave., SW
Washington, DC 20585

Hasselmann, T. K.
ACTA
2790 Skypark Dr., Suite 310
Torrance, CA 90505-5345

Hunt, D. L.
SDRC
11995 El Camino Real, Suite 200
San Diego, CA 92130

Kabe, A. M.
Structural Dynamics Department
Aerospace Corporation
P.O. Box 92957 - M4/911
Los Angeles, CA 90009-2957

Levine, M. B.
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099

Mair, H. U.
Institute of Defense Analyses
Operational Evaluation Division
1801 North Beauregard Street
Alexandria, VA 22311-1772

Paul Messina
DP 51
Dept. of Energy
1000 Independence Ave., SW
Washington, DC 20585

Morris, M. D.
Department of Statistics
304A Snedecor Hall
Iowa State University
Ames, IA 50011-1210

Osman, H.
Boeing Company
5301 Bolsa Ave.
Huntington Beach, CA 92647

Park, K. C.
Department of Aerospace Engineering
University of Colorado at Boulder
Campus Box 429
Boulder, CO 80309-0429

Penzold, L.
Mechanical and Environmental
Engineering
University of California-Santa Barbara
126 Engineering III
Santa Barbara, CA 93106

Peterson, L. D.
Department of Aerospace Engineering
University of Colorado at Boulder
Campus Box 429
Boulder, CO 80309-0429

William Reed
DP 51
Dept. of Energy
1000 Independence Ave., SW
Washington, DC 20585

Smith, S. W.
Mechanical Engineering Department
University of Kentucky
475 Anderson Hall
Lexington, KY 40506-0046

Zimmerman, D. C.
Department of Mechanical Engineering
4800 Calhoun Road
Houston, Texas, 77204-4792

Los Alamos National Laboratory
Mail Station 5000
PO Box 1663
Los Alamos, NM 87545
Attn: S. W. Doebbling, MS P946
Attn: C. Farrar, MS P946
Attn: K. M. Hanson, MS P940
Attn: F. M. Hemez, MS P946
Attn: J. M. Hyman, MS B284
Attn: M. McKay, MS F600

University of California
Lawrence Livermore National Laboratory
7000 East Ave.
P. O. Box 808
Livermore, CA 94550
Attn: Peter Brown, MS L-561
Attn: Randy Christensen, MS L-066
Attn: Richard Klein, MS L-023
Attn: S. Lee, MS L-560
Attn: Cynthia Nitta, MS L-096

SANDIA INTERNAL

MS 0453	M. W. Callahan, 2103	MS 0835	S. N. Kempka, 9111
MS 0453	D. L. McCoy, 2104	MS 0834	A. C. Ratzel, 9112
MS 0475	R. C. Hartwig, 2105	MS 0826	W. Hermina, 9113
MS 1393	F. F. Dean, 2106	MS 0834	J. Johannes, 9114
MS 0447	J. O. Harrison, 2111	MS 0834	K. S. Chen, 9114
MS 0447	P. D. Hoover, 2111	MS 0825	W. H. Rutledge, 9115
MS 0479	J. D. Gruda, 2151	MS 0825	F. G. Blottner, 9115
MS 0479	P. A. Sena, 2151	MS 0836	E. Hertel, 9116
MS 0479	M. H. Abt, 2151	MS 0836	R. O. Griffith, 9117
MS 0482	V. J. Johnson, 2161	MS 0835	J. S. Peery, 9121
MS 0481	M. A. Rosenthal, 2167	MS 0847	S. W. Key, 9121
MS 0481	T. F. Hendrickson, 2167	MS 0847	G. M. Reese, 9121
MS 0481	W. C. Moffat, 2167	MS 0555	M. S. Garrett, 9122
MS 0481	K. D. Meeks, 2168	MS 0847	H. S. Morgan, 9123
MS 0481	K. Ortiz, 2168	MS 0847	A. F. Fossum, 9123
MS 0509	J. S. Rottler, 2300	MS 0847	D. R. Martinez, 9124
MS 0537	A. Muyschondt, 2314	MS 0847	K. F. Alviin, 9124 (20)
MS 0769	D. S. Miyoshi, 5800	MS 0557	T. G. Carne, 9124
MS 0759	I. V. Waddoups, 5845	MS 0847	J. L. Dohner, 9124
MS 0759	M. S. Tierney, 5845	MS 0847	R. V. Field, 9124
MS 0782	R. V. Matalucci, 5861	MS 0847	D. J. Segalman, 9124
MS 0708	P. S. Veers, 6214	MS 0557	T. W. Simmermacher, 9124
MS 0746	R. M. Cranwell, 6411	MS 0557	D. O. Smallwood, 9124
MS 0746	D. J. Anderson, 6411	MS 0847	S. Wjotkievich, 9124
MS 0746	J. E. Campbell, 6411	MS 0557	T. J. Baca, 9125
MS 0746	D. G. Robinson, 6411	MS 0847	C. R. Adams, 9125
MS 0746	L. P. Swiler, 6411	MS 0847	C. W. Fulcher, 9125
MS 0977	W. R. Cook, 6524	MS 0557	D. L. Gregory, 9125
MS 0977	S. M. DeLand, 6524	MS 0847	D. W. Lobitz, 9125
MS 1137	A. L. Hodges, 6534	MS 0557	R. L. Mayes, 9125
MS 1137	G. K. Froehlich, 6535	MS 0847	H. P. Walther, 9125
MS 0771	M. S. Y. Chu, 6800	MS 0847	R. A. May, 9126
MS 0779	J. C. Helton, 6849	MS 0847	T. D. Hinnerichs, 9126
MS 9405	T. M. Dyer, 8700	MS 0827	J. R. Stewart, 9131
MS 9042	W. A. Kawahara, 8725	MS 0828	J. Moya, 9132
MS 9161	E. P. Chen, 8746	MS 0828	T. Y. Chu, 9132
MS 9003	K. E. Washington, 8900	MS 0828	M. Pilch, 9133
MS 9012	P. E. Nielan, 8920	MS 0828	R. S. Baty, 9133
MS 9217	M. L. Koszykowski, 8950	MS 0828	B. F. Blackwell, 9133
MS 1110	L. J. Lehoucq, 8950	MS 0828	K. J. Dowding, 9133
MS 0828	T. C. Bickel, 9101	MS 0828	K. E. Metzinger, 9133
MS 0828	R. K. Thomas, 9102	MS 0828	T. L. Paez, 9133
MS 0841	J. A. Fernandez, 9102	MS 0828	V. J. Romero, 9133
MS 0836	C. W. Peterson, 9104	MS 0828	A. Urbina, 9133
		MS 0828	W. R. Witkowski, 9133
		MS 0828	A. R. Lopez, 9133

MS 0828 W. L. Oberkampf, 9133 (10)
 MS 0828 C. Romero, 9133
 MS 1135 D. B. Davis, 9134
 MS 0321 W. J. Camp, 9200
 MS 0318 G. S. Davidson, 9201
 MS 0316 J. E. Kelly, 9202
 MS 0316 P. F. Chavez, 9204
 MS 0819 T. G. Trucano, 9211
 MS 1110 S. Chakerian, 9211
 MS 0847 M. S. Eldred, 9211
 MS 0847 A. A. Giunta, 9211
 MS 1110 W. Hart, 9211
 MS 1110 V. Leung, 9211
 MS 1110 C. A. Phillips, 9211
 MS 0847 J. R. Red-Horse, 9211
 MS 1110 B. van Bloeman Waanders,
 9211

 MS 0318 P. Heermann, 9215
 MS 1109 C. F. Diegert, 9215
 MS 1111 S. S. Dosanjh, 9221
 MS 1110 D. Womble, 9222
 MS 1110 D. M. Day, 9222
 MS 1110 R. Lehoucq, 9222
 MS 1110 S. Minkoff, 9222
 MS 0321 A. L. Hale, 9224
 MS 1111 G. Heffelfinger, 9225
 MS 0847 R. Leland, 9226
 MS 0847 C. R. Dohrmann, 9226
 MS 1111 B. Hendrickson, 9226
 MS 0820 P. Yarrington, 9232
 MS 0419 R. G. Easterling, 9800
 MS 9003 D. L. Crawford, 9900
 MS 0829 B. M. Rutherford, 12323 (5)
 MS 0829 K. V. Diegert, 12335 (5)
 MS0829 T. J. Kerschen, 12335
 MS 0337 R. D. Skocypec, 15002
 MS 1179 J. R. Lee, 15340
 MS 0188 D. L. Chavez, 4001, LDRD
 Office

 MS 0899 Technical Library, 4916 (2)
 MS 0612 Review and Approval Desk,
 4912 (1) for DOE/OSTI

 MS 9018 Central Technical Files, 8940-2
 MS 0161 Patent and Licensing Office,
 11500

Intentionally Left Blank