ELSEVIER

Reliability Engineering and System Safety 75 (2002) 333–357

RELIABILITY ENGINEERING & SYSTEM SAFETY

www.elsevier.com/locate/ress

Error and uncertainty in modeling and simulation

William L. Oberkampf[a,*], Sharon M. DeLand[b], Brian M. Rutherford[c], Kathleen V. Diegert[d], Kenneth F. Alvin[e]

[a]*Validation and Uncertainty Estimation Department, MS 0828, Sandia National Laboratories, Albuquerque, NM 87185-0828, USA*
[b]*Mission Analysis and Simulation Department, MS 1137, Sandia National Laboratories, Albuquerque, NM 87185-1137, USA*
[c]*Statistics and Human Factors Department, MS 0829, Sandia National Laboratories, Albuquerque, NM 87185-0829, USA*
[d]*Reliability Assessment Department, MS 0830, Sandia National Laboratories, Albuquerque, NM 87185-0830, USA*
[e]*Structural Dynamics and Smart Systems Department, MS 0847, Sandia National Laboratories, Albuquerque, NM 87185-0847, USA*

Received 14 April 2000; accepted 8 September 2001

Abstract

This article develops a general framework for identifying error and uncertainty in computational simulations that deal with the numerical solution of a set of partial differential equations (PDEs). 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. Our view incorporates the modeling and simulation phases that are recognized in the systems engineering and operations research communities, but it adds phases that are specific to the numerical solution of PDEs. In each of these phases, general sources of uncertainty, both aleatory and epistemic, and error are identified. Our general framework is applicable to any numerical discretization procedure for solving ODEs or PDEs. To demonstrate this framework, we describe a system-level example: the flight of an unguided, rocket-boosted, aircraft-launched missile. This example is discussed in detail at each of the six phases of modeling and simulation. Two alternative models of the flight dynamics are considered, along with aleatory uncertainty of the initial mass of the missile and epistemic uncertainty in the thrust of the rocket motor. We also investigate the interaction of modeling uncertainties and numerical integration error in the solution of the ordinary differential equations for the flight dynamics.

Keywords: Modeling; Simulation; Nondeterministic features; Epistemic uncertainty; Subjective uncertainty; Aleatory uncertainty; Stochastic uncertainty

1. Introduction

Realistic modeling and simulation of complex systems must include the nondeterministic features of the system and the environment. By ‘nondeterministic’ we mean that the response of the system is not precisely predictable because of the existence of uncertainty in the system or the environment, or human interaction with the system. Nondeterminism is thoroughly ingrained in the risk assessment community, where the emphasis is on estimating the safety of the system in abnormal or failure environments. Many examples exist in the literature of analyses for nuclear reactors, environmental impact, earthquake engineering and marine systems. The emphasis in the risk assessment field has been directed towards representing and propagating parameter uncertainties through mathematical models of the physical event. For reactor safety, significant emphasis has also been placed on identifying and analyzing fault trees and event trees in abnormal or failure event scenarios. The vast majority of this work has used probabilistic methods to represent sources of uncertainty and then sampling methods, such as Monte Carlo sampling, to propagate the sources through a deterministic model.

Our focus in this article is on a framework for identifying error and uncertainty in modeling and computational simulation. Our framework emphasizes models that are given by a set of partial differential equations (PDEs) that are to be solved numerically, although the framework is also applicable to modeling in general. We stress a clear distinction between the specification of the system, which is modeled by a set of PDEs, and the environment, which should be representative of the boundary conditions (BCs) and excitation for the PDEs. Our distinction between the system and the environment is similar to that taken in the thermodynamic analysis of engineering systems. We make a

[*] Corresponding author. Fax: +1-505-844-4523.
E-mail address: wloberk@sandia.gov (W.L. Oberkampf).

0951-8320/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0951-8320(01)00120-X

distinction between error and uncertainty so that the issues of representation and propagation of each is aided. Our taxonomy also distinguishes between errors that result from acknowledged sources, such as mathematical modeling approximations, and unacknowledged sources, such as computer programming errors. We differentiate between aleatory and epistemic uncertainty based on whether a source of nondeterminism is irreducible or reducible, respectively.

The issue of numerical solution error is generally ignored in risk assessment analyses and nondeterministic simulations. Neglecting numerical solution error can be particularly detrimental to uncertainty estimation when the mathematical models of interest are cast in terms of nonlinear 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 interactions. It is fair to say that the field of numerical error estimation is considered completely separate from that of uncertainty estimation. Although many authors in the field of numerical error estimation refer to solution error as 'numerical uncertainty', we believe this interpretation confuses the issue. Since we concentrate on systems described by the numerical solution of PDEs, we directly include possible sources of numerical and nonnumerical error in our framework and maintain a distinction between these and the sources of uncertainty.

We consider nondeterministic physical behavior originating from a very broad range of aleatory and epistemic uncertainties, in addition to inaccuracy due to modeling and simulation errors. Aleatory uncertainty is also referred to in the literature as irreducible uncertainty, inherent uncertainty, variability and stochastic uncertainty. The mathematical representation essentially always used for aleatory uncertainty is a probability or frequency distribution. Propagation of these distributions through a modeling and simulation process has been well developed in the disciplines mentioned above.

Epistemic uncertainty as a source of nondeterministic behavior derives from lack of knowledge of the system or the environment. In the literature, it is also referred to as reducible uncertainty, subjective uncertainty and cognitive uncertainty [1–12]. Although the distinction between aleatory and epistemic uncertainty has been in debate for the last decade, primarily in the risk assessment community, we firmly accept the distinction in our proposed framework. Once accepting this segregation of aleatory and epistemic uncertainty, one is faced with the question: Are probability (or frequency) distributions appropriate mathematical representations of uncertainty or should other representations, such as possibility theory or Dempster–Schafer theory, be used? Although this debate is raging in the literature, our proposed framework for modeling and simulation is not affected by this issue.

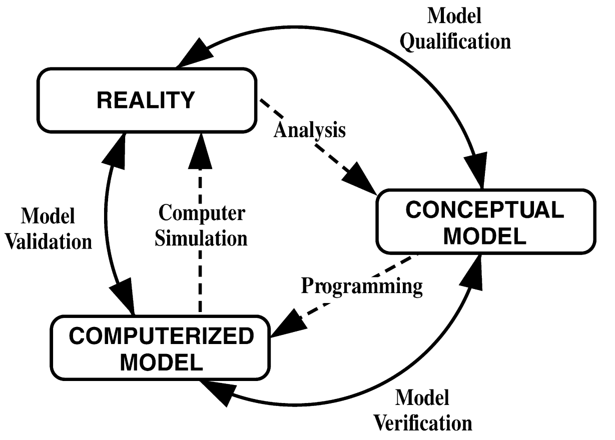
This article proposes a comprehensive, new framework (or structure) of the general phases of modeling and simulation. This structure is composed of six phases, which represent a synthesis of the activities recognized in the systems engineering and operations research communities, the risk assessment community, and the computational mathematics community. The phases are (1) conceptual modeling of the physical system (2) mathematical modeling of the conceptual model (3) discretization and algorithm selection for the mathematical model (4) computer programming of the discrete model (5) numerical solution of the computer program model, and (6) representation of the numerical solution. Characteristics and activities of each of the phases are discussed as they relate to a variety of disciplines in computational mechanics and thermal sciences. We also discuss the distinction between uncertainty and error that might occur in any of the phases of modeling and simulation.

To demonstrate this framework, we describe a system-level example: the flight of a rocket-boosted, aircraft-launched missile. In this example, we consider the missile to be a relatively short-range, i.e. 37 km (20 nm), unguided 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: aleatory uncertainty of the initial mass of the missile and epistemic uncertainty in the thrust of the rocket motor because of an unknown initial motor-temperature. To illustrate epistemic uncertainty due to mathematical modeling, we pursue two models with differing levels of physics: a six-degree-of-freedom (6-DOF) and a three-degree-of-freedom (3-DOF) model. Latin Hypercube Sampling (LHS) was used to propagate the aleatory uncertainty of initial mass through each of the two mathematical models. For 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 systems engineering and operations research (OR) communities have developed many of the general principles and procedures for modeling and simulation. Researchers in this field have made significant progress in defining and categorizing the various phases of modeling and simulation. (For texts in this field, see Refs. [13–18].) The areas of emphasis in OR include definition of the problem entity, definition of the conceptual model, assessment of data and information quality, validation methodology, and usage of simulation results as an aid in decision making. Although



© 1979 by Simulation Councils, Inc.

Fig. 1. View of modeling and simulation by the Society for Computer Simulation [19].

this work does not take a strong numerical methods perspective, it is very helpful in providing a constructive philosophical approach for identifying sources of uncertainty and error, as well as in 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 [19]. Included as Fig. 1, the diagram shows that analysis is used to construct a conceptual model of reality. Programming converts the conceptual/mathematical model into a computerized model. Then computer simulation is used to simulate 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 or the activities necessary for the estimation of uncertainty.

Jacoby and Kowalik developed a more detailed view for the phases of modeling and simulation in 1980 (Fig. 2) [20]. Their view not only better defined the phases of modeling and simulation but also emphasized the mathematical modeling aspects of the process. After the purpose or objective of the modeling effort is clarified and refined, a prototype modeling effort is conducted. The activities described in this effort are similar to those referred to in the present literature as the conceptual modeling phase. In the preliminary modeling and mathematical modeling phases, various alternate mathematical models are constructed and their feasibility is evaluated. In the solution technique phase, the numerical methods for solving the mathematical model, or models, are specified. In the computer program phase, the actual coding of all the numerical methods is conducted, as well as debugging of the code. In the model phase, all activities related to model validation, i.e. comparisons with experimental data, and checks on the reasonableness of predicted results are described. In the modeling result phase, the interpretation of results is conducted and an attempt is made to satisfy the original purpose of the modeling and simulation effort. The feedback and iterative nature of the entire process is represented by the dashed-loop circling the modeling and simulation effort.

Throughout the 1980s, Sargent [21,22] made improvements toward generalizing the concepts of modeling and simulation shown in Fig. 1. In this regard, his most important contribution was the development of general procedures for verification and validation of models and simulations. An extension of the phases of modeling and simulation was made by Nance [23] and Balci [24] to include the concept of the life cycle of a simulation. Major phases added by Nance and Balci to the earlier

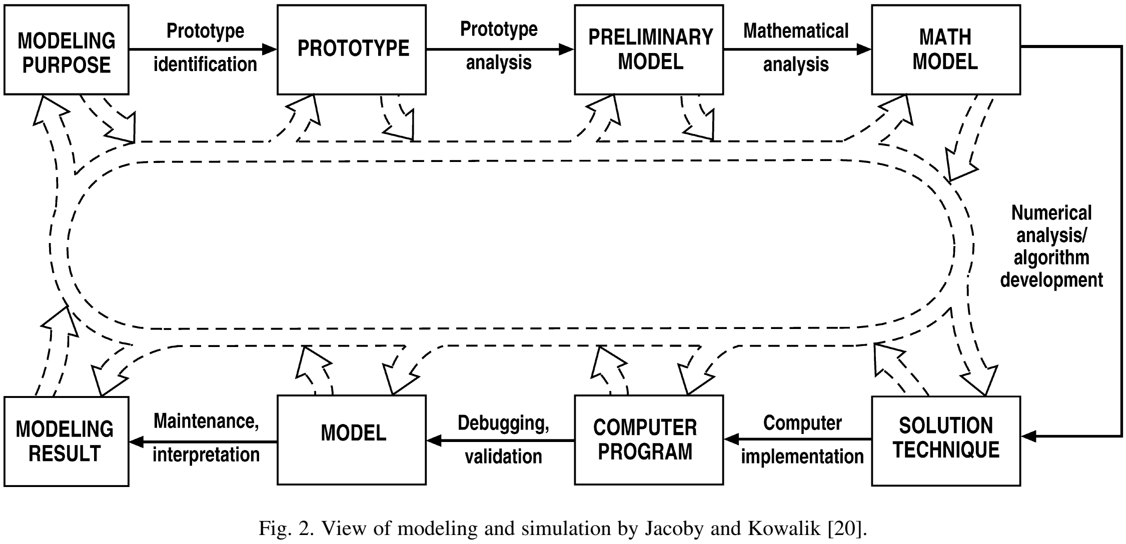


Fig. 2. View of modeling and simulation by Jacoby and Kowalik [20].

description were System and Objectives Definition, Communicative Models, and Simulation Results. Even though the Objectives Definition and Simulation Results phases were specifically identified earlier by Jacoby and Kowalik [20], there is no indication this work was recognized. Communicative Models were described by Nance and Balci as 'a model representation which can be communicated to other humans, can be judged or compared against the system and the study objective by more than one human' [23].

Work in the risk assessment community, specifically nuclear reactor safety and environmental impact of radionuclides, has not directly addressed the phases of modeling and simulation. Instead, investigators have concentrated on the possible sources that could contribute to uncertainty in risk assessment predictions. Reactor safety analyses have developed extensive methods for constructing possible failure and event-tree scenarios that aid in risk assessment [4,12,25–28]. Analyses of the risk of geologic repositories for the disposal of low-level and high-level nuclear waste have used scenario analyses and have identified sources of indeterminacy and inaccuracy occurring in other phases of the risk analysis. Specifically, these analyses have identified different types of sources occurring in conceptual modeling, mathematical modeling, computer code implementation, and experimentally measured or derived model input data [29,30].

The development of the present framework for the phases of modeling and simulation builds on much of this previous work. Our framework can thus be viewed as a synthesis of this reviewed literature, with three substantial additions. First, we describe a more precise distinction between the system and the environment. Second, we place more emphasis on the distinction between aleatory and epistemic uncertainty in the analysis. Third, we include a dominant element in the simulation of complex physical processes; the numerical solution of nonlinear PDEs. Our integration of these perspectives will be presented and discussed in Sections 2.3 and 3.

2.2. *Segregation of uncertainty and error*

Sources of uncertainty and error are associated with each phase of modeling and simulation. Examining the literature in fields that deal with the simulation of complex systems or physical processes (e.g. operations research, risk assessment, structural dynamics, and fluid dynamics), one finds that most authors do not carefully distinguish between what they mean by uncertainty and error; and more commonly, they do not distinguish between aleatory and epistemic uncertainty. Even when these terms have been defined, their definitions are typically couched in the restricted context of the particular subject or do not address the issue of error [4,25,27,28,31]. During the last ten years some authors in the risk assessment field have begun to clearly distinguish between some of these sources, particularly the distinction between aleatory and epistemic uncertainty [1–5,7–11,32–42]. The risk assessment field is the first to use the separate notion and treatment of aleatory uncertainty and epistemic uncertainty in practical applications. We are convinced of the constructive value of this distinction and we will adopt essentially the same definitions used by these authors.

We use the term *aleatory uncertainty* to describe the *inherent* variation associated with the physical system or the environment under consideration. Sources of aleatory uncertainty 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. As mentioned earlier, aleatory uncertainty is also referred to in the literature as irreducible uncertainty, inherent uncertainty, variability and stochastic uncertainty. Aleatory uncertainty is generally quantified by a probability or frequency distribution when sufficient information is available to estimate the distribution.

We define *epistemic uncertainty* as a *potential* inaccuracy in any phase or activity of the modeling process that is due to *lack of knowledge*. As mentioned previously, our definition of epistemic uncertainty is also referred to in the literature as reducible uncertainty, subjective uncertainty, and cognitive uncertainty. The first feature that our definition stresses is 'potential', meaning that the inaccuracy may or may not exist. In other words, there may be no inaccuracy, 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 epistemic uncertainty is that its fundamental cause is incomplete information. Incomplete information can be caused by vagueness, nonspecificity, or dissonance [43,44]. Vagueness characterizes information that is imprecisely defined, unclear, or indistinct, e.g. imprecise set membership. 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. For example, one could have the situation where only one value in a set is correct or appropriate, but because of insufficient information, the true value is not known. 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 of subsets belong to multiple sets that either do not overlap or overlap slightly. In addition to probabilistic modeling, a number of mathematical theories are available for modeling epistemic uncertainty, for example, possibility theory [45,46], evidence (Dempster/Shafer) theory [47,48], fuzzy set theory [49,50], imprecise probability theory [51,52] and Bayesian estimation [53,54].

We define *error* as a *recognizable* inaccuracy in any phase or activity of modeling and simulation that is *not*

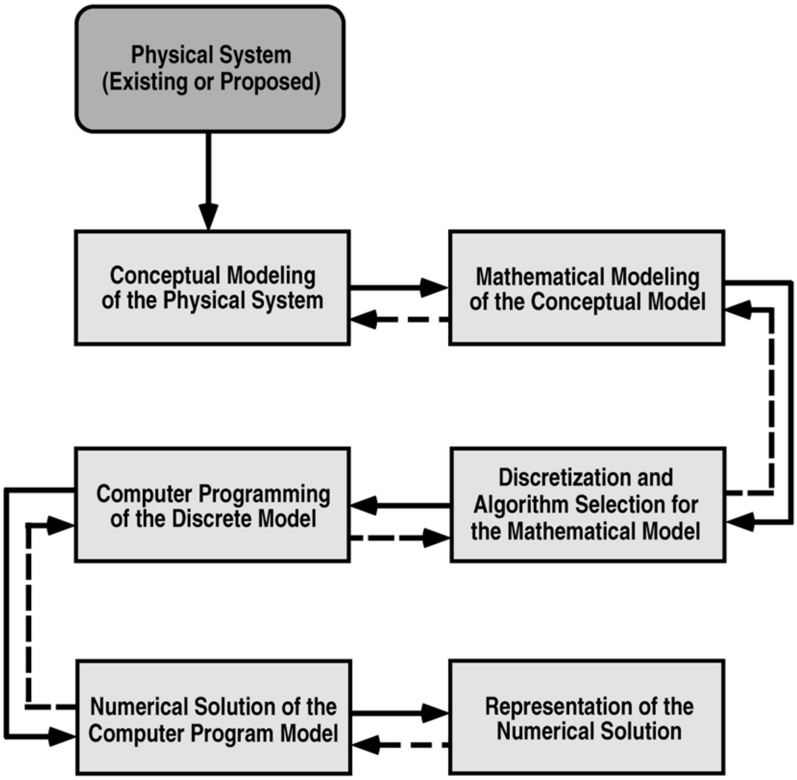


Fig. 3. Proposed phases for computational modeling and simulation.

due to lack of knowledge. Our definition stresses that the inaccuracy is identifiable or knowable upon examination; that is, the inaccuracy is not caused by lack of knowledge. Essentially, there is an agreed-upon or correct approach that is considered to be more accurate or correct. 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 cost or schedule. For example, the error may be deemed acceptable for the analysis requirements or because of the excessive computational cost to correct it. This recognizability implies a segregation of error types: an error can be either *acknowledged* or *unacknowledged*. Acknowledged errors are those inaccuracies that *are* recognized by the analysts. When acknowledged errors are introduced by the analyst into the modeling or simulation process, the analyst typically has some idea of the magnitude or impact of such errors. Examples of acknowledged errors are finite precision arithmetic in a computer, assumptions and approximations made to simplify the modeling of a physical process, and conversion of PDEs into discrete numerical equations. Unacknowledged errors are those inaccuracies that are *not* recognized by the analyst, but they are recognizable. Examples of unacknowledged errors are blunders or mistakes; e.g. the analyst intended to do one thing in the modeling and simulation but as a result of human error perhaps, did another. Unfortunately, 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 a computer program reveals that a coding error has been made, or by others because of redundancy procedures built into the analysis.

2.3. Proposed phases of modeling and simulation

Fig. 3 depicts our representation of the phases of modeling and simulation. The phases represent collections of activities or tasks required in a large-scale simulation analysis, particularly models given by PDEs and their numerical solution. The ordering of the phases implies an information and data flow that indicates which activities are likely to impact decisions and methodology occurring in later phases. However, there is significant feedback and interaction between the phases, as indicated by the dashed lines in Fig. 3. These phases follow the recent work of Refs. [55,56]. The paragraphs below provide brief descriptions of each of these phases. The modeling and simulation process is initiated by a set of questions posed by a designer or decision maker. The information to address these questions can be provided (at least in part) through a computer simulation analysis.

Conceptual modeling of the physical system. Our initial phase encompasses developing a specification of the physical system and the environment. Specification development includes determining which physical events, or sequence of events, and which types of coupling of different physical processes will be considered. It also includes identifying elements of the system and environment that will be treated as nondeterministic. These determinations should be based on the general requirements for the modeling and simulation effort, together with information regarding possible system response sensitivity to these events, processes and elements. The physical system can be an existing system or process, or it can be a proposed system or process. During the conceptual modeling phase, no mathematical equations are written, but the fundamental assumptions regarding 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 (PRA) analyses of nuclear reactor safety. Whether or not the event sequence will eventually be analyzed should not be a factor that impacts its inclusion in the conceptual modeling phase.

After the system and environment have been carefully specified, options for various levels of possible physics couplings should be identified, even if it is considered unlikely that 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 without returning to this phase and adding it. For example, if a particular type of physics coupling is recognized as important late in an analysis, then one must return to the conceptual modeling phase where it is to be added. This new thread can weave its way through the remainder of the phases and it would likely add new branching threads. Another activity conducted in the conceptual modeling phase is to identify all of the system and environment characteristics that might be treated nondeterministically. Consideration should be given to the treatment of these characteristics as aleatory or epistemic uncertainties. However, issues such as mathematical representation and propagation of these characteristics should be deferred until later phases.

Mathematical modeling of the conceptual model. The primary activity in this phase is to develop detailed and 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 process, 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 PDEs, auxiliary conditions, BCs, and initial conditions (ICs) for the system. For example, if the problem being addressed is a fluid-structure interaction, then all of the coupled fluid-structure PDEs must be specified, along with any fluid or material-property changes that might occur as a result of their interaction. The integral form of the equations could also be considered, but this type of formulation is not addressed in the present discussion.

Another function addressed during the mathematical modeling phase is the selection of 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 commonly 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 aleatory uncertainties generally requires that probability distributions can be established, either through data analysis or through subjective judgments. Common sources of aleatory uncertainty include system parameters, BCs, and ICs that may vary randomly from component to component and/or from system to system.

Mathematical modeling can result in an epistemic uncertainty or an acknowledged error when alternative models can be used to address the same aspects of the problem. Presumably, only one model is more correct for the simulation, but this information is not generally known beforehand, i.e. in a prediction. Furthermore, a preferred model might be too expensive for use exclusively in the analysis and, as a result, less accurate models would be used for portions of the analysis.

Our emphasis on comprehensiveness in the mathematical model should not be interpreted as an emphasis on complexity of the model. The predictive power of a model depends on its ability to correctly identify the dominant controlling factors and their influences, not upon its completeness or complexity. A model of limited, but known, applicability is often more useful than a more complete model. This dictum of engineering seems to be forgotten today with the advent of rapidly increasing computing power. The clear tendency, observable in most fields of engineering and science, is to use very complex models and then make one or two 'heroic' simulations. We do not believe this approach is the best use of available resources in a system analysis.

An additional point concerning the incompleteness of models should be made. Any mathematical model, regardless of its physical level of detail, is *by definition* a simplification of reality. Any complex engineering system, or even individual physical processes, contains phenomena that are not represented in the model. Statements such as 'full physics simulations' can only be considered as marketing jargon. This point was succinctly stated twenty years ago by Box [57]: 'All models are wrong, some are useful'.

Discretization and algorithm selection for the mathematical model. This phase accomplishes two activities related to converting the mathematical models into a form that can be addressed through computational analysis. The first activity involves conversion of the continuum mathematics form of the mathematical model into a discrete, or numerical, model. Simply stated, the mathematics are mapped from a calculus problem to an arithmetic problem. During the discretization phase, all of the spatial and temporal differencing methods, discretized BCs, 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 numerically specified. The discretization phase focuses on the conversion from continuum mathematics to discrete mathematics, *not* on numerical solution issues. Note that we are only referring to continuum and discrete mathematics, *not* continuum and discrete models of physics, such as Eulerian and Lagrangian models. We strongly believe that the continuum mathematical model and the discrete model should be separately represented in the phases of modeling and simulation. 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 activity of this phase focuses on specifying the methodology that dictates which computer runs will be performed in a later phase of the analysis to accommodate the nondeterministic aspects of the problem. For example, a Monte Carlo method or response surface method could be chosen for propagating uncertainties. Also, this activity would include decisions made on how to design and execute computer experiments.

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 [58,59]. These efforts have made a significant impact in areas such as commercial graphics and mathematics 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 engineering applications, as well as most applications written for massively parallel computers.

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 values and discrete solutions exist with finite precision. For example, a spatial grid distribution and a time step are fully specified. Dependent variables and independent variables, space and time, exist *only* at discrete points. However, these points may be altered during subsequent computer runs.

Multiple computational solutions are usually 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 numerical solution 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 has 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 deterministic BCs and ICs. For the individual solutions, the primary activity is to construct what-appears-to-be continuous functions based on the discrete solutions obtained in the previous phase. For example, one can construct the interpolated dependent variables given in the PDEs, which are based on the discrete solution of the PDEs; however, the construction involves new approximations.

We have specifically included representation of the numerical solution as a separate phase in the modeling and simulation process because of the sophisticated software that is being developed to comprehend modern complex simulations. This area includes three-dimensional graphical visualization of a solution, animation of a solution, use of sound for improved interpretation, and use of virtual reality, which allows analysts to 'go into the solution space'. Some may argue that this final phase is simply 'postprocessing' of the computational data. We believe, however, this description does not do justice to the rapidly growing importance of this area and the likelihood that it introduces unique types of errors. In addition, by referring to this phase as representation of the numerical solution, we are able to include types of errors that are due not simply to the modeling and simulation of the system, but also to the processing of the computed solution and to the conclusions drawn the refrom.

Summary. The phases of modeling and simulation described above illustrate the major components involved in planning and conducting a large-scale simulation analysis. When viewed from the planning aspect, the issues confronted in each phase may be addressed simultaneously. For example, in most large-scale system simulations the activities will be performed by different groups of people with different areas of expertise, such as professional planners, physicists, chemists, engineers, computer programmers, and numerical analysts. A ‘feedback’ aspect indicated in Fig. 3, but not explicitly discussed here, is the use of sensitivity analyses in a large-scale analysis. Sensitivity analyses and scoping studies are critical when there are hundreds of uncertainties in an analysis. Sensitivity analyses and scoping studies are clear examples of how feedback from the solution representation phase occurs in a large-scale analysis. There is, however, a clear sequential aspect to the phases as shown in Fig. 3. Two key sequential features of this illustration are that decisions must be made at each phase and that continuous parameters and model specification information propagate through the remaining phases. In most cases, the decisions made at one phase will impact the models formulated or activities conducted in later phases.

3. Missile flight example

3.1. Description of the problem

Our example to demonstrate the framework considers the flight of a rocket-boosted, aircraft-launched missile. In our analysis, we make the following assumptions about 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. Thus, we are attempting to predict future plausible events, not to analyze an event in the past or update models based on past observations of the system.

Fig. 4 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 uncertainty (either aleatory uncertainty ‘A uncertainty’ or epistemic uncertainty ‘E uncertainty’) and error (either acknowledged error or unacknowledged error) that typically occur in each activity. We now discuss in detail the activities that are conducted in each of the phases and explain how these activities are applied to the missile flight example.

3.2. Conceptual modeling activities

As seen in Fig. 4, we have identified four major activities that are conducted in the conceptual modeling phase: system/environment specification, scenario abstraction, coupled physics specification and nondeterministic specification. The system/environment-specification activity consists primarily of carefully identifying the physical or conceptual elements that are considered part of the system and those that are 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 thermodynamic analyses. The state of a system is influenced by processes internal to the system (endogenous processes) and also processes or activities external to the system (exogenous effects). Exogenous processes or activities are considered part of the environment. A system is influenced by the environment, but the environment *cannot* be influenced by the system [14]. 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 judgment and are not unique. As a result, these specifications pose one of the most difficult conceptual issues in modeling and simulation.

Fig. 5 shows three possible system/environment specifications for the missile flight example. The specifications are listed from the most physically inclusive (with regard to the system specification) to the 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 missile, the flow field of the missile, and the rocket exhaust are coupled to the flow field of the launching aircraft. Thus, the missile and rocket exhaust could interact with the aircraft flow field, but the aircraft structure, for example, could not change its geometry or deformation due to the rocket exhaust. Another example allowed by this 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, atmosphere near the missile, launching aircraft and target are considered part of the environment. Even though this is the simplest specification considered, it still allows for significant complexities in the analysis. Note that the missile flight example presented here will only pursue System/Environment Specification 3.

The scenario-abstraction activity attempts to identify all

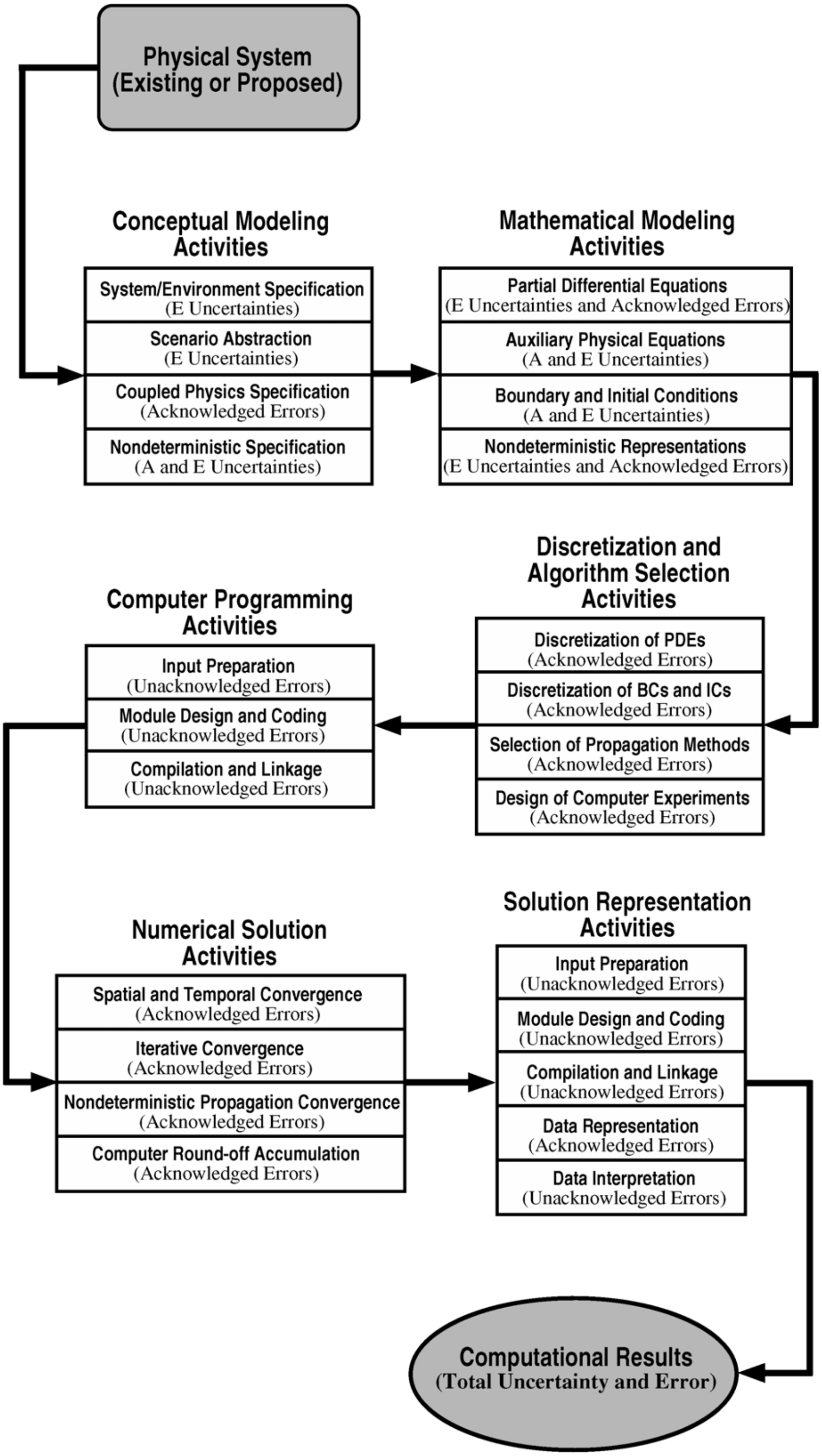


Fig. 4. Activities conducted in the phases of computational modeling and simulation.

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, this activity can be straightforward. However, complex engineered systems 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. Constructing

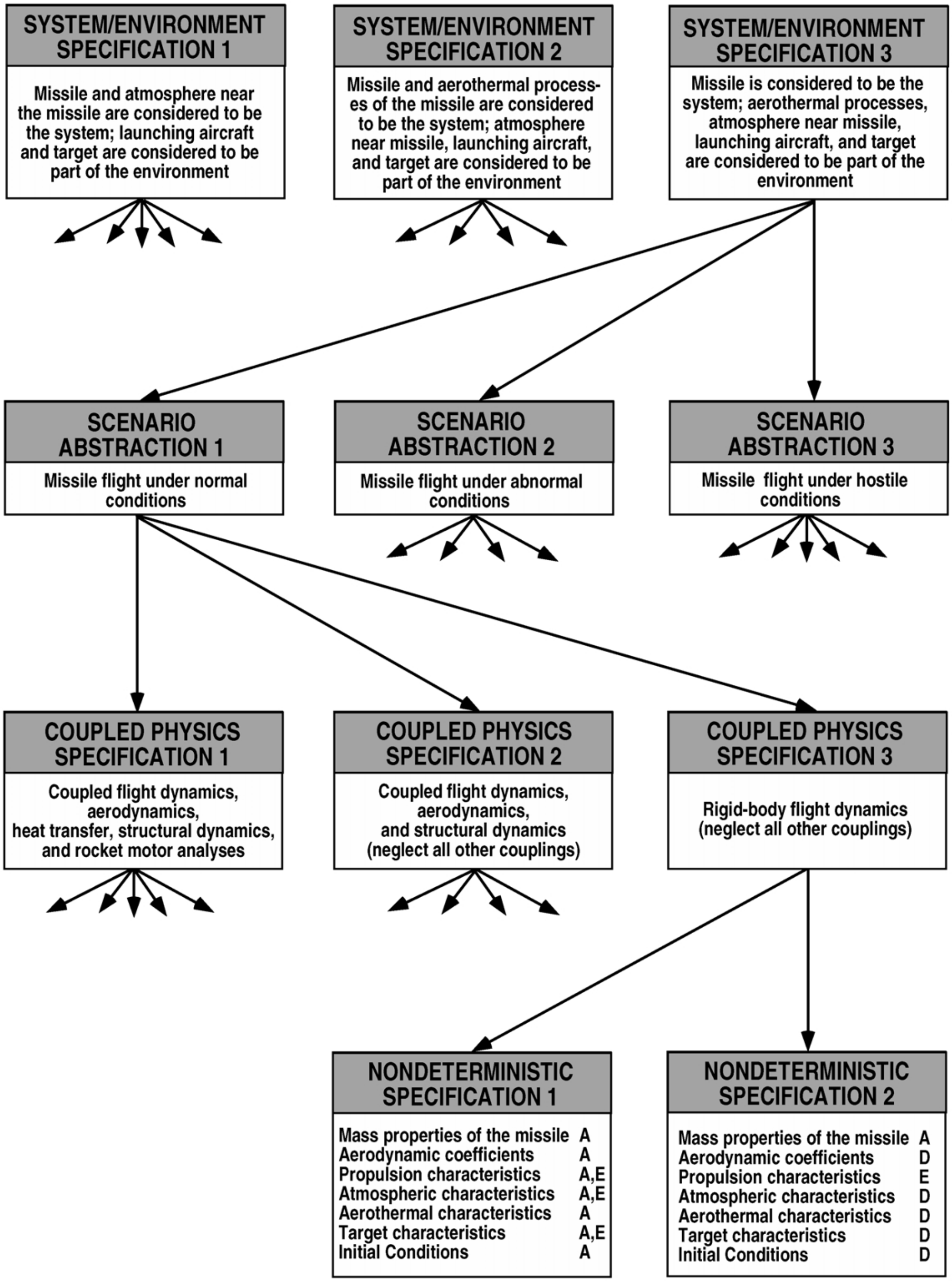

SYSTEM/ENVIRONMENT SPECIFICATION 1
Missile and atmosphere near the missile are considered to be the system; launching aircraft and target are considered to be part of the environment
SYSTEM/ENVIRONMENT SPECIFICATION 2
Missile and aerothermal processes of the missile are considered to be the system; atmosphere near missile, launching aircraft, and target are considered to be part of the environment
SYSTEM/ENVIRONMENT SPECIFICATION 3
Missile is considered to be the system; aerothermal processes, atmosphere near missile, launching aircraft, and target are considered to be part of the environment
SCENARIO ABSTRACTION 1
Missile flight under normal conditions
SCENARIO ABSTRACTION 2
Missile flight under abnormal conditions
SCENARIO ABSTRACTION 3
Missile flight under hostile conditions
COUPLED PHYSICS SPECIFICATION 1
Coupled flight dynamics, aerodynamics, heat transfer, structural dynamics, and rocket motor analyses
COUPLED PHYSICS SPECIFICATION 2
Coupled flight dynamics, aerodynamics, and structural dynamics (neglect all other couplings)
COUPLED PHYSICS SPECIFICATION 3
Rigid-body flight dynamics (neglect all other couplings)
NONDETERMINISTIC SPECIFICATION 1
Mass properties of the missile A
Aerodynamic coefficients A
Propulsion characteristics A,E
Atmospheric characteristics A,E
Aerothermal characteristics A
Target characteristics A,E
Initial Conditions A
NONDETERMINISTIC SPECIFICATION 2
Mass properties of the missile A
Aerodynamic coefficients D
Propulsion characteristics E
Atmospheric characteristics D
Aerothermal characteristics D
Target characteristics D
Initial Conditions D

Fig. 5. Conceptual modeling activities for the missile flight example.

scenario abstractions for these complex systems is a mammoth undertaking. The field of engineering that has achieved the highest development of scenario abstraction is the PRA of nuclear power plants. PRA techniques construct a many-branched event tree for complex operating and failure scenarios. Even though the probability of occurrence of certain events may be extremely low, these events should be considered and analyzed for failure of nuclear power plants and other high-consequence systems.

The scenario abstraction considered here includes both event-tree and decision-tree construction. Decision-tree construction does not necessarily depend on events but can identify possible results or analysis paths that are based on modeling decisions.

As shown in Fig. 5, the missile flight example identifies three broad classes of scenarios: missile flight under normal, abnormal and hostile conditions. Normal conditions are those that can be reasonably expected, 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 like snow or lightning. Examples of flight under hostile conditions would be detonation of nearby defensive systems, damage to missile components resulting from small-arms fire and damage from laser or microwave defensive systems. The three scenario classes considered here have been commonly used for military systems like nuclear weapons. Note that the missile flight example will only pursue Scenario Abstraction 1. With the increasing concern of terrorist attacks on civilian systems such as buildings, commercial aircraft, bridges and dams, these three scenario classes may, unfortunately, prove to be more broadly useful in the future.

The third activity, coupled physics specification Figs. 4 and 5, identifies and carefully distinguishes the possible alternatives for physical and chemical processes in the system, and the coupling between these processes 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. 5), we identify three levels of physics coupling, although more alternatives could be identified. Coupled Physics Specification 1 essentially couples all physics that could exist in this decision-thread of the analysis. 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 nonrigid-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. Coupled Physics Specification 2 would result in a time-dependent, coupled fluid/structure 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 is allowed to respond only to inputs or forcing functions from the environment. Structural dynamics is removed from the analysis; i.e. only rigid-body dynamics is considered. Note that the missile flight example will only pursue Coupled Physics Specification 3.

Before addressing the last activity of conceptual modeling, we will consider the possible sources of aleatory and epistemic uncertainty and error that could occur in the three activities discussed so far. The activities of developing system/environment specifications and scenario abstractions introduce epistemic uncertainties into the modeling and simulation process. These epistemic uncertainties arise primarily because of what is *not* included or scenarios that are *not* imagined, but are possible. Modarres, Kaminskiy and Krivtsov [60] refer to this type of uncertainty as 'completeness uncertainty'. The wider the scope of the analysis is or the more complex the system is, the larger is the number of possibilities that exist for epistemic uncertainties due to lack of knowledge about aspects of the modeled system and environment. Indeed, a common weakness of modern technological risk analyses is the exclusion, either intentionally or unintentionally, of unusual events, effects and possibilities [61]. For example, automatic control systems designed to ensure safe operation of complex systems can fail in unexpected ways, or the safety systems can be overridden during testing or maintenance. During construction of coupled physics specifications, acknowledged error is of primary concern. A hierarchical ordering of levels of physical coupling in conceptual models can commonly be constructed. Based on experience with similar systems, previous analyses, failure consequences, economic and liability 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.

For the missile flight example, we list only two alternative nondeterministic specifications, as shown in Fig. 5. Nondeterministic Specification 1 includes the following aleatory uncertainties (indicated by an AU in Fig. 5): mass properties of the missile, aerodynamic force and moment coefficients, aerothermal heating characteristics and ICs at missile launch. These are considered aleatory uncertainties because they are usually associated with random variation due to manufacturing processes or physically random processes. If many missiles were manufactured, for example, there would normally be sufficient inspection data so that a representative probability distribution for each parameter could be constructed. Nondeterministic Specification 1 also includes the following as aleatory and/or epistemic uncertainties (indicated by an AU, EU in Fig. 5): propulsion characteristics, atmospheric characteristics and target characteristics. These quantities could be considered as aleatory uncertainties, 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 for the random variation due to manufacturing have been constructed and these models are based on the known age and temperature of the propellant. If the age of the propellant in a particular motor is not known or the temperature of the propellant in a particular motor is not known, the statistical models are of little value in estimating the variation in the motor's performance. A similar argument can be made for estimating the uncertainty in atmospheric characteristics like wind conditions or temperature. Without specifying additional knowledge, such as location on earth, month of year, or even time of day, statistical models would include these effects as additional random variables.

The missile flight example will only pursue Nondeterministic Specification 2. For this specification, we chose one parameter of the mass properties of the missile as an aleatory uncertainty and one characteristic of the propulsion system as an epistemic uncertainty. We pursue Specification 2 in the example to distinguish the characteristics of each and to show how these parameters might be represented differently in a computational simulation. All other parameters are considered deterministic (indicated by a D in Fig. 5).

3.3. Mathematical modeling activities

As shown in Fig. 4, we have identified four major activities in the mathematical modeling phase: formulation of the PDEs, selection of all auxiliary equations that supplement the differential equations, formulation of all ICs and BCs 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 that are required to complete the PDEs. Examples are turbulence-modeling equations in fluid dynamics, equations of state in shock wave physics, material-constitutive equations in solid dynamics, neutron cross-sections in neutron transport and environmental excitation conditions. The auxiliary equations can be of any type, e.g. algebraic equations, integral equations, or PDEs. The BCs and ICs provide the required closure equations needed for all PDEs.

Formulation of the nondeterministic representations is based on the needs of the analysis, as well as the quantity and quality of relevant information available. Aleatory uncertainties commonly dominate the nondeterministic features of the auxiliary physical equations and the BC and IC activities. The most common aleatory uncertainties are those due to inherent randomness of continuous parameters in these equations. Aleatory uncertainties are nearly always represented by probability distributions. In some cases, the form of these distributions is inferred from first principles of the processes used to determine the parameter values. In most cases, the distributions are chosen based on very little data, on convention, or convenience. Parameters associated with the probability distributions are then estimated when sufficient data are available or assigned values based on a subjective assessment when insufficient data are available.

Epistemic 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 epistemic uncertainties that occur in the PDEs are limited knowledge of the equations for turbulent-reacting flow, steam explosions and crack propagation in materials, particularly nonhomogeneous and nonisotropic materials. For physical processes that are well understood, inaccuracies in certain models should be considered as errors rather than epistemic uncertainties. This guideline is based on the argument that if significant knowledge of the process exists, a set of alternative models can be convincingly ordered in terms of increasing accuracy. In the modeling of fluid dynamic turbulence, the models can be generally 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 the other because certain lower-order models can be very accurate for specialized 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. The invariable case is 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 with which they are familiar, schedule constraints and 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 role. 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 6-DOF model and a 3-DOF model (Fig. 6). Both 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. 5). The translational

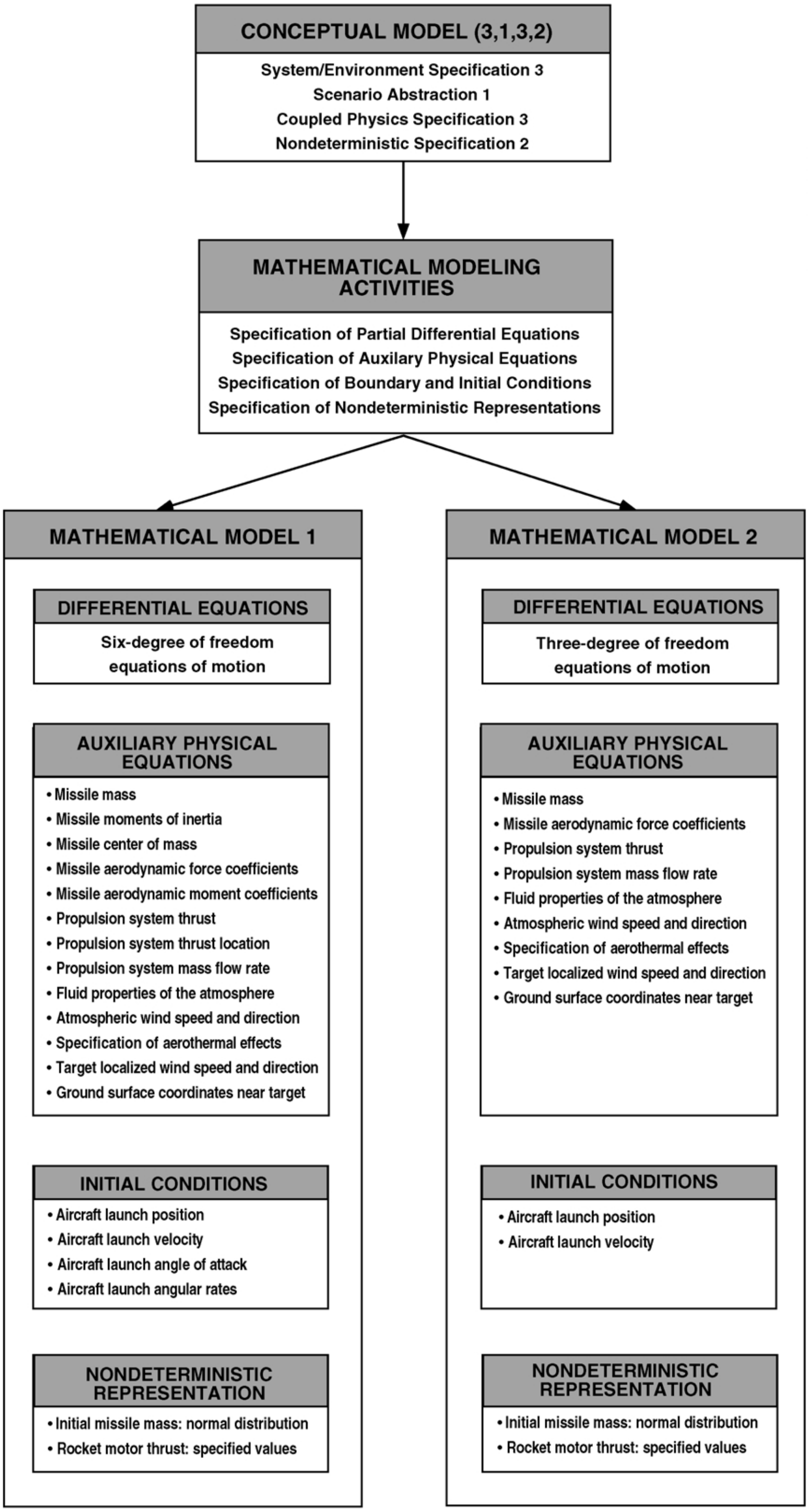

CONCEPTUAL MODEL (3,1,3,2)
System/Environment Specification 3
Scenario Abstraction 1
Coupled Physics Specification 3
Nondeterministic Specification 2
MATHEMATICAL MODELING ACTIVITIES
Specification of Partial Differential Equations
Specification of Auxilary Physical Equations
Specification of Boundary and Initial Conditions
Specification of Nondeterministic Representations
MATHEMATICAL MODEL 1
DIFFERENTIAL EQUATIONS
Six-degree of freedom equations of motion
AUXILIARY PHYSICAL EQUATIONS
• Missile mass
• Missile moments of inertia
• Missile center of mass
• Missile aerodynamic force coefficients
• Missile aerodynamic moment coefficients
• Propulsion system thrust
• Propulsion system thrust location
• Propulsion system mass flow rate
• Fluid properties of the atmosphere
• Atmospheric wind speed and direction
• Specification of aerothermal effects
• Target localized wind speed and direction
• Ground surface coordinates near target
INITIAL CONDITIONS
• Aircraft launch position
• Aircraft launch velocity
• Aircraft launch angle of attack
• Aircraft launch angular rates
NONDETERMINISTIC REPRESENTATION
• Initial missile mass: normal distribution
• Rocket motor thrust: specified values
MATHEMATICAL MODEL 2
DIFFERENTIAL EQUATIONS
Three-degree of freedom equations of motion
AUXILIARY PHYSICAL EQUATIONS
• Missile mass
• Missile aerodynamic force coefficients
• Propulsion system thrust
• Propulsion system mass flow rate
• Fluid properties of the atmosphere
• Atmospheric wind speed and direction
• Specification of aerothermal effects
• Target localized wind speed and direction
• Ground surface coordinates near target
INITIAL CONDITIONS
• Aircraft launch position
• Aircraft launch velocity
NONDETERMINISTIC REPRESENTATION
• Initial missile mass: normal distribution
• Rocket motor thrust: specified values

Fig. 6. Mathematical models for the missile flight example.

equations of motion can be written as

$$m\frac{\mathrm{d}\vec{\mathbf{V}}}{\mathrm{d}t} = \Sigma\vec{\mathbf{F}} \tag{1}$$

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

$$[\mathbf{I}]\frac{\mathrm{d}\vec{\boldsymbol{\omega}}}{\mathrm{d}t} = \Sigma\vec{\mathbf{M}} + \vec{\boldsymbol{\omega}} \times \{[\mathbf{I}]\cdot\vec{\boldsymbol{\omega}}\} \tag{2}$$

where [**I**] is the inertia tensor of the vehicle, $\vec{\boldsymbol{\omega}}$, the angular velocity and $\Sigma\vec{\mathbf{M}}$ 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. A description of the derivation of the 3-DOF and 6-DOF equations of motion are given in Ref. [62]. Although the 3-DOF and 6-DOF equations are ordinary differential equation (ODE) models instead of the PDE models stressed in the present work, key aspects of the present framework can still be exercised.

For the 3-DOF and 6-DOF 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, this ordering is commonly not possible for models of complex physical processes or systems. As a result, the mathematical model approximation would no longer be considered as an acknowledged error as in the present case, but an epistemic uncertainty.

Fig. 6 lists all the auxiliary equations and ICs that are needed for each mathematical model. As would be expected of higher-fidelity models, the 6-DOF model requires physical information well beyond that required by the 3-DOF model. This poses the question: When does the lack of information for the additional 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 to be considered in many analyses. It is fallacious, but unfortunately common, to claim that the higher the fidelity of the physics model, the better the results. The uncertainty of parameters and the greater computer resources required to solve higher-fidelity models are critical factors in estimating nondeterministic system response. In addition, constraints on computer resources can obviate the accuracy of a higher-fidelity model.

The two nondeterministic parameters considered in the missile flight example are 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 an epistemic uncertainty due to the unknown temperature of the solid propellant. We choose a nominal value and two bounding values: the normal operating temperature, the 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, i.e. there is no random variation in motor performance due to any other factors. It is clear that the epistemic uncertainty in propulsion thrust can be steadily reduced as information is added to the analysis. For example, if the temperature at launch is known precisely, the propulsion uncertainty could be eliminated, given the present assumptions.

3.4. Discretization and algorithm selection activities

The discretization and algorithm selection phase accomplishes two related activities. First, it converts the continuum mathematics model, i.e. the differential equations, into a discrete mathematics problem suitable for numerical solution. Secondly, it provides the methodology determining 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, not uncertainties, are the dominant loss-of-confidence issue in this phase (note that, for the remainder of the article, when we refer to 'errors', we will be referring only 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 [63,64]. Traditional nondeterministic methods applied to systems described by differential equations are commonly approximations to stochastic differential equations, even though this is rarely recognized. Stated differently, traditional nondeterministic methods are commonly a conversion of a stochastic differential equation into multiple solutions of deterministic differential equations. Since the discrete solution to stochastic differential equations is much less developed than the discrete solution for deterministic differential equations, this approximation is almost always made [65].

As shown in Fig. 4, we identify four activities in the discretization and algorithm selection phase: discretization of the PDEs, discretization of the BCs and ICs, 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 infinitesimal limit and the continuum form of the equations. Consistency of the discretization is normally evaluated by analytically proving that the numerical algorithm approaches the continuum equations as the discretization size of all independent variables approaches zero. For simple differencing methods, this evaluation is straightforward. For complex differencing schemes, such as essentially nonoscillatory (ENO) 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, for example: Are the conservation laws satisfied for finite spatial grid sizes or are mass, momentum and energy only conserved in the limit as grid size approaches zero? Does the numerical damping approach zero as the mesh size approaches zero? Note that the discretization of PDEs is also involved in the conversion of Neumann and Robin's, i.e. derivative, BCs into discrete equations. We have included the conversion of continuum ICs to discrete ICs because spatial singularities may be part of the ICs, not because there are derivatives involved. An example is the time-dependent decay of a vortex for which the IC is given as a singularity. Our point is also valid, indeed much more common, when singularities or discontinuities are specified as part of the BCs.

The selection of propagation methods and the design of computer experiments in Fig. 4 both address 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 aleatory and epistemic uncertainties through the computational phases of the analysis. Examples of methods for propagating aleatory uncertainties include reliability methods [25], sampling methods such as Monte Carlo or Latin Hypercube [66,67], and statistical design approaches [68]. Methods for the propagation of epistemic uncertainties represented by nontraditional theories, e.g. possibility theory and fuzzy sets, are a subject of current research [48,69–71]. The design of computer experiments is driven largely 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 other variables and parameters. The 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 estimate of uncertainty introduced into the analysis in this phase. This activity is performed here because the detailed specification of inputs and models will impact programming requirements, as well as the running of the computer model in the numerical solution phase. Selection of propagation methods and the design of computer experiments may be performed differently for different mathematical models and may involve the specification of probabilities associated with different model choices, where available information warrants specification of probabilities.

For the missile flight example, the same discretization method was applied to both the 6-DOF and 3-DOF mathematical models. This resulted in two discretized models that differ only in the differential equations being solved. A Runge–Kutta–Fehlberg (RKF) 4(5) method was chosen to solve each system of ODEs [72]. The RKF method is fifth-order accurate at each time step, and the integrator coefficients of Ref. [73] were used. The method provides an estimate of the local truncation error, i.e. truncation error 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. A more detailed description of the numerical integration procedure in given in Ref. [62].

The method chosen for propagation of the missile mass aleatory uncertainty was the Latin Hypercube Sampling (LHS) method. LHS employs stratified random sampling for choosing discrete values from a probabilistically defined nondeterministic variable or parameter. For propagation of the epistemic uncertainty of the motor performance, we simply chose three possible propulsion characteristics to bind the solution and provide an interval-valued result. The experimental design activity 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 two models and determining how results from these models might be combined to maximize the accuracy and efficiency of the computations.

3.5. Computer programming activities

Fig. 4 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), there is the subtler problem of undefined code behavior. Such behavior 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 that are unknown 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 analyst to use previously developed data handling and numerical analysis algorithms. Unfortunately, the analyst also inherits the undiscovered or undocumented errors in these libraries. There is also the possibility that when the analyst uses library routines he misunderstands or makes an error in the values passed to the library routines.

TAOS was the computer code used for the missile flight example [74]. This general-purpose flight dynamics code has been 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 because program execution was performed only on Unix workstations.

The capture and elimination of programming errors, while not generating much excitement with many researchers and engineering analysts, remains a major cost factor in producing highly verified software. Some researchers experienced only with model problems, do not appreciate the magnitude of the issue. They feel it is simply a matter of carelessness that can be easily remedied by software quality assurance practices. The high number of inconsistencies, static errors, and dynamic, i.e. run-time, errors in well tested *commercial* computer codes was recently investigated by Hatton [75]. He concluded “All the evidence … suggest that the current state of software implementations of scientific activity is rather worse that we would ever dare to fear, but at least we are forewarned”. Assessing software quality is becoming much more difficult because of massively parallel computers. In our opinion, the complexities of optimizing compilers for these machines, message passing, and 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. For example, the code does not execute identically, and does not produce the same numerical result, from one run to another. 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.

3.6. Numerical solution activities

As shown in Fig. 4, 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 numerical accuracy to which nonlinear algebraic, or transcendental, discrete equations are solved. Iterative convergence error normally occurs in two different procedures involved in obtaining the numerical solution: (1) during the iterative convergence that should 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 that should 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 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 adjusting or further specifying inputs that determine 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. Propagation convergence is determined by the change in the approximation from one computer run to the next. It is clear that the nondeterministic propagation convergence errors, as well as the errors discussed in the previous paragraph, are all acknowledged errors.

For the missile flight 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 models, 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. In the TAOS code, if the largest local truncation error of all the state variables is less than 0.6 of the error criterion, 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 between aleatory and epistemic 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, i.e. over the spatial domain of the PDE, are commonly made with finite element methods, but local error estimates are not usually available. For finite difference and finite volume methods, Richardson’s method can be used to estimate local truncation error [76,77]. However, this estimation method can become quite computationally expensive for complex problems.

3.7. Solution representation activities

In the solution representation phase shown in Fig. 4, 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 activity includes two types of similar activities: (1) the representation of individual solutions over the independent variables of the PDEs and (2) 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 postprocessor. Examples are oscillations of the continuous function between discrete solution points due to the use of a high-order polynomial function in the postprocessor, and interpolation of the discrete solution between multiblock grids such that mass, momentum and energy are not conserved. Note that we mean inaccurate construction with respect to the discrete solution, and 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, with the shock wave 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, the shock wave in the discrete representation is no longer discontinuous. As a result, the construction (solution representation) error should be judged with respect to the continuous function approximation of the discrete solution, and not the continuum model. The discontinuity of the shock wave was lost in the discretization step and could not be recovered here.

Representation of a nondeterministic simulation from the individual deterministic computer runs refers to the compilation of these multiple solutions into statistical measures that can be used to address the requirements of the analysis. This activity 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 that 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 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 misinterprets the individual or summary solutions. Examples of interpretation errors are (1) concluding that a computed solution is chaotic when it is not and (2) interpreting a computed flow as turbulent when it is only a spurious numerical solution [63,64]. Importantly, our definition of data 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

Fig. 7 illustrates the multiple models, numerical solutions and solution representations that are addressed in the missile flight example. As shown in the figure, six conceptual models are identified, many more are implied, but for illustration in this paper, only one is selected for further development and analysis. This single conceptual model spawns two alternative mathematical descriptions, the 3-DOF and 6-DOF models, both of which are carried through the remaining phases of the modeling and simulation process. For simplicity, Fig. 7 then shows the further development of only one of these mathematical models, although it is understood that identical development of Mathematical Model 1 is taking place in parallel with Mathematical Model 2. The discretization and programming phases identify alternative model choices that are not considered further in this example. 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 merged to represent the nondeterministic solution.

It is clear from the missile flight example that the modeling and simulation process for complex systems involves the identification and use of multiple scenarios, analyses and computations. At each phase of this process, it is often 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, as indicated in Fig. 7.

4. Missile flight example computational results

Before the computational results from the missile flight example are presented, it is necessary to provide a few details about the calculations. The missile is assumed to be launched from an aircraft flying straight and level at an altitude of 9.144 km (30 Kft) above sea level and at a speed of 213.4 m/s (700 ft/s). Assume a spherical, nonrotating earth. Define an earth-fixed, three-dimensional, Cartesian

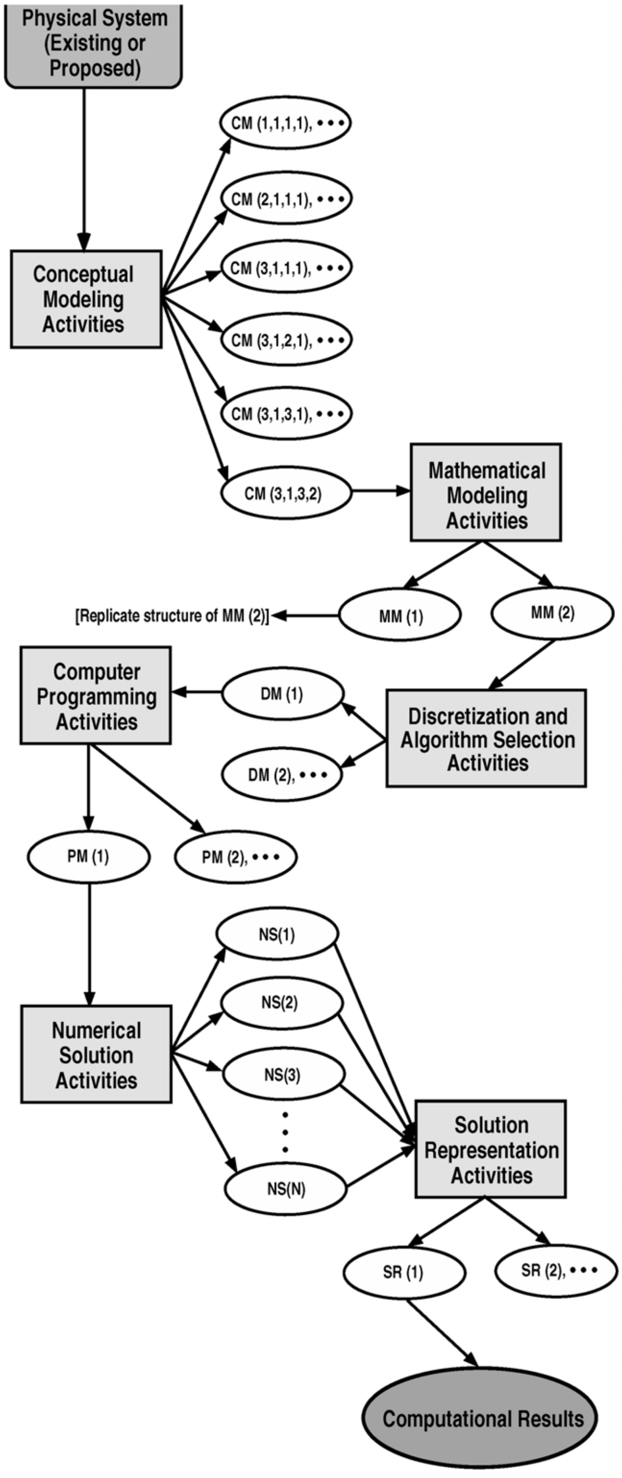

Physical System (Existing or Proposed)
Conceptual Modeling Activities
CM (1,1,1,1), • • •
CM (2,1,1,1), • • •
CM (3,1,1,1), • • •
CM (3,1,2,1), • • •
CM (3,1,3,1), • • •
CM (3,1,3,2)
Mathematical Modeling Activities
[Replicate structure of MM (2)]
MM (1)
MM (2)
Computer Programming Activities
DM (1)
DM (2), • • •
Discretization and Algorithm Selection Activities
PM (1)
PM (2), • • •
Numerical Solution Activities
NS(1)
NS(2)
NS(3)
NS(N)
Solution Representation Activities
SR (1)
SR (2), • • •
Computational Results

Fig. 7. Tree structure for models, solutions, and representations in the missile flight example.

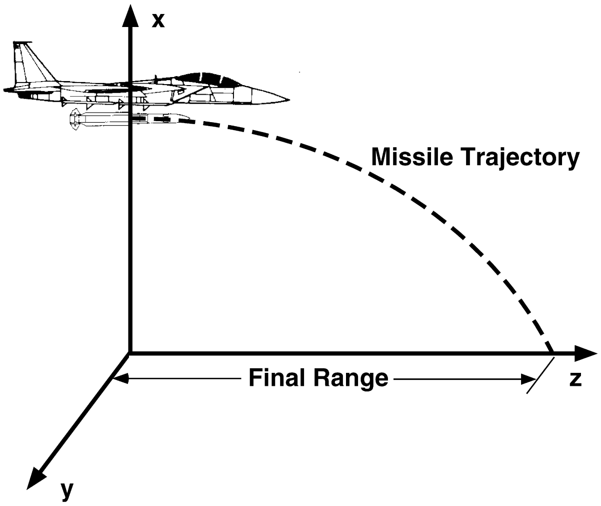


Fig. 8. Cartesian coordinate system for missile trajectory.

coordinate system, where x is vertical, z is in the direction of the aircraft flight, and y is normal to the xz plane (Fig. 8). Let the origin of the xyz coordinate system be at sea level, directly below the missile center-of-gravity at the IC. Assuming zero disturbance of the aircraft on the missile during launch and assuming uniform freestream flow approaching the missile, the ICs for the 6-DOF equations of motion are

$$x = 9.144\ \text{km}, \qquad y = z = 0, \qquad V_x = V_y = 0,$$

$$V_z = 213.4\ \text{m/s}, \qquad \alpha = \beta = \phi = 0, \qquad p = q = r = 0.$$

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 ICs 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 US Standard Atmosphere and that the winds are zero over the entire trajectory [78]. The trajectory calculation is terminated when $x = 0$, i.e. when the missile reaches sea level.

For convenience, detailed missile characteristics were taken to be those of the Improved Hawk missile, since these characteristics were readily available [79]. Missile moments of inertia, center of mass, rocket motor thrust and mass flow rate of the rocket motor are functions of time during operation of the rocket motor but are constant after the motor's burnout. The rocket motor nominally operates for 24.5 s, which is about half of the total flight time of the missile. The aerodynamic force and moment coefficient derivatives are assumed constant with pitch, yaw and roll angle of the missile, i.e. linear aerodynamics is assumed. However, the aerodynamic force and moment coefficient derivatives are functions of Mach number. The only system response measure presented here is the final range of the missile, since it captures most of the trajectory characteristics of interest. Detailed information on the missile moments of inertia, center of mass, rocket motor thrust, mass flow rate of the rocket motor, aerodynamic force and moment coefficients, and other flight dynamics characteristics are given in Ref. [62].

To illustrate the combined effects of aleatory and epistemic uncertainty 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, three thrust models, and five selected values of numerical integration solution error. Our purpose is to study and understand the effects and interactions of these sources of uncertainty and error on the system response measure of interest.

4.1. *Effects of mass uncertainty*

The first source of nondeterministic behavior examined was the aleatory uncertainty of the initial mass of the missile. The mean initial mass was 625.49 kg (1378.98 lb), of which 332.01 kg (731.96 lb) was inert mass and 293.48 kg (647.02 lb) was propellant. As mentioned in Section 3.3, a normal probability distribution for initial mass uncertainty was assumed. The standard deviation, σ_w, was assumed to be 4.54 kg (10 lb). Although it is not important for this problem, $\sigma_w = 4.54$ kg, is consistent with actual missile systems of this size [80]. 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 uncertainty 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 criteria 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 and 1 s, for one 6-DOF and one 3-DOF solution, respectively.

Since computer run time was not an issue, we computed 500 LHS solutions for both the 6-DOF and 3-DOF models. Fig. 9 shows the histogram from the LHS, centering the mass at 625.5 kg (1379 lb) and using bins of width 2.27 kg (5 lb). 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 625.497 kg (1378.984 lb), and $\sigma_w = 4.533$ kg (9.993 lb). The 500 samples are roughly a factor of 10 higher than is normally needed. We chose this large number to 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 masses. Indeed, for all results given in this analysis, exactly the same sampled initial missile masses were used.

Fig. 10 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 mass. It is clear from the very well behaved system response measure that 5–10 LHS samples would have been sufficient to characterize this response measure. However, for our analysis, we required sampling errors that were much less than typical analyses. It is also seen in Fig. 10 that the lower-fidelity model (3-DOF) introduces a bias error of 74 m (243 ft) in range, which is constant for all masses sampled. The generation of a bias error in the response of the system is disturbing because this error might go undetected if the higher-fidelity model results or experimental measurements were not available. In general, one does not expect this result. Lower-fidelity models are used with the hope that the computational results will be distributed around the correct answer and not contain a *systematic* error in response. 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 vertical plane causes a small mean angle-of-attack during most of the trajectory. Computational results from the 6-DOF trajectory show this value to be about 0.01–0.02° after the initial disturbance at launch decays. This angle of attack causes a lift component on the missile, i.e. a small

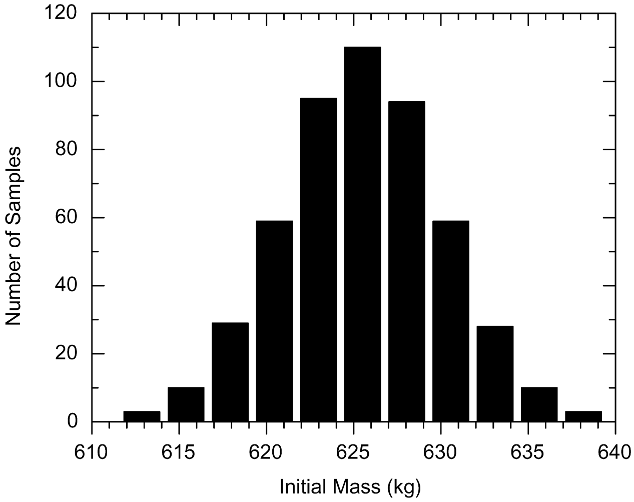


Fig. 9. Histogram from LHS for mass uncertainty.

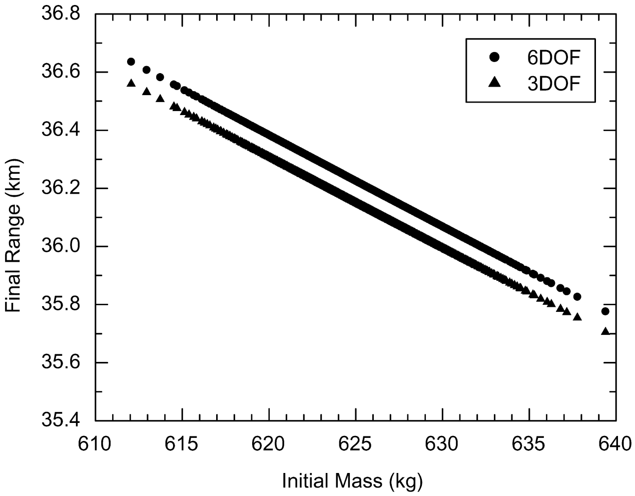


Fig. 10. Uncertainty in range due to uncertainty in initial mass.

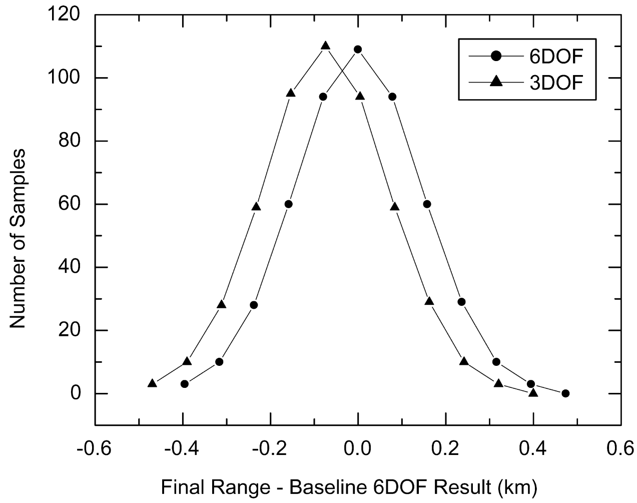


Fig. 11. Frequency data from LHS for range offset due to initial mass.

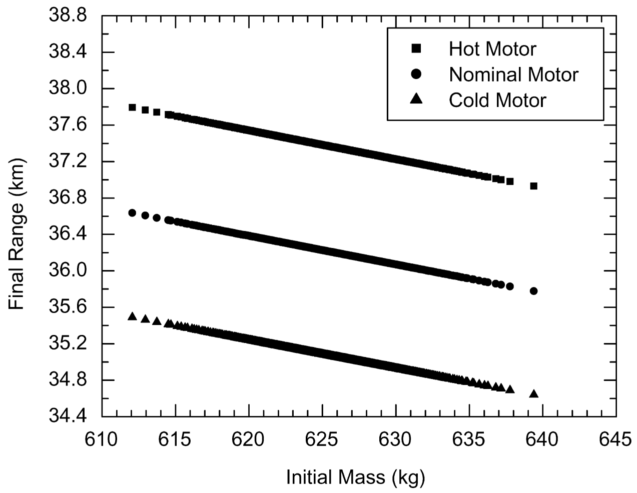


Fig. 12. Uncertainty in range due to thrust and mass uncertainty for 6-DOF model.

gliding effect, which results in a slightly longer trajectory. The lower-fidelity model does not, indeed cannot, 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 aerodynamic lift coefficient.

Fig. 11 shows frequency data of the LHS samples as a function of range offset from the mean value range for the 6-DOF trajectory: 36.210 km (19.552 nm). That is, the range computed for the mean mass of 625.49 kg (1378.98 lb) for the 6-DOF trajectory is defined to have zero offset. In this figure, the bias error in range of 74 m (243 ft) 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. 10. For the 6-DOF model, the standard deviation in range is computed to be 143 m (468 ft), whereas for the 3-DOF model, $\sigma_R = 142$ m (466 ft).

4.2. *Effects of thrust uncertainty*

As discussed in Section 3.3, our approach to determining the uncertainty in the trajectories due to unknown 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 [81]. At the highest allowed temperature of 49°C (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 −29°C (−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.

Fig. 12 shows the 6-DOF computed range of the missile for each of the three temperature conditions of the motor as a function of initial mass uncertainty. It can be seen from Fig. 12 that, as expected, the motor temperature uncertainty produces a shift in range: the high-temperature motor flying 1.16 km (0.625 nm) further than the nominal motor temperature, and the cold motor flying 1.14 km (0.616 nm) shorter than the nominal motor. The linearity of the missile range as a function of mass 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 mass uncertainty. The uncertainty in

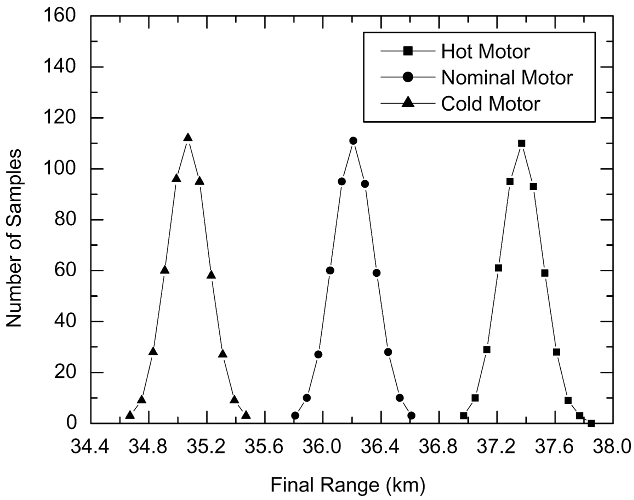


Fig. 13. Frequency data from LHS for range uncertainty due to thrust uncertainty for 6-DOF model.

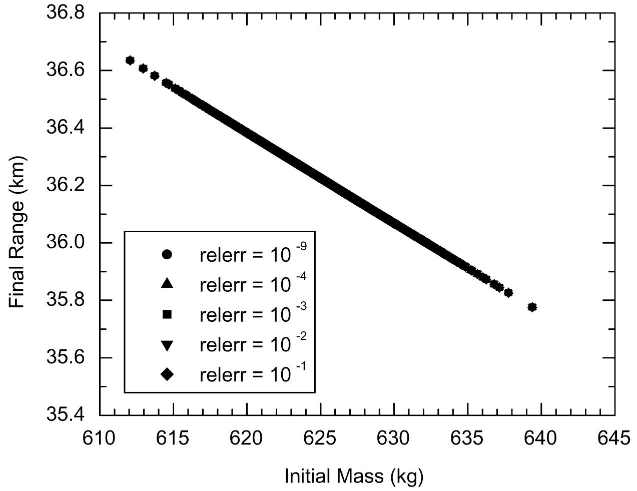


Fig. 14. Uncertainty in range due to solution error and mass uncertainty for 6-DOF model.

range due to unknown rocket-motor temperature is 2.30 km (1.24 nm). The uncertainty in range due to mass uncertainty can be calculated as $4\sigma_R = 4 \times 0.143$ km $= 0.572$ km (0.308 nm), which is only 25% of the uncertainty due to thrust.

Fig. 13 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 1.14 km (0.616 nm) toward shorter range, whereas the hot motor is shifted about the same amount toward longer range. The standard deviation in range for the hot and cold motors is nearly identical: $(\sigma_R)_{hot} =$ 143 m (470 ft) and $(\sigma_R)_{cold} = 141$ m (463 ft). Recall that these values are essentially the same as the value of the nominal motor, $(\sigma_R)_{nom} = 143$ m (468 ft). 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 74 m (243 ft) bias in range, as discussed in Section 4.1.

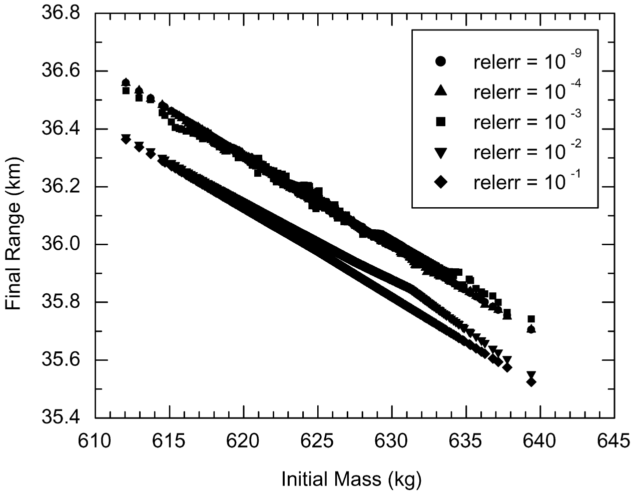


Fig. 15. Uncertainty in range due to solution error and mass uncertainty for 3-DOF model.

We argue that the source of the 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 aleatory uncertainty instead of epistemic uncertainty. The argument is that a probability distribution could be constructed based on measuring motor temperatures for a large number of actual missile deployments, experimentally. The uncertainty 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 aleatory uncertainty approach could lead to misleading estimates of system performance for certain deployment situations. For example, if the deployment were in Alaska during the winter versus Saudi Arabia during the summer, the mean range of the missile would be of little value. Additional knowledge of the type of deployment would change the representation. A deployment at a permanent installation with significant environmentally controlled space would be quite different from a makeshift battlefield deployment. As more and different kinds of knowledge are introduced into the analysis, representations other than precise probability distributions may be more appropriate, e.g. belief and plausibility measures in Dempster–Shafer theory. Belief and plausibility measures state the lower and upper bounds, respectively, of the imprecise probabilitistic information. However, guidance on developing these representations based on available information is poorly developed compared to probability theory.

4.3. Effects of numerical integration error

As discussed in Section 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 RKF 4(5) method, and the time step is adjusted at each step so that the truncation error is less than the specified error criterion. Fig. 14 shows the computed range of the missile for the 6-DOF model using the nominal thrust profile as a function of the mass uncertainty 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 result was not expected. Experience leads us to believe that as the error criterion increased greatly, the accuracy of the solution would degrade progressively. For certain state variables, like 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 temporal 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 pitch rate q, and yaw rate r. Both q and r have a frequency of 1–2 Hz, which limits the maximum time step to 0.1–0.2 s 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.

Fig. 15 shows the 3-DOF computed range using the nominal thrust profile as a function of mass uncertainty for five per-step, relative truncation-error criteria. These five error criteria are the same as those used in the 6-DOF calculation illustrated in Fig. 14. 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}, a slight roughness in the range as a function of mass can be seen. For a 10^{-3} error, the amplitude in roughness of range increases to 65 m (213 ft). This variation in amplitude can occur over a very small change in mass. For example, near the mean mass of 625.49 kg (1378.99 lb), a jump of 65 m can be seen over a change in mass of less than 0.05 kg. This type of unrealistic 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.

As the numerical error is increased further, to 10^{-2} and 10^{-1}, Fig. 15 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 mass for 10^{-2} error but becomes constant at a value of 185 m (608 ft) for 10^{-1} error. In addition, the range becomes an extraordinarily smooth function of mass, with 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 s (all solutions presented attempt to use $\Delta t = 0.1$ s in stepping from $t = 0$). If the step size results in an estimated truncation error that satisfies the error criterion, the step is accepted. If the estimated error is 0.6, or less of the error criterion, the time step is accepted and the time step is increased for the next step. If it does not meet with the error criterion, the time step is decreased until the error criterion is met. For the 3-DOF model, the time step increases rapidly because all state variables are extremely smooth as a function of time, relative to the 6-DOF model. When the error criterion is changed from 10^{-4} to 10^{-3} Fig. 15, there is a rapid loss in accuracy of the major physical characteristic of the 3-DOF trajectory: the motor thrust profile. From the IC until 4.5 s, the motor thrust is roughly 84,000 N (19,000 lb). Then, it rapidly drops to a sustained thrust value of about 16,000 N (3600 lb) for 20 s, after which thrust terminates. For error criteria less than 10^{-5}, the numerical solution 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. Summary comments

Probably the most surprising computational results obtained in the missile flight example are those related to the aggregation and interaction of numerical solution error with uncertainty. The counterintuitive result that the higher fidelity model is much less sensitive to solution error than the lower-fidelity model needs further comment. The discussion given previously regarding the controlling factor in solution error for each model explains 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 and 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 nonadaptive grid-generation methods, which is the norm for the solution of the PDEs, very different sensitivities would occur than those observed here. Nonadaptive grid methods would be analogous to a constant time-step method in the solution of ODEs. For the present example, 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 mass and nominal thrust. As the time step increases, the numerical error for both models increases, but the error in the 6-DOF model 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

Table 1
Error in range for 6-DOF and 3-DOF for constant time steps

Time step (s)	0.001	0.01	0.07	0.09	0.1
6-DOF (m)	0.0	7	70	736	∞
3-DOF (m)	0.0	7	73	96	107

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.

5. Summary and conclusions

We have presented a comprehensive, new framework for modeling and simulation that blends the perspective of three technical communities: the systems view from the operations research community, propagation of uncertainty from the risk assessment community, and the numerical solution of PDEs from the computational physics community. The activities that are conducted in each of the six phases of modeling and simulation are discussed in detail. Consistent with recent work in the risk assessment community, we carefully define and distinguish between aleatory and epistemic uncertainty. In addition, we define and discuss acknowledged and unacknowledged errors. In each of the activities in each phase of modeling and simulation, we discuss which type of source (aleatory or epistemic uncertainty or error) typically dominates the activity. Particular emphasis is given to distinguishing the continuous and discrete mathematical modeling activities and to the nondeterministic features of the analysis. Our framework applies regardless of whether the discretization procedure for solving the PDEs is based on finite elements, finite volumes, or finite differences.

The formal distinction between aleatory and epistemic uncertainty in this framework drives one toward different mathematical representations for each: probabilistic representations for aleatory uncertainty, and various other modern information theories for representation of epistemic uncertainty. One approach that has been used for epistemic uncertainty is Bayesian probability. This approach takes a subjective view of probability as a measure of degree of belief in a hypothesis. Although we believe this is a step in the right direction to represent epistemic uncertainty, we do not believe it is satisfactory. We recommend research into modern theories of uncertainty-based information, such as possibility theory, evidence (Dempster/Shafer) theory, fuzzy set theory, and imprecise probability theory. However, these theories are not well developed when compared to probabilistic inference. In addition, none of these theories, except fuzzy set theory, has been applied to engineering analysis problems. If one were to take the step and represent aleatory uncertainty probabilistically, and epistemic uncertainty with one of the new theories, then one must face the question of propagating these components concurrently in the modeling and simulation process. They are not simply or uniquely combinable. Propagation methods of this type are even more of a research topic.

We believe that the usefulness of the present framework results from two aspects. First, it formalizes and merges a broad range of activities conducted in complex system modeling and modern computational simulation. It collects into one picture all activities so that each one can be clearly distinguished, relationships can be unambiguously depicted and assumptions can be formalized. The framework can be viewed as a many-branched event-and-decision tree and, as such, the connection and propagation of scenarios, uncertainties, and modeling decisions and assumptions are unequivocal. Second, the framework identifies how and where sources of aleatory and epistemic uncertainty and error contribute to the modeling and simulation process. For the analysis of complex systems, this formal recognition of sources of uncertainty and error shows the compounding effect and rapid growth of each source through the modeling and simulation process. Some have referred to this growth of source combinations through the modeling and simulation process as ‘overwhelming’. However, it is an issue that must be faced by analysts and decision makers who use the results of modeling and simulation.

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 Jon Helton, Rob Easterling, Tim Trucano and Vicente Romero of Sandia for their comments and suggestions in reviewing an earlier version of this article.

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

References

[1] Beck MB. Water quality modeling: a review of the analysis of uncertainty. Water Resour Res 1987;23:1393–442.

[2] Bogen KT, Spear RC. Integrating uncertainty and interindividual variability in environmental risk assessment. Risk Anal 1987;7:427–36.

[3] Vamos T. Epistemic background problems of uncertainty. Presented at First Int Symp Uncertainty Model Anal, College Park, MD, 1990.

[4] Morgan MG, Henrion M. Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis. 1st ed. New York: Cambridge University Press, 1990.

[5] Hoffman FO, Hammonds JS. Propagation of uncertainty in risk assessments: the need to distinguish between uncertainty due to lack of knowledge and uncertainty due to variability. Risk Anal 1994;14:707–12.

[6] Helton JC. Treatment of uncertainty in performance assessments for complex systems. Risk Anal 1994;14:483–511.

[7] Rowe WD. Understanding uncertainty. Risk Anal 1994;14:743–50.

[8] Hora SC. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. Reliab Engng Syst Safety 1996;54:217–23.

[9] Frey HC, Rhodes DS. Characterizing, simulating, and analyzing variability and uncertainty: an illustration of methods using an air toxics emissions example. Human Ecol Risk Assess 1996;2:762–97.

[10] Ferson S, Ginzburg LR. Different methods are needed to propagate ignorance and variability. Reliab Engng Syst Safety 1996;54:133–44.

[11] NCRP. A guide for uncertainty analysis in dose and risk assessments related to environmental contamination. National Council on Radiation Protection and Measurements, Bethesda, MD NCRP Commentary No. 14 May 10, 1996.

[12] Frank MV. Treatment of uncertainties in space nuclear risk assessment with examples from Cassini mission applications. Reliab Engng Syst Safety 1999;66:203–21.

[13] Zeigler BP. Multifaceted modelling and discrete event simulation. 1st ed. Orlando: Academic Press, 1984.

[14] Neelamkavil F. Computer simulation and modelling. 1st ed. New York: Wiley, 1987.

[15] Law AM, Kelton WD. Simulation modeling and analysis. 2nd ed. New York: McGraw-Hill, 1991.

[16] Bossel H. Modeling and simulation. 1st ed. Wellesley, MA: A.K. Peters, Ltd, 1994.

[17] Banks J. Handbook of simulation. New York: Wiley, 1998.

[18] Zeigler BP, Praehofer H, Kim TG. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. 2nd ed. San Diego, CA: Academic Press, 2000.

[19] Schlesinger S. Terminology for model credibility. Simulation 1979;32:103–4.

[20] Jacoby SLS, Kowalik JS. Mathematical modeling with computers. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[21] Sargent RG. Simulation model validation. In: Oren TI, Zeigler BP, Elzas MS, editors. Simulation and model-based methodologies: an integrative view, Berlin: Springer, 1984. p. 537–55.

[22] Sargent RG. An expository on verification and validation of simulation models. Presented at Winter Simulation Conference, Sacramento, CA, 1985.

[23] Nance RE. Model representation in discrete event simulation: the conical methodology. Virginia Polytechnic Inst State University Dept Computer Science, Technical Rep CS81003-R, 1981.

[24] Balci O. Guidelines for successful simulation studies. Presented at Proc 1990 Winter Simul Conf, New Orleans, LA, 1990.

[25] Ang AHS, Tang WH. Probability concepts in engineering planning and design: vol. II decision, risk, and reliability. New York: Wiley, 1984.

[26] Hauptmanns U, Werner W. Engineering risks evaluation and valuation. 1st ed. Berlin: Springer, 1991.

[27] Modarres M. What every engineer should know about reliability and risk analysis. New York: Marcel Dekker, 1993.

[28] Kumamoto H, Henley EJ. Probabilistic risk assessment and management for engineers and scientists. 2nd ed. New York: IEEE Press, 1996.

[29] Davis PA, Price LL, Wahi KK, Goodrich MT, Gallegos DP, Bonano EJ, Guzowski RV. Components of an overall performance assessment methodology. Sandia National Laboratories. Albuquerque, NM NUREG/CR-5256; SAND88-3020, February 1990.

[30] Davis PA, Bonano EJ, Wahi KK, Price LL. Uncertainties associated with performance assessment of high-level radioactive waste repositories. Sandia National Laboratories, Albuquerque, NM NUREG/CR-5211; SAND88-2703, November 1990.

[31] Ang AHS, Tang WH. Probability concepts in engineering planning and design: vol. I basic principles. 1st ed. New York: Wiley, 1975.

[32] Giere RN. Knowledge, values, and technological decisions: a decision theoretic approach. In: Mayo DG, Hollander RD, editors. Acceptable evidence: science and values in risk management, environmental ethics and science policy series, New York: Oxford University Press, 1991. p. 183–203.

[33] Apostolakis G. A commentary on model uncertainty. Presented at Proc Workshop I Adv Topics Risk Reliab Anal — Model Uncertainty: Its Character Quantification, Annapolis, MD, 1994.

[34] Haimes YY, Barry T, Lambert JH. When and how can you specify a probability distribution when you don't know much?. Risk Anal 1994;14:661–706.

[35] Ayyub BM. The nature of uncertainty in structural engineering. In: Ayyub BM, Gupta MM, editors. 1st ed. Uncertainty modelling and analysis: theory and applications, vol. 17. New York: Elsevier, 1994. p. 195–210.

[36] Ferson S. What Monte Carlo methods cannot do. Human Ecol Risk Assess 1996;2:990–1007.

[37] Helton JC, Burmaster DE. Guest editorial: treatment of aleatory and epistemic uncertainty in performance assessments for complex systems. Reliab Engng Syst Safety 1996;54:91–94.

[38] Rai SN, Krewski D, Bartlett S. A general framework for the analysis of uncertainty and variability in risk assessment. Human Ecol Risk Assess 1996;2:972–89.

[39] Parry GW. The characterization of uncertainty in probabilistic risk assessments of complex systems. Reliab Engng Syst Safety 1996;54:119–26.

[40] Paté-Cornell ME. Uncertainties in risk analysis: six levels of treatment. Reliab Engng Syst Safety 1996;54:95–111.

[41] Ayyub BM. Uncertainty modeling and analysis in civil engineering. Boca Raton, FL: CRC Press, 1998.

[42] Cullen AC, Frey HC. Probabilistic techniques in exposure assessment: a handbook for dealing with variability and uncertainty in models and inputs. New York: Plenum Press, 1999.

[43] Klir GJ, Wierman MJ. Uncertainty-based information: elements of generalized information theory, vol. 15. Heidelberg: Physica-Verlag, 1998.

[44] Cox E. The fuzzy systems handbook. New York: AP Professional, 1998.

[45] Dubois D, Prade H. Possibility theory: an approach to computerized processing of uncertainty. New York: Plenum Press, 1986.

[46] de Cooman G, Ruan D, Kerre EE. In: Hirota K, Klir GJ, Sanchez E, Wang P-Z, Yager RR, editors. Foundations and applications of possibility theory, Singapore: World Scientific, 1995.

[47] Guan J, Bell DA. Evidence theory and its applications, vol. I. Amsterdam: Elsevier, 1991.

[48] Almond RG. Graphical belief modeling. 1st ed. London: Chapman & Hall, 1995.

[49] Klir GJ, Folger TA. Fuzzy sets, uncertainty, and information. 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[50] Cox E. The fuzzy systems handbook: a practitioner's guide to building, using, and maintaining fuzzy systems. 2nd ed. San Diego, CA: AP Professional, Division of Academic Press, 1999.

[51] Walley P. Statistical reasoning with imprecise probabilities. London: Chapman & Hall, 1991.

[52] Kozine I. Imprecise probabilities relating to prior reliability assessments. Presented at First Int Symp Imprecise Probab Appl, Ghent, Belgium, 1999.

[53] Box GEP, Tiao GC. Bayesian inference in statistical analysis. New York: Wiley, 1992.

[54] Berger JO. Statistical decision theory and Bayesian analysis. Berlin: Springer, 1985.

[55] Oberkampf WL, Diegert KV, Alvin KF, Rutherford BM. Variability, uncertainty, and error in computational simulations. Presented at AIAA/ASME Joint Thermophys Heat Transfer Conf, Albuquerque, NM, 1998.

[56] Oberkampf WL, DeLand SM, Rutherford BM, Diegert KV, Alvin KF. A new methodology for the estimation of total uncertainty in computational simulation. Presented at AIAA/ASME/ASCE/AHS/ASC Struct, Struct Dyn, Mater Conf Exhibit, St. Louis, MO, 1999.

[57] Box GEP. Sampling and Bayes inference in scientific modeling and robustness. J Statist Soc A 1980;143:383–430.

[58] Lewis RO. Independent verification and validation. 1st ed. New York: Wiley, 1992.

[59] Knepell PL, Arangno DC. Simulation validation: a confidence

assessment methodology. 1st ed. Washington, DC: IEEE Computer Society Press, 1993.

[60] Modarres M, Kaminskiy M, Krivtsov V. Reliability engineering and risk analysis. New York: Marcel Dekker, 1999.

[61] Tenner E. Why things bite back. New York: Alfred A. Knopf, 1996.

[62] Oberkampf WL, DeLand SM, Rutherford BM, Diegert KV, Alvin KF. Estimation of total uncertainty in computational simulation. Sandia National Laboratories, Albuquerque, NM SAND2000-0824, 2000.

[63] Yee HC, Sweby PK. Global asymptotic behavior of iterative implicit schemes. Int J Bifurcation Chaos 1994;4:1579–611.

[64] Yee HC, Sweby PK. Aspects of numerical uncertainties in time marching to steady-state numerical solutions. AIAA J 1998;36:712–24.

[65] Kloeden PE, Platen E. Numerical solution of stochastic differential equations. New York: Springer, 1992.

[66] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 1979;21:239–45.

[67] Iman RL, Conover WJ. A distribution-free approach to introducing rank correlation among input variables. Commun Statist, Simul Comput 1982;11:311–34.

[68] Box EP, Draper NR. Emperical model-building and response surfaces. New York: Wiley, 1987.

[69] Klir GJ. Probabilistic versus possibilistic conceptualization of uncertainty. In: Ayyub BM, Gupta MM, Kanal LN, editors. Analysis and management of uncertainty: theory and applications, vol. 13. New York: North-Holland, 1992. p. 13–26.

[70] Bier VM. Fuzzy set theory, probability theory, and truth functionality. In: Ayyub BM, Gupta MM, Kanal LN, editors. Analysis and management of uncertainty: theory and applications, vol. 13. New York: North-Holland, 1992. p. 65–78.

[71] Muhanna RL, Mullen RL. Development of interval based methods for fuzziness in continuum mechanics. Presented at Third Int Symp Uncertainty Model Anal Ann Conf North American Fuzzy Info Process Soc, College Park, MD, 1995.

[72] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes in FORTRAN. New York: Cambridge University Press, 1992.

[73] Cash JR, Karp AH. A variable order Runge–Kutta method for initial-value problems with rapidly varying right-hand sides. ACM Trans Math Software 1990;16:201–22.

[74] Salguero DE. Trajectory analysis and optimization software (TAOS). Sandia National Laboratories, Albuquerque, NM SAND99-0811, May 1999.

[75] Hatton L. The T experiments: errors in scientific software. IEEE Comput Sci Engng 1997;4:27–38.

[76] Fletcher CAJ. Computational techniques for fluid dynamics, vol. 1. New York: Springer, 1988.

[77] Roache PJ. Verification and validation in computational science and engineering. Albuquerque, NM: Hermosa Publishers, 1998.

[78] NOAA, NASA and USAF. US standard atmosphere. Washington, DC: US Government Printing Office, 1976.

[79] Rollstin LR. Personal communication, 1998.

[80] Rollstin LR, Fellerhoff RD. Aeroballistic and mechanical design and development of the Talos-Terrier-Recurit (Tater) rocket system with flight test results. Sandia National Laboratories, Albuquerque, NM SAND74-0440, February 1976.

[81] CPIA. Rocket motor manual. Chemical Propulsion Information Agency, Laurel, MD, CPIA/M1, 1982.