3

Modeling and computational simulation

The phrases *modeling and simulation* and *computational simulation* are becoming prevalent in a wide variety of technical, economic, governmental, and business activities (Schrage, 1999). Indeed, the phrases are becoming so common that one is even beginning to see them in the mass media. What do they mean? These phrases can have a wide variety of meanings depending on the field and the context. Here, we are concerned with the fields of the physical sciences and engineering. By examining the fundamentals of modeling and simulation (M&S) and scientific computing, our goal is to see the similarities in model formulation and computational issues across a wide range of physical systems. Our approach to scientific computing emphasizes the similarities that exist in mathematical form and structure of models in many technical disciplines. Then a framework, an overarching structure, is constructed for either attacking more detailed features of the system or for attacking more complex systems. Commonly, more complex systems involve coupling different types of physical phenomena, incorporation of additional elements from the system or the surroundings, and effects of human intervention in the system. Similarities in model formulation issues exist because many of the model properties are not determined by their physical nature, but by their mathematical structure, the interaction of the system with the surroundings, and the similarity in the nature of the system responses.

Many of the difficult issues that must be dealt with in verification, validation, and uncertainty quantification (VV&UQ) can be traced back to ambiguities and inconsistencies in the model formulation, the mapping of continuum mathematics models to discrete mathematics models, and vague or improper characterizations of uncertainties. This chapter deals with many of those issues by examining the fundamentals of M&S.

We begin by carefully defining and discussing the terms system, surroundings, environments, and scenarios. We discuss the importance of constructing models such that they can produce nondeterministic simulations. To help clarify this concept, we discuss an example for the nondeterministic oscillation of a simple mechanical system. We then discuss the six phases of computational simulation: conceptual modeling, discretization and algorithm selection, computer programming, numerical solution, and solution representation. We close the chapter with a detailed example of the flight dynamics of a missile that demonstrates each of these six phases.

3.1 Fundamentals of system specifications

3.1.1 Systems and surroundings

The concept and understanding of the meaning of a system is probably the most important element in modeling. The definition of a system that is most useful for modeling physical systems is:

System: a set of physical entities that interact and are observable, where the entities can be a specified quantity of matter or a volume in space.

For those with some background in thermodynamics, this definition is similar to that given for a system in thermodynamics. The stress in this definition is on physical entities that can interact and are observable. As in thermodynamics, this definition allows for the system to be closed or open. A closed system means there is no exchange of mass with the surroundings of the system. An open system can have mass flow into and out of the system. A system can have forces acting on it, work done on it, and energy exchanged with the surroundings. Also, a system can be time-invariant (static) or time-variant (dynamic).

Our definition of a system, although very broad, is actually more restrictive than that used in many fields, for example, Operations Research (Neelamkavil, 1987). Our definition excludes human organizations, governments, societies, economies, and human mental processes. However, these are valid topics of modeling in many fields. All of these types of entity can be considered as living or sentient entities and are, by almost any measure, much more complex entities than physical systems of interest here. In using the present definition of a system, however, the physical body of a person, or any part or organ of the body, could be considered as a system. For example, the physiological, mechanical, and chemical changes of an organ exposed to various wavelengths of the electromagnetic spectrum could be considered within our definition of a system.

The state of a system can be influenced as a result of (a) processes internal to the system, i.e., endogenous processes, and (b) processes or activities external to the system, i.e., exogenous effects. Influences or activities not considered as part of the system are considered as part of the surroundings of the system. Since the complement to the system is the surroundings, it is important that a precise definition be given:

Surroundings: all entities and influences that are physically or conceptually separate from the system.

A system is influenced by the surroundings, but the surroundings are *not* modeled as part of the system (Neelamkavil, 1987). In most models, the system *responds* to the surroundings, but the surroundings are *not* influenced by the system. In rare modeling situations, the surroundings can be influenced by the system. When this occurs, one of two possible modeling changes must occur.

- A separate mathematical model is constructed for the surroundings. Then the surroundings become another system that interacts with the first system.

- A weak coupling is constructed between the system and the surroundings. The surroundings are not considered another system, but they can respond in very simple, specific ways. That is, the weak coupling is dependent on the response of the system, typically as a result of some type of experimentally observed correlation function that represents how the surroundings respond to specific processes modeled within the system.

The distinction between the system and the surroundings should not be simply thought of as a physical boundary, or location outside the physical system. The distinction between the system and the surroundings can be entirely conceptual. The decision of what physical elements and features should be considered as part of the system, and what should be considered part of the surroundings, depends on the purpose of the analysis. System-surroundings specifications are not always well thought out and, as a result, they can cause modeling errors or conceptual inconsistencies in the formulation of a mathematical model. Finally, humans can be a conceptual part of a system. When a human is part of the system, they are referred to as an *actor* (Bossel, 1994). By *actor* we mean an element of a system that can influence the system in some physical way, or respond to events or activities occurring in the system in a conscious manner. *Conscious manner* usually means with a goal or purpose in mind, but it does not necessarily mean what would normally be considered rational or logical. The actor can also be unpredictable, unreliable, or acting with an unknown value system or some malicious agenda. Actors can become important elements in many complex physical systems; for example, systems that are human controlled or controlled by a combination of a human and a computer, such as a safety control system. Another example is accidental or unanticipated human involvement in a system that is normally thought of as isolated from humans or under computer control.

A few examples of systems and surroundings are in order to help clarify these concepts:

Example 1: Orbiting spacecraft

Consider a spacecraft in orbit around the Earth as a system. Assume the system behavior of interest is the three-degree-of-freedom orbital dynamics of the spacecraft. The spacecraft would be considered as the system and the primary characteristics of the system would be the mass and velocity of the craft. The surroundings would be represented by the forces acting on the craft: (a) the gravitational force of the Earth, Moon, Sun, and other planets; (b) the aerodynamic or molecular drag on the craft; and (c) the solar wind and electrostatic forces on the vehicle. If the spacecraft has a thrust control system onboard to change its orbital parameters, then the force exerted by the thrusters on the spacecraft would be part of the *surroundings*. However, the mass of the thrusters and their propellants are part of the system. As the thrusters are fired, the mass of the system would change due to consumption of propellants. As a result, the system is actually an open system since mass is leaving the system.

Example 2: Beam deflection

Consider the deflection of a beam clamped on one end and free on the other. Assume the system behavior of interest is the static deflection of the beam under a specified loading. The mass, material, and geometric properties of the beam would be considered as the system. The surroundings would be the static load distribution and how the clamped end affects the deflection of the beam. For example, the clamped end may be assumed to be perfectly rigid, i.e., no deflection or rotation occurs at the clamped end. Alternatively, the clamped end could be considered to have no translational deflection, but rotational deflection occurs around three orthogonal axes. For example, three rotational spring stiffnesses could be used to represent the clamped end as part of the surroundings. The system is influenced by the surroundings, but the surroundings are not influenced by the system. One could add fidelity and complexity to the model by including a first order approximation concerning how the clamped end could lose some of its rotational stiffness as a function of the number of deflection cycles of the beam. If this complexity were added to the model, it would be based on a combination of the predicted motion of the beam, and on a correlation of data from observed experiments of how the clamped end lost stiffness as a function of the number of deflection cycles. For this case, however, the clamp is still part of the surroundings, but the surroundings could change in a very specific way due to processes occurring within the system.

Example 3: Electronic circuit

Consider the electronic circuitry of a common television set as a system. Assume the physical behavior of interest of the system is the current flow through all of the electrical components of the television circuitry when the TV is switched on. The initial state of the system is considered as that before it is switched on. In addition, assume that the TV is plugged into an electrical power outlet before it is switched on. That is, electrical power is applied to certain parts of the circuitry, but not all, in what is commonly called a *stand-by* or *ready* mode of the circuit. The final state of the system is considered to be the current flow in the circuitry after the TV is switched on for some time. One type of analysis would be to consider the functionality of all of the electrical components as elements of the system, and everything else as part of the surroundings. This type of problem would be purely an initial value problem, given by a system of ordinary differential equations (ODEs). One could also consider the electrical and magnetic characteristics of the components as a function of time, e.g., as they increased in temperature due to current flow and heating of nearby components. A more complex type of analysis would be to consider the thermo-physical properties and the physical geometric characteristics of each of the components as part of the system. For this type of analysis, the surroundings would be the air around each of the components, the physical connections of each of the electrical components to the various circuit boards, and the radiation heat transfer with other electrical components and the surroundings. This type of system would be represented mathematically as an initial-boundary value problem given by a system of partial differential equations (PDEs). An additional factor to consider in these examples is human intervention affecting the system, e.g., a human switched on

the TV either by the physical switch on the television or by a remote control unit. A related factor that could be considered in the system is the mistreatment of the remote control unit by a child, such as rapid on/off switching of the unit. For these systems, the human would be considered as part of the surroundings.

3.1.2 Environments and scenarios

Scientific computing is often used to address the performance, safety or reliability of a system that is exposed to a wide variety of environments. We use the following definition.

Environment: the external condition or situation in which the system can be exposed to; specifically: normal, abnormal, or hostile conditions.

The three classes of environment (normal, abnormal, and hostile) were first formally defined as part of analysis of the safety, performance, and reliability of nuclear weapons in the US (AEC, 1966).

The *normal* system environment refers to either one of the following two conditions: (a) the operating environment in which the system is typically expected to operate or function and achieve its performance goals, or (b) an expected storage, shipping, or at-the-ready condition of the system. A normal operating environment for a system depends *entirely* on what should be the expected operating conditions of the system. For example, what may be considered a normal operating environment for one system may be considered an abnormal environment for another system. Examples of what could be considered a normal operating condition of some engineering systems are high temperature, pressure, or humidity; chemical or corrosive environments; vacuum conditions; a system covered or infiltrated with ice, snow, or sand; and a high intensity electromagnetic or radiation environment. Examples of typical storage, shipping, or at-the-ready conditions include a spacecraft either in ground storage before launch or in orbit in a non operational storage mode; a gas turbine engine being shipped from the manufacture or refurbishment facility to the user of the system; and safety or emergency systems for a nuclear power reactor that are at-the-ready. When a system is analyzed in a normal operating environment, the most common characteristics of interest are its performance and reliability. For systems in storage or shipping, the most common characteristics of interest are possible degradation of the system due to the environment, and safety of the system. For systems in at-the-ready environments, the most common characteristics of interest are such things as the response time of the system to full capability, and the degradation of the system performance or reliability as a function of time at-the-ready.

An *abnormal* environment of a system refers to either: (a) some type of accident or damaged-state environment, or (b) a very unusual condition that could put the system in jeopardy or cause it to be unsafe, even if the system is not operational. Examples of accident or damaged-state environments are: loss of primary coolant accident in a nuclear power plant, loss of electrical or hydraulic power during flight of an aircraft, exposure of the system to an accidental fire or explosive environment, flight control of a two-engine aircraft

during one engine out conditions, and structural integrity of a hypersonic flight vehicle with damage to the thermal protection system. Some examples of systems in very unusual environments are: exposure of a nuclear power plant to an earthquake; lightning strike on a system during operation, shipping, or storage; operation of the system at temperatures or pressures outside of its normal operating conditions; and over-riding the safety control systems during a safety check or proof-testing of a system. When a system is analyzed in an abnormal environment, the most common characteristic of interest is the safety of the system.

A *hostile* environment is one in which the system is under any type of attack, in the sense that the intent of the attacker is to do harm to the system, defeat or disable the system, or render it unsafe. The hostile environment can expose the system to attack from either inside the system or by way of the surroundings of the system. Types of attack can be physical damage or destruction of the system, modifying or taking over computer control of the operation of the system, altering the security or safety of the system, or electromagnetic attack over any portion of the electromagnetic spectrum. Military systems have always been evaluated with respect to performance in hostile environments. Before the terrorist attacks on the US in September 2001, very few privately owned facilities or public works were analyzed with respect to effects of hostile environments. Some examples of hostile environments for military systems are: battle damage due to small-arms fire on the system, exposure of electronic equipment to high-power microwaves or millimeter waves, attack of the computer control system either by an insider or through a connection to the internet, and attack of a ground vehicle by an improvised explosive device. When military systems are analyzed for a hostile environment, the most common characteristics of interest are system performance and safety. When civilian facilities and public works are analyzed for a hostile environment, system safety is the most common concern.

Given any environment that the system could be exposed to, one can also consider various scenarios that could occur, given the context of the environment being considered. We use the following definition.

Scenario: a possible event, or event sequence, to which a system in a given environment could be exposed.

Given this definition, scenarios are typically identified at the conceptual modeling phase. This phase was discussed in Sections 2.1.4 and 2.2.3, and will be discussed in more detail in Section 3.4.1 below. It should be noted that a scenario does *not* mean a particular realization of the response of a system. A scenario usually refers to an ensemble of possible system responses, all resulting from a common situation or sequence of situations that are identified with a specific environment. An example of an ensemble of system responses would be one or more cumulative distribution functions characterizing the nondeterministic response of the system. A scenario can be specified as a particular event sequence, or an entire event tree or fault tree.

Figure 3.1 depicts a given environment of interest and M scenarios that might be considered for analysis of a system. One example would be the specification of all of the

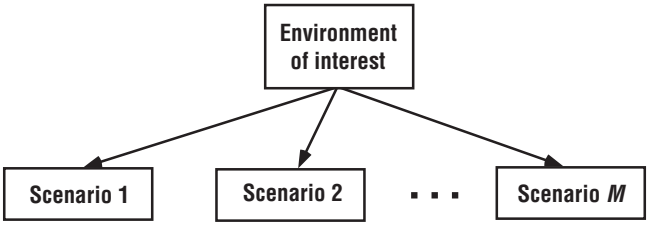

Figure 3.1 Environment-scenario tree.

extreme, or corner, conditions within the normal operating environment of the system; each identified as a scenario. For abnormal and hostile environments, it is especially important to identify multiple environments of interest within each category because there can be such a wide range of situations within each of the abnormal and hostile environments. As a result, there can be multiple environment-scenario trees like Figure 3.1 for a system in an abnormal environment. Identifying a number of scenarios for a given environment does not necessarily mean that each scenario will be analyzed. The environment-scenario tree only tries to identify possible conditions of the system that could be analyzed, for example, depending on the possible consequences or risk associated with each scenario. Examples of scenarios that could be considered for various environments of systems are (a) for a normal operating environment of a gas turbine engine on a transport aircraft in flight, consider the effect on engine performance of the scenarios of flight through rain, snow, freezing rain, and ice pellets (sleet), (b) for an accident environment of a hybrid automobile powered by both an internal combustion engine and a large battery unit, consider the scenarios of fire, explosion, and hazardous chemicals to the occupants of the automobile, bystanders or others involved in the accident, and emergency rescue personnel attending to the accident, and (c) for a hostile environment, consider the dispersion and transport of chemical or biological agents due to atmospheric winds, rain, storm drains, surface water, municipal water systems, surface vehicles, and people.

3.2 Fundamentals of models and simulations

3.2.1 Goals of scientific computing

The reasons for individuals, organizations, or governmental bodies to undertake activities in scientific computing are wide-ranging and diverse. However, these reasons can be grouped into the generation of new information or knowledge about systems or processes. This new information may then be used to influence the system or process analyzed, to design new and more capable systems, help avoid the detrimental and catastrophic effects of a possible situation, or it may be used to influence an individual's, organization's, or society's view of a system or process. The following is a general categorization of the motivations for scientific computing in engineering and the physical sciences (Bossel, 1994).

1 *Scientific knowledge*: scientific knowledge means knowledge generated solely for the improved understanding of the Universe and humankind's place in it. Probably the clearest example of the use of scientific computing for generation of scientific knowledge is in astrophysics. The knowledge generated in astrophysics improves human understanding of the Universe and their place in it, without any aspect of influence on the system or use of the knowledge toward other practical or earthly applications.
2 *Technological knowledge*: technological knowledge means the generation of knowledge used in some way for the creation of applied knowledge or the creation of new physical systems or processes. Technological knowledge is probably the most common type of knowledge generated from scientific computing in engineering and the applied physical sciences. In engineering simulations, the majority of this knowledge is used to design new engineering systems, improve the efficiency of existing systems, or assess the impact of existing or proposed systems. For systems not yet in existence, key drivers in scientific computing are issues such as: (a) creation and design of more capable systems than are presently on the market, particularly if they are a competitor's product; (b) creation of new materials and manufacturing processes to reduce costs and time to market; and (c) prediction of the potential environmental impact of new systems or manufacturing processes, not only at the present time, but in centuries to come. For the generation of new technological knowledge for systems already in existence, examples are (a) improvement of the performance, safety, or reliability of existing systems; (b) optimization of chemical processes for improved production output; (c) improvements in fuel consumption mileage of transportation vehicles; and (d) safety improvements to existing and future nuclear power plants, particularly improvements to address new threats.

Scientific computing is taking on increased public scrutiny, particularly for risk assessment of high consequence systems. The catastrophic failure of nuclear reactor number four at the Chernobyl nuclear power plant in Ukraine in 1986 riveted worldwide attention on the dangers of nuclear power as does the impact of environmental disasters beyond national borders. Although scientific computing primarily deals with technological issues, these issues commonly become convolved with public perceptions of risk, national responsibilities for the impact of technologies on the global climate, and organizational responsibilities for product and environmental liability. As is clear from these examples, the importance of scientific computing for the generation of technological knowledge is greatly expanding. Correspondingly, there is a compelling need to construct models that are technically sound, where the assumptions and uncertainties in the models are clearly revealed, and the modeling results are comprehensible to a wide range of audiences. The technical difficulties in achieving these goals in scientific computing are daunting in themselves; however, within the inevitable human, social, cultural, and political context, achieving them becomes nearly impossible.

The most obvious alternative method to the generation of new scientific and technological knowledge is the actual execution of the physical process or event that is of interest, i.e., conduct of a physical experiment. There are advantages and disadvantages to the physical experiment route, just as there are to the scientific computing approach. Some factors that should be considered in choosing scientific computing as compared to a physical experiment are the following (Neelamkavil, 1987; Bossel, 1994).

1 The cost and/or time schedule required to conduct a physical experiment may be considerably more than with scientific computing. Clear examples of this from recent history are in the simulation of electrical circuit functionality and performance as compared to the physical construction of an electrical circuit. Modern electrical circuits are designed almost entirely by scientific computing because of the speed and minimal cost of simulation. Another example is the simulation of large-scale structures, such as buildings and bridges. In the distant past, large-scale structures were built by trial and error, whereas during modern times there has been heavy reliance on scientific computing.
2 Because of the time scales involved in some physical process, it may be completely unrealistic to consider a physical experiment. For example, in disposal of radioactive nuclear wastes the time scales of decay of the wastes are on the order of thousands of years. An example with international environmental and economic dimensions is the long-term impact of burning of fossil fuels on global climate change.
3 Physical experiments with the actual system could possibly lead to unacceptable hazards or risks; cause large-scale disruptions in society, the economy, or the environment; be physically or financially infeasible; or not be allowed by international treaty. For example, consider the modification or repair of some large-scale structure such as a high-rise office building. Scientific computing would obviously be used to determine how the building structure might be modified for useful life extension or improved tolerance to physical attack. Scientific computing could be used to optimize the improvements to the structure with essentially no risk to the physical structure. An example that demonstrates the infeasible nature of a physical experiment is the response of a nuclear power plant to an earthquake. It is essentially impossible to generate a full-scale earthquake of proper amplitude and wavelength of ground motion for an experiment.

It should also be pointed out that there are limitations and weaknesses to using scientific computing in the generation of technological knowledge (Neelamkavil, 1987; Bossel, 1994). These are given in the following.

1 The cost and/or time required for construction of a mathematical model of a physical process may be excessive or impossible at the present time. For example, consider the detailed mathematical modeling of bolted joints between structural members in a structural dynamics problem. Even if one restricts the problem to the same two materials in contact at the joint, the physics and material science issues that must be addressed in the mathematical modeling are extraordinary. Some of the detailed aspects that must be addressed for an accurate mathematical model are (a) elastic and plastic deformation and irreversible changes of the material near the joint due to the compression of the bolt, (b) motion of the joint in six degrees of freedom (three translational plus three rotational), and (c) friction and heating between the two materials bolted together when they microscopically move and deform with respect to one another. In addition to these aspects of mathematical modeling, the model must take into account the uncertainty in each of these due to assembly, manufacturing, surface finish, oxidation, or corrosion of the material interfaces as a function of age and the surroundings, and deformation history.
2 The cost and/or time required to conduct the computational analysis may be excessive or economically unproductive. For example, consider the simulation of turbulent flow in fluid dynamics using as a model the time-dependent Navier–Stokes equations. If one attempts to solve these equations computationally, using what is referred to as direct numerical simulation, instead of using time-averaged models or turbulence models, one must have exceptionally powerful computer resources

available. Except for research interest into fluid dynamic turbulence, the costs of these types of simulation are prohibitive for high Reynolds number flows. Another example that may demonstrate inadequate schedule responsiveness of computational analyses is the time required to construct three-dimensional meshes for complex, multi-component assemblies.

3 Quantitative assessment of model accuracy may be difficult or impossible to attain because experimental measurements may be difficult or impossible to obtain, the cost may be prohibitively expensive, or the experiments may not be practical or allowed. Some examples are (a) obtaining detailed experimental measurements during hypervelocity impact of a particle on a spacecraft structure; (b) obtaining certain experimental data for the physiological response of humans to toxic chemicals; (c) conducting an experiment on the explosive failure of a full-scale reactor containment building; and (d) obtaining sufficient input and output data for the response of the global environment to a large-scale atmospheric event, such as a volcanic eruption or the impact of an sizeable asteroid.

3.2.2 Models and simulations

Diverse types of model are used in a wide range of disciplines. Neelamkavil (1987) gives a general and well-founded definition that covers many different types of models for physical systems.

Model: a representation of a physical system or process intended to enhance our ability to understand, predict, or control its behavior.

There are several variants of the definition of simulation in the literature, but we will use the following concise definition.

Simulation: the exercise or use of a model to produce a result.

Simulation of the behavior of systems can be achieved by two types of mathematical model (Bossel, 1994). The first type is referred to as an empirical or phenomenological model of the system. This type of mathematical model of the system is based on observations of how the system responds under different input, or stimulation, conditions. The representation commonly does not make an attempt to describe any of the detailed processes involved inside the system or determine why the system responds in the way it does. The system is considered to be a *black box* and the only issue is the global relationship between the inputs and outputs of the system. One relates the observed behavior of the system to the perceived influences on the system using some type of mathematical representation, such as statistical correlation or regression fit methods. An example of this type of model is the dynamic response of a structure to a bolted or riveted joint. If the system is only considered to be the joint, then an empirical model can be constructed of how the structure responds to the joint. The model would represent, say, the structural stiffness and torsional damping of the joint. The information to construct this type of model is usually obtained from experimental measurements of the dynamic response of the structure. Parameter identification methods are applied to the structural response to determine the input–output relationship over a range of conditions.

The second type of mathematical model is the physical law or explanatory model. For this type of model a great deal of information must be known about the actual processes occurring inside the system. In the physical sciences and engineering, this type of model is the one of principal interest. Past observations of the behavior of the system are primarily of value in determining what physical processes and laws must be considered and what can be ignored in the system, and secondarily, how physical modeling parameters can be adjusted to best represent the response of the system. Examples of this type of physical law model are Newton's second law, Fourier's law of heat conduction, the Navier–Stokes equations, Maxwell's equations, and Boltzman's equation. Many physical law models were devised more than a hundred years ago and they form the foundations of the modern analysis of physical systems. With the creation of extremely powerful computers, the technological impact of these fundamental laws is unprecedented in history.

The general strategy of model building that should be followed is to include *only* the elements and processes that are important to achieve the goals of the computational analysis. Often, preliminary simulations show that changes and improvements in the modeling approach are needed to achieve the goals of the analysis. Albert Einstein's classic advice in this matter was: "Make the model as simple as possible, but no simpler." This strategy of using the simplest possible theory to explain reality is also referred to as Occam's Razor. The predictive power of a model depends on its ability to correctly identify the dominant controlling factors and their influences, *not* upon its completeness. A model of limited, but known, applicability is generally more useful from a system design or decision-making perspective than a more complete model that requires more detailed information and computing resources.

Many fields of engineering and the physical sciences have largely ignored the argument for using moderate complexity models as opposed to higher complexity models. The argument is made that with continually increasing computing power, analysts should develop increasingly complex models, so as to include all the possible processes, effects, and interactions. One way in which the level of complexity of the model is constrained is to include the most physics complexity, while still seeking to obtain a solution within the time and computer resources available. There is some credence given to this constraint, but it is seldom realized in practice. Increasing physics modeling complexity is *always* at the expense of simulation result timeliness, nondeterministic simulations, investigation of possible environments and scenarios, sensitivity analyses, and investigation of the effect of alternative modeling approaches. Stated differently, most fields of engineering and the physical sciences are still entrenched in deterministic simulations, so they do not factor in the need for many simulations in order to conduct uncertainty quantification and sensitivity analysis. In many fields of engineering and the physical sciences, the vast increases in computer power have been consumed by increased modeling complexity, often leading to only limited improvement in risk-informed decision-making.

The construction of mathematical models of systems *always* involves simplifications of the physical reality. Modeling simplifications can usually be thought of as one of three types: omission, aggregation, and substitution (Pegden *et al.*, 1990). Omission simply means that

certain physical characteristics, processes, features, or events of a system are ignored. For example, suppose one is interested in modeling the heat flux from the surface of a heated solid. If convective heat transfer is the dominant heat transfer mechanism, then the radiation heat transfer might be neglected in the model of the system. Simplification of a model by aggregation means that a characteristic is not ignored, but is combined or lumped together into a roughly equivalent characteristic. For example, in fluid dynamics if the mean free path of the atoms or molecules is much less than the characteristic length of the geometric features in the flow field, then a continuum fluid field is normally assumed. Simplification of a model by substitution means that some complex characteristics are replaced by a simpler characteristic. For example, in modeling of hydrocarbon fuel combustion the number of intermediate gas species that exist is typically in the hundreds. Depending on the needs of the simulation, a substitution combustion model for the number of gas species may only number 10 to 20.

The mathematical models of interest here are primarily given by PDEs or integro-differential equations. These equations result in initial value, boundary value, or initial-boundary value problems. The PDEs can be elliptic, parabolic, or hyperbolic in character, have two or more independent variables, and have one or more dependent variables. The PDEs describe the relationship between the dependent variables within the system, given the effect of the surroundings on the system. Information provided about the surroundings, by way of boundary conditions and system excitation conditions, is *independent* information needed for the solution of the PDEs. The differential equations can be solved by a wide variety of numerical methods, such as finite element, finite difference, or finite volume methods.

In addition to the primary PDEs of interest, there are commonly submodels, or auxiliary models, that can be stated in a variety of mathematical forms: algebraic, transcendental, table-lookup, matrix, differential, or integral equations. Examples of submodels are PDEs for modeling fluid dynamic turbulence, integro-differential equations for material constitutive properties in shock physics, and integral equations for linear viscoelasticity models in solid mechanics. The submodels can also be stated, all or in part, in tabular form so that numerical interpolation functions are used to construct the required functional relationship.

Figure 3.2 depicts a system of interest and its surroundings. For most systems of interest considered here, the key types of information describing the system are geometry, initial conditions, and physical modeling parameters. For simple systems, the geometry can be specified by engineering drawings and information concerning how the system is assembled. For most engineering systems, however, all the minute details in the geometry are specified in a computer aided design (CAD) software package. In addition, computer aided manufacturing (CAM) software may be used for more detail on the actual manufacturing and assembly process, such as deburring, riveting and bolting procedures, and electrical cable bending and pulling procedures. For systems that are modeled as an initial value problem, initial conditions (ICs) provide required information concerning (a) the initial state of all the dependent variables in the PDEs and (b) the initial state of all other physical

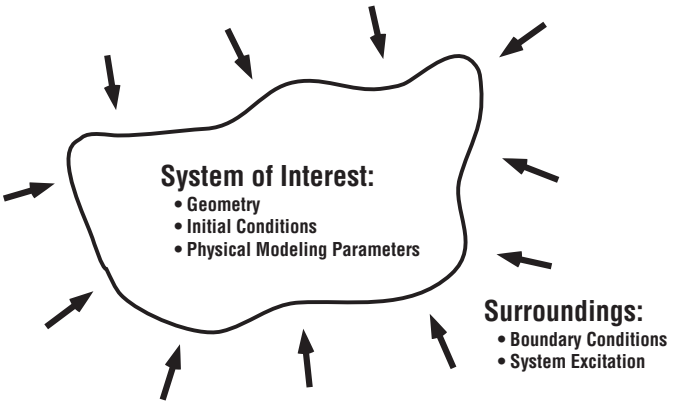

Figure 3.2 Types of information in the system and the surroundings.

modeling parameters, including geometric parameters, that could be dependent on time. As a result, the IC data could be a function of the remaining independent variables in the PDEs. The final element of information characterizing the model of the system is the physical modeling parameters. Examples of physical modeling parameters are Young's modulus, mass density, electrical conductivity, thermal conductivity, parameters in constitutive equations, damping and stiffness of assembled joints in a structure, effective chemical reaction rate, and thermal contact resistance in heat transfer. Some parameters can describe global characteristics of the system and some can vary as a function of both the independent and dependent variables in the PDEs. As will be discussed in Chapter 13, Predictive capability, some parameters can be measured independent of the system being modeled and some must be inferred based on the particular model being used and observations of the response of the system.

Two types of information must be provided concerning the surroundings: boundary conditions (BCs) and system excitation. BCs provide separate information concerning the dependent variables of the PDEs along the boundary of the domain. BCs can be dependent on one or more of the independent variables of the PDEs. These independent variables are typically other spatial dimensions and time, if the problem is formulated as an initial-boundary value problem. For example, in a structural dynamics problem the loading on the structure by way of the BCs can be time dependent. Examples of different types of BCs are: Dirichlet, Neumann, Robin, mixed, periodic, and Cauchy. System excitation refers to how the surroundings affect the system, *other than* through the BCs. System excitation always results in a change in the form of the PDEs being solved. Sometimes system excitation is referred to as a change in the right hand side of the PDEs to represent the effect of the surroundings on the system. Common examples of system excitation are (a) a force field acting on the system, such as due to gravity or an electric or magnetic field, and (b) energy deposition distributed through the system, such as by electrical heating or chemical reactions.

3.2.3 Importance of nondeterministic simulations

In many science and engineering communities, particularly for research activities, predictions are viewed strictly as deterministic predictions. Commonly, the purpose of these investigations is to discover new physical phenomena or new characteristics of systems or processes. Nondeterministic characteristics of the phenomena or system are of secondary importance. Sometimes in these analyses, it is explicitly stated or implied that since the investigator is only interested in *nominal* values of the output quantities, he/she can attempt to compute these quantities using the nominal values for the input quantities. The nominal values of the uncertain inputs may be specified as the mean value of each of the probability distributions for the inputs. It is *rarely true*, however, that the statistical mean of the output can be determined by performing a calculation for a single set of inputs chosen to be the statistical mean of each of the uncertain inputs. Stated another way, the mean value of the output cannot be computed by performing a simulation using the mean value of all input parameters, except when the mapping of inputs to outputs is linear in the parameters. Linearity in the parameters essentially never occurs when the mapping of inputs to outputs is given by a differential equation, *even a linear* differential equation. How much in error this approximation is depends on the characteristics of the system, particularly the nonlinearity in the input to output mapping of uncertain quantities.

In most engineering applications, as well as applications in the physical sciences, deterministic simulations are unacceptable approximations. The effort devoted to estimating nondeterministic effects can vary widely depending on the goals of the computational analysis, the expected performance, safety, and reliability of the system, and the possible consequences of system failure or misuse. An example of a computational analysis that might require little effort for nondeterministic effects is a system that is relatively simple, the use of the system by the customer is well understood, and the risk of injury or misuse of the system is minimal. For example, the computational analysis may only consider a few important design parameters as aleatory uncertainties, i.e., precisely known random variables, and not consider any epistemic uncertainties, i.e., uncertainties due to lack of knowledge. The SRQs of interest computed by the computational analysis would then be expressed as probability density functions (PDFs) or cumulative distribution functions (CDFs).

Simulation of complex engineering systems, expensive commercial systems, and high-consequence systems must include the nondeterministic features of the system and the surroundings, in addition to the analysis of normal, abnormal, and hostile environments. Several fields that regularly employ nondeterministic simulations are nuclear reactor safety (Hora and Iman, 1989; Morgan and Henrion, 1990; NRC, 1990; Hauptmanns and Werner, 1991; Breeding *et al.*, 1992; Helton, 1994), underground contamination of toxic and radioactive waste materials (LeGore, 1990; Helton, 1993; Helton *et al.*, 1999; Stockman *et al.*, 2000), civil and structural engineering (Ayyub, 1994; Ayyub, 1998; Ben-Haim, 1999; Melchers, 1999; Haldar and Mahadevan, 2000a, Moller and Beer, 2004; Ross, 2004; Fellin *et al.*, 2005; Tung and Yen, 2005; Ang and Tang, 2007; Choi *et al.*, 2007; Vinnem, 2007),

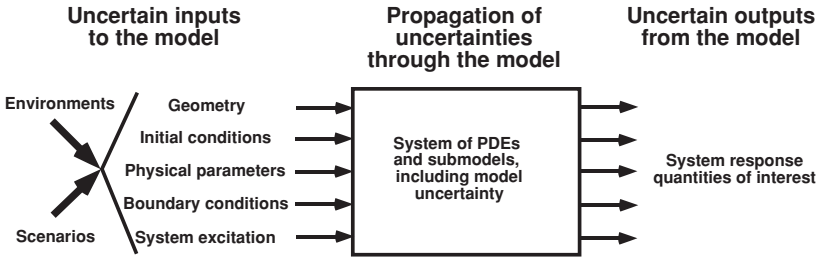

Figure 3.3 Propagation of input uncertainties to obtain output uncertainties.

environmental impact assessment (Beck, 1987; Bogen and Spear, 1987; Frank, 1999; Suter, 2007), and broader fields of risk assessment and reliability engineering (Kumamoto and Henley, 1996; Cullen and Frey, 1999; Melchers, 1999; Modarres *et al.*, 1999; Bedford and Cooke, 2001; Andrews and Moss, 2002; Bardossy and Fodor, 2004; Aven, 2005; Nikolaidis *et al.*, 2005; Ayyub and Klir, 2006; Singpurwalla, 2006; Singh *et al.*, 2007; Vose, 2008; Haimes, 2009). The emphasis in most of these fields has been directed toward representing and propagating parameter uncertainties through the mathematical model to obtain uncertain system responses. The majority of this work has used traditional probabilistic methods or Bayesian methods, where no real distinction is made between aleatory and epistemic uncertainties.

3.2.4 Analysis of nondeterministic systems

The key issue in nondeterministic simulations is that a single solution to the mathematical model is no longer sufficient. A set, or ensemble, of calculations must be performed to map the uncertain input space to the uncertain output space. Sometimes, this is referred to as *ensemble simulations* instead of nondeterministic simulations. Figure 3.3 depicts the propagation of input uncertainties through the model to obtain output uncertainties. The number of individual calculations needed to accurately accomplish the mapping depends on four key factors: (a) the nonlinearity of the PDEs; (b) the nonlinearity of the mapping in terms of the uncertain quantities; (c) the nature of the uncertainties, i.e., whether they are aleatory or epistemic uncertainties; and (d) the numerical methods used to compute the mapping. The number of mapping evaluations, i.e., individual numerical solutions of the mathematical model, can range from several to hundreds of thousands. Obviously, this latter value is shocking to those accustomed to a single solution to a set of PDEs.

With the descriptions given above, we can write the formal model structure that maps to the SRQs of interest as

$$\mathbb{M}(E, S; D, G, IC, MP, BC, SE) \rightarrow SRQ. \tag{3.1}$$

$\mathbb{M}$ is the specification of the mathematical model, E is the environment of the system, S is the scenario of the system, D is the differential or integro-differential equation describing the system, G is the geometry of the system, IC are the initial conditions of the system, MP

are the model parameters of the system, *BC* are the boundary conditions imposed by the surroundings, and *SE* is the system excitation imposed by the surroundings. *D*, *G*, *IC*, *MP*, *BC*, and *SE* are all conditional on the specified environment *E* and the scenario of interest *S*. If *D*, *G*, *IC*, *MP*, *BC*, and *SE* are all completely specified, either deterministically or nondeterministically, then the mathematical model $\mathbb{M}$, as given by Eq. (3.1), is referred to as the strong definition of a model (Leijnse and Hassanizadeh, 1994). The weak definition of a model, according to Leijnse and Hassanizadeh (1994), is one where only *D* is specified, given *E* and *S*. The weak definition of a model could then be written

$$\mathbb{M}(E, S; D) \rightarrow SRQ. \tag{3.2}$$

For a model given by Eq. (3.2), the SRQs can not be numerically computed because of the lack of specificity in the model. In addition, the weak definition of a model cannot be validated.

Many techniques exist for propagating input uncertainties through the mathematical model to obtain uncertainties in the SRQs. For a detailed discussion of many methods, see the following texts: (Kumamoto and Henley, 1996; Cullen and Frey, 1999; Melchers, 1999; Modarres *et al.*, 1999; Haldar and Mahadevan, 2000a; Bedford and Cooke, 2001; Ross, 2004; Aven, 2005; Ayyub and Klir, 2006; Singpurwalla, 2006; Ang and Tang, 2007; Kumamoto, 2007; Suter, 2007; Vose, 2008; Haimes, 2009). Sampling techniques are the most common approach because of a number of advantages: (a) they can be applied to essentially any type of mathematical model, regardless of the model's complexity or nonlinearity; (b) they can be applied to both aleatory and epistemic uncertainties, regardless of the magnitude of the uncertainties; and (c) they are not intrusive to the numerical solution of the mathematical model, i.e., sampling is done outside of numerical solution to the PDEs. Their key disadvantage is that they are computationally expensive because the number of mapping evaluations can be very large in order to obtain the statistics of interest for the SRQs. Sampling essentially solves the nondeterministic PDEs by segmenting the solution into multiple deterministic problems. If the nondeterministic PDEs are linear, it is well accepted that this segmented approach converges to the nondeterministic solution as the number of samples becomes large. If the PDEs are nonlinear, however, the correctness of this approach has not been proven, in general. See Taylor and Karlin (1998); Kloeden and Platen (2000); Serrano (2001); and Oksendal (2003) for detailed discussions of the numerical solution of stochastic PDEs.

The particular approach used in this text for nondeterministic simulations is *probability bounds analysis* (PBA) (Ferson, 1996; Ferson and Ginzburg, 1996; Ferson, 2002; Ferson *et al.*, 2003; Ferson and Hajagos, 2004; Ferson *et al.*, 2004; Kriegler and Held, 2005; Aughenbaugh and Paredis, 2006; Baudrit and Dubois, 2006). PBA is closely related to two more well-known approaches: (a) two-dimensional Monte Carlo sampling, also called nested Monte Carlo, and second order Monte Carlo (Bogen and Spear, 1987; Helton, 1994; Hoffman and Hammonds, 1994; Helton, 1997; Cullen and Frey, 1999; NASA, 2002; Kriegler and Held, 2005; Suter, 2007; Vose, 2008; NRC, 2009), and (b) evidence theory, also called Dempster–Shafer theory (Krause and Clark, 1993; Almond, 1995; Kohlas and

Monney, 1995; Klir and Wierman, 1998; Fetz *et al.*, 2000; Helton *et al.*, 2004, 2005; Oberkampf and Helton, 2005; Bae *et al.*, 2006). PBA is an approach that can be concisely described as a combination of interval analysis and traditional probability theory. PBA stresses the following perspectives: (a) mathematically characterize input uncertainty as either aleatory or epistemic; (b) characterize the model uncertainty as epistemic uncertainty; (c) map all input uncertainties through the model, typically using sampling techniques, while keeping each type of uncertainty separate; and (d) portray the uncertainty in the SRQs as a probability box, (p-box). A p-box is special type of cumulative distribution function that represents the set of all possible CDFs that fall within the prescribed bounds. As a result, probabilities can be interval-valued quantities as opposed to a single probability. A p-box expresses both epistemic and aleatory uncertainty in a way that does not confound the two. Two-dimensional Monte Carlo commonly retains some of the probabilistic nature in the sampling of the epistemic uncertainties, whereas PBA maintains a strict separation between aleatory and epistemic.

PBA typically uses standard sampling techniques, such as Monte Carlo and Latin Hypercube sampling (Cullen and Frey, 1999; Ross, 2006; Ang and Tang, 2007; Dimov, 2008; Rubinstein and Kroese, 2008). In the sampling process, however, the samples taken from the aleatory and epistemic input uncertainties are treated differently. The samples taken from aleatory uncertainties are treated as probabilistic realizations, i.e., a probability of occurrence is associated with each sample. The samples taken from the epistemic uncertainties are treated as *possible* realizations and, as a result, each sample is given a probability of *unity*. The reason epistemic uncertainties are treated this way is that they are samples drawn from interval-valued quantities. That is, all that can be claimed is that *all values* drawn from within the interval are possible, because the likelihood of one sample compared to another is unknown. This is a weaker statement of knowledge than claiming that all values within the interval are *equally* possible, i.e., a uniform PDF over the interval. As a result, the structure of a p-box for the SRQ is such that over the range where epistemic uncertainty exists, one will have an interval-valued range of probabilities. That is, over the range of epistemic uncertainty, the *most precise statement that can be made* about the SRQ is that the probability can be no larger than the computed value and no smaller than the computed value, given the epistemic uncertainty in the input. A distribution of this type is sometimes referred to as an *imprecise probability distribution*.

A simple example using PBA will be given in the next section. A more detailed discussion of PBA will be given in Chapter 13.

Types of uncertain quantities that can occur in a mathematical model are: parameters, event state specifications, independent variables, dependent variables, geometry, ICs, BCs, system excitation, and SRQs. Most parameters are viewed as continuous parameters, although it can be a simplification for mathematical convenience. For example, the number of mesh points in a numerical solution is considered continuous, even though it can only take on integer values. Uncertain parameters are usually specific values drawn from a population of a finite sample space. For example, consider a simple electrical circuit with an inductor, capacitor, and resistor. If the value of the resistance is considered to be uncertain due to

manufacturing variability, then the resistance is usually treated as a continuous random variable. Some parameters are discrete, or quantized, values and they must be considered as such. For example, a switch on a control system may only have two settings (on or off), and in a safety analysis of a system with an access door, the door may be considered as only fully open or fully closed.

Event state specifications have some similarities to parameters that can take on discrete values, but event state specifications are primarily directed toward analyzing or finding specific system states that can severely impact the safety or reliability of a complex system (Kumamoto and Henley, 1996; Modarres *et al.*, 1999; Haimes, 2009). For example, fault-tree analyses are a deductive process to try and determine, given an undesirable event called the top event, all the system or component faults that could possibly happen to cause the top event. A similar technique is an event-tree analysis. If the successful operation of a system depends heavily on the chronological operation of units or subsystems, or the action of individuals, then possible events are considered to try and determine if undesirable events or states could occur.

SRQs can simply be the dependent variables in the PDEs in the mathematical model. They can also be more complex quantities such as derivatives of dependent variables, functionals of dependent variables, or complex mathematical relations between dependent variables and their frequency of occurrence. For example, an SRQ that is a functional of the dependent variables in the analysis of the plastic deformation of a structure would be the total strain energy absorbed by the structure as a function of time. If *any* input quantity to the model is nondeterministic, the SRQs are, in general, also nondeterministic.

When an uncertainty analysis is complete, it is commonly followed by a sensitivity analysis. A sensitivity analysis uses the nondeterministic results computed for the uncertainty analysis, but it attempts to answer somewhat different questions related to the system of interest. Sometimes sensitivity analyses are referred to as *what-if* or perturbation analyses of the system. The computational expense added by a sensitivity analysis is typically minimal compared to the uncertainty analysis because additional function evaluations are usually not needed. Two of the most common questions raised in a sensitivity analysis are the following.

First, what is the rate of change of SRQs of interest with respect to the uncertain input quantities? Here the focus is on local derivatives of SRQs with respect to uncertain inputs, all other input quantities remaining fixed at a specified value. This type of analysis is usually referred to as a *local sensitivity analysis*. When these derivatives are computed for a variety of input quantities, one can then rank the magnitude of the sensitivity of the output quantity with regard to the various input quantities. Note that these derivatives, and the resulting ranking of input quantities, can strongly depend on the values chosen for the input quantities. That is, the sensitivity derivatives typically vary widely over the range of uncertainty of the system design quantities and the range of operating conditions of the system.

Second, what uncertain inputs have the largest effect on SRQs of interest? Here the focus is *not* on the uncertainty of the SRQs, but on which input uncertainties have the largest

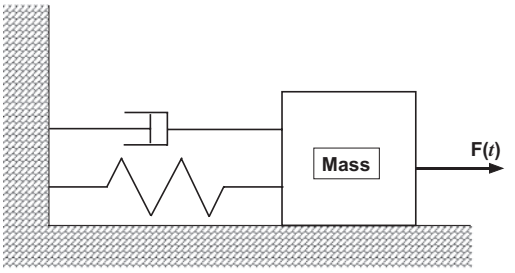

Figure 3.4 Example of a mass–spring–damper system.

global effect on SRQs of interest. This type of analysis is usually referred to as a *global sensitivity analysis*. Here *global* refers to a specific environmental condition and a specific scenario. For example, suppose there are ten uncertain parameters in a design study of the performance of some system under a normal environment and a given scenario. A global sensitivity analysis could rank order the uncertain design parameters according to which parameters produce the largest effect on a particular system performance measure, given the range of uncertainty of each design parameter. The answer to this question is of great value not only in optimization of design parameters of the system, but also for possibly restricting the operational parameters of the system that are imposed by the surroundings. For a detailed discussion of sensitivity analyses, see Kleijnen (1998); Helton (1999); Cacuci (2003); Saltelli *et al.* (2004); Helton *et al.* (2006); Saltelli *et al.* (2008); and Storlie and Helton (2008).

3.2.5 Example problem: mechanical oscillation

Consider the simulation of the oscillation of a mass–spring–damper system that is acted upon by a time dependent excitation force (Figure 3.4). The ordinary differential equation describing the oscillation of the system is given by

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + c\frac{\mathrm{d}x}{\mathrm{d}t} + kx = F(t),$$
$$\text{Initial conditions: } x(0) = x_0 \quad \text{and} \quad \left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)_{t=0} = \dot{x}_0, \tag{3.3}$$

where $x(t)$ is the displacement of the mass as a function of time, m is the mass of the system, c is the damping coefficient, k is the spring constant, and $F(t)$ is the external forcing function.

Consider two nondeterministic variants of this system.

3.2.5.1 Aleatory uncertainty

For the first system, assume that all features of the system, save one, are exactly known, i.e., deterministic. The damping coefficient, c, the spring constant, k, the initial state of the

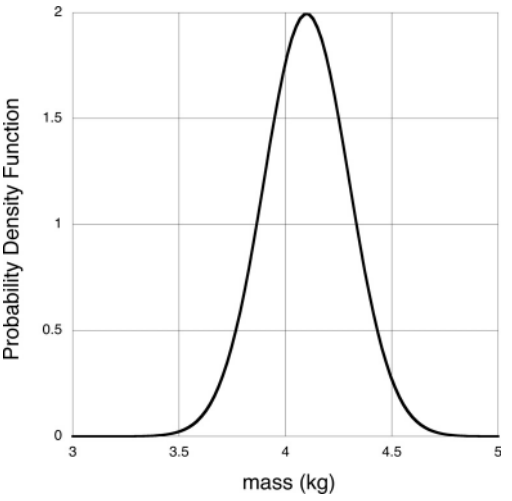

Figure 3.5 Probability density function of the system mass population.

system, x_0 and $\dot{x}_0$, and the forcing function, $F(t)$, are precisely known. These values are given as

$$c = 1 \quad \text{N/m/s}, \quad k = 100 \quad \text{N/m}, \quad x_0 = 10 \quad \text{m}, \quad \dot{x}_0 = 0 \quad \text{m/s}, \tag{3.4}$$

and

$$F(t) = \begin{cases} 0 & \text{for } 0 \leq t < 2 \text{ s} \\ 1000\, N & \text{for } 2 \text{ s} \leq t \leq 2.5 \text{ s} \\ 0 & \text{for } 2.5 \text{ s} < t. \end{cases} \tag{3.5}$$

As can be seen from Eq. (3.4) and Eq. (3.5), the initial conditions are a displacement of 10 m and a velocity of zero. In addition, it is seen that the excitation function, $F(t)$, only comes into play during the time period of 2 to 2.5 s.

The mass of the system, m, is nondeterministic due to variability in its manufacturing. A large number of inspections have been made of the manufactured masses that are used in the system so that a precise probability density function (PDF) for the population can be generated. It was found that the PDF of the population could be accurately represented by a normal (Gaussian) distribution with a mean of 4.2 kg and a standard deviation of 0.2 kg, as shown in Figure 3.5.

Since Eq. (3.3) is linear, the solution to the mathematical model can be written analytically, i.e., in closed-form, or it can be solved numerically using a standard ODE solver. For our simulation, Eq. (3.3) was converted into two first-order ODEs and then solved numerically using MATLAB's Runge-Kutta 4(5) method, *ode45*. The numerical solution error for each time-step advancement was required to be less than 10^{-3} for the relative error, and less than 10^{-6} absolute error, for each dependent variable.

Since the nondeterministic nature of the system is purely aleatory, traditional sampling methods can be used to propagate the mass uncertainty into uncertainty of the SRQs of

interest. We used Monte Carlo sampling incorporated in MATLAB's normal distribution sampler *randn* in order to obtain samples for the mass. The mean, μ, was set to 4.2 kg, and the standard deviation, σ, was set to 0.2 kg. In nondeterministic simulations using sampling, a random number seed is required so that one can reproduce precisely the same sequence of random numbers. This technique is referred to as pseudo-random number generation. In the MATLAB program *randn*, the default seed of 0 was used, with the number of samples, n, of 10, 100, and 1000.

Various SRQs can be computed, for example, position, velocity, and acceleration, $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$, respectively. Figure 3.6, Figure 3.7, and Figure 3.8 show $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$, respectively, for time up to 10 s, and $n = 10$, 100, and 1000. The expected oscillatory motion is seen in each of the SRQs. The effect of the excitation function during the time period of 2 to 2.5 s cannot be seen in the displacement plot, it is just barely noticeable in the velocity plot, and is clearly visible in the acceleration plot. In each plot, every Monte Carlo sample that is computed is shown. As a result, it is difficult to see any of the individual numerical solutions in the plots for $n = 100$ and 1000.

Since the nondeterministic simulation of the SRQs is a distribution of results, as opposed to an individual deterministic result, it is appropriate to interpret the results in terms of statistical measures of the SRQs. Table 3.1 and Table 3.2 show the estimated mean and standard deviation of $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$, at $t = 1$ and 5 s, respectively, as a function of the number of samples computed, including 10 000 samples. The $\hat{}$ symbol indicates that the values for μ and σ are sample values for the mean and standard deviation as opposed to exact values of the population. As expected, both $\hat{\mu}$ and $\hat{\sigma}$ will change as a function of the number of samples computed because of relatively few samples. In the limit as the number of samples increases, $\hat{\mu} \to \mu$ and $\hat{\sigma} \to \sigma$. As expected in Monte Carlo sampling, there is relatively little change in μ and σ after 100 samples for most cases. The results in the tables are given to three significant figures for each set of samples.

Another traditional method of presenting the results of a nondeterministic system is to show a plot of the CDF of each of the SRQs. The CDF shows the fraction of the population that would have a value less than, or equal to, a particular value of the SRQ. A CDF shows the distributional information concerning a nondeterministic quantity, as opposed to some type of summary measure of the distribution, such as a mean or standard deviation. When a limited number of samples are computed, or measured in an experiment, the CDF is referred to as an empirical distribution function (EDF). The EDF shows the fraction of the *sampled* population that would have a value less than, or equal to, a particular value of the SRQ. Another traditional method of showing nondeterministic results is to show histograms of each of the SRQs. Although this can be helpful for certain situations, we do not generally use this method because it requires the analyst to pick a bin size for the histogram. Picking a bin size is actually an assumption that must be made in the analysis in order to show the statistical results. We are of the viewpoint that the fewer assumptions made in an analysis, particularly in a statistical analysis, the better.

Figure 3.9 and Figure 3.10 show the EDF of $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$, at $t = 1$ and 5 s, respectively, for each of the number of samples computed. The most notable feature in each

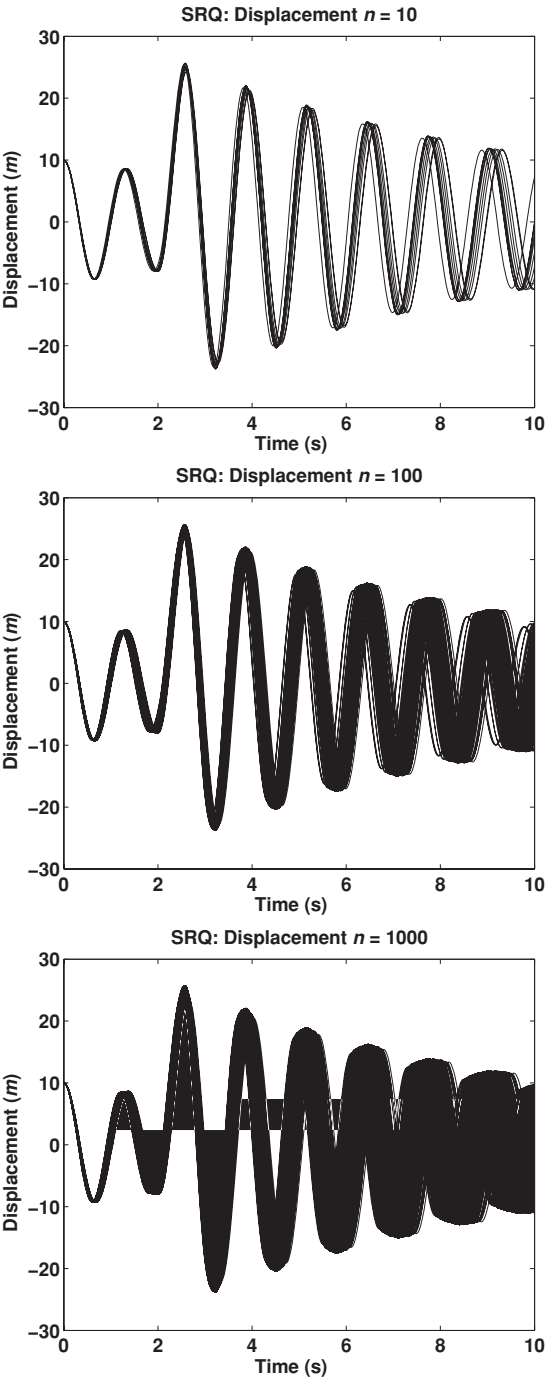

Figure 3.6 Mass displacement as a function of time for aleatory uncertainty.

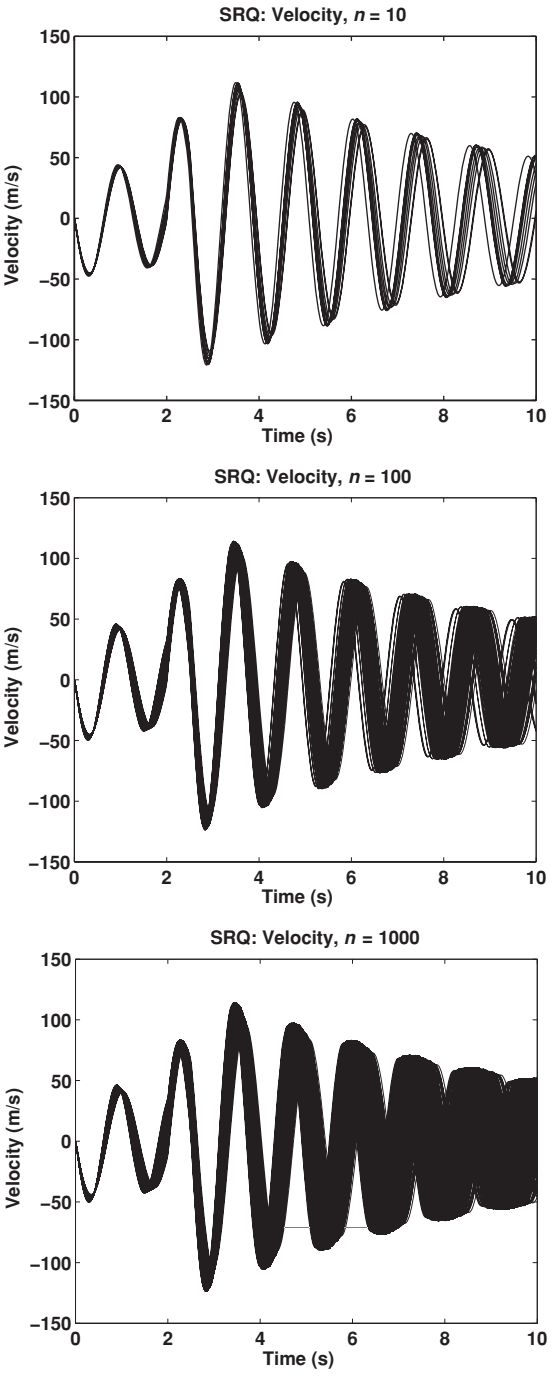

Figure 3.7 Mass velocity as a function of time for aleatory uncertainty.

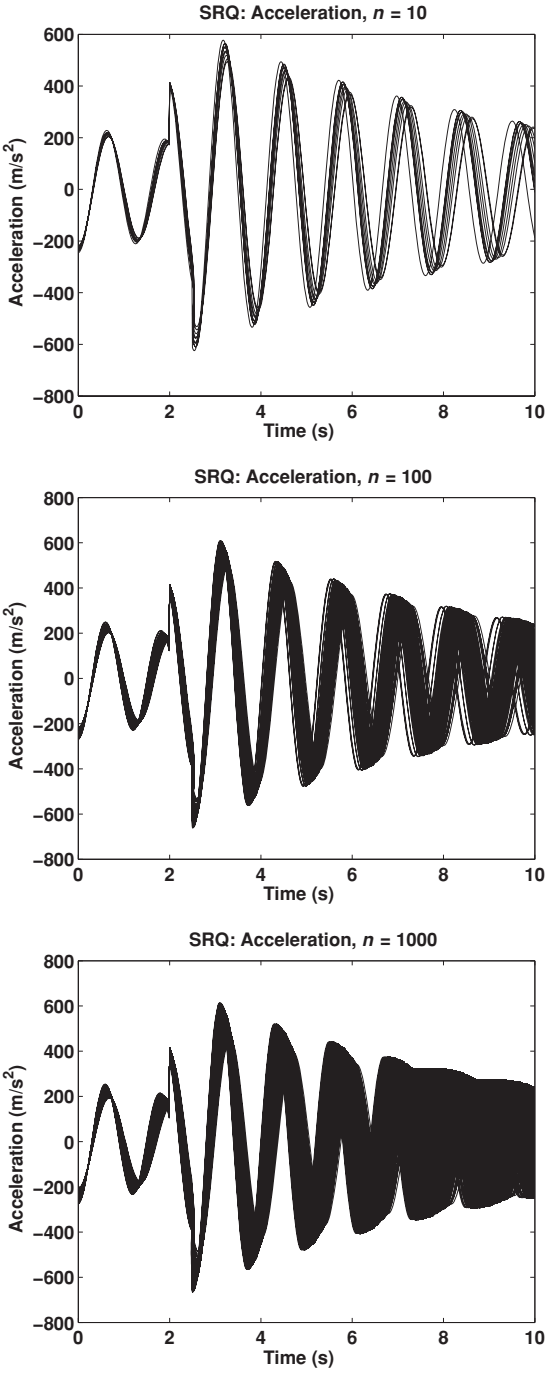

Figure 3.8 Mass acceleration as a function of time for aleatory uncertainty.

Table 3.1 *Statistics for displacement, velocity, and acceleration for $t = 1$ s.*

	$\hat{\mu}$	$\hat{\sigma}$
Displacement x		
Samples (n)		
10	0.65	0.76
100	1.20	0.96
1000	1.29	1.03
10 000	1.29	1.01
Velocity $\dot{x}$		
Samples (n)		
10	42.5	0.286
100	42.4	0.462
1000	42.4	0.458
10 000	42.4	0.478
Acceleration $\ddot{x}$		
Samples (n)		
10	−25.5	18.5
100	−39.7	25.3
1000	−42.1	26.8
10 000	−42.0	26.2

graph, particularly to those not familiar with EDFs, is the stair-step nature of the curve for $n = 10$ samples. The reason for this characteristic is that there are so few samples to characterize the true distribution. It is seen in every plot with 10 or 100 samples that the EDF (a) is very rough and gives the appearance that it may be discontinuous, (b) apparently contains a bias error because it is commonly shifted to the left or right of the high fidelity EDF for $n = 1000$, and (c) is deficient in showing any of the tails of the high fidelity distribution. The rough, or stair-step, nature of the plot results from the fact that there is a jump in probability at each of the observed samples. With only 10 samples, each sample *must* represent a probability jump of 0.1. None of the EDFs are discontinuous, but each one is a stair-step where the height of each step is $1/n$. With few samples, there is commonly a bias error in the computed distribution. This tendency of a bias error due to a low number of samples can also be seen in computing μ and σ of the distributions, see Table 3.1 and Table 3.2. Since Monte Carlo sampling is an unbiased estimator of the statistics, the bias error approaches zero as the number of samples becomes very large. The inaccuracy in the computed tails of the distributions is usually referred to as the inaccuracy of computing low probability events with a small number of samples. This is a well-known deficiency in Monte Carlo sampling and it is discussed in Chapter 13. For a more detailed discussion, see Cullen and Frey (1999); Ang and Tang (2007); Dimov (2008); and Vose (2008).

Table 3.2 *Statistics for displacement, velocity, and acceleration for $t = 5$ s.*

	$\hat{\mu}$	$\hat{\sigma}$
Displacement x		
Samples (n)		
10	10.4	4.33
100	12.7	4.10
1000	13.0	4.69
10 000	13.1	4.53
Velocity $\dot{x}$		
Samples (n)		
10	70.9	14.2
100	57.3	26.2
1000	54.0	26.9
10 000	54.4	26.5
Acceleration $\ddot{x}$		
Samples (n)		
10	−260	106
100	−320	105
1000	−329	118
10 000	−330	114

3.2.5.2 *Epistemic uncertainty*

All of the characteristics of this system are the same as the previous system, except for the nondeterministic character of the mass. For this case the system manufacturer concluded that the variability of the system response was unacceptably large due to the variability in the masses used. The primary concern was that, based on the characterization of the variability in the mass as a normal distribution, there could be masses with very low and very high values. Although there were very low probabilities associated with the very low and very high masses, these situations could cause the system to fail. Because the consequence of system failure was severe, project management became quite concerned with respect to legal liability.

As a result, the project manager found another supplier of masses for their system that claimed they could produce masses with a guaranteed bound on the variability of the masses they produced. The new supplier's procedure was to reject any masses produced during the production process that were either below or above a specification set by the customer. However, to cut costs, they did not weigh the production masses that pass their inspection process to determine what the variability is of their delivered product. Consequently, the new supplier could only guarantee that the uncertainty of the masses they delivered to the customer were within the specified interval. As a result, when the customer simulated their

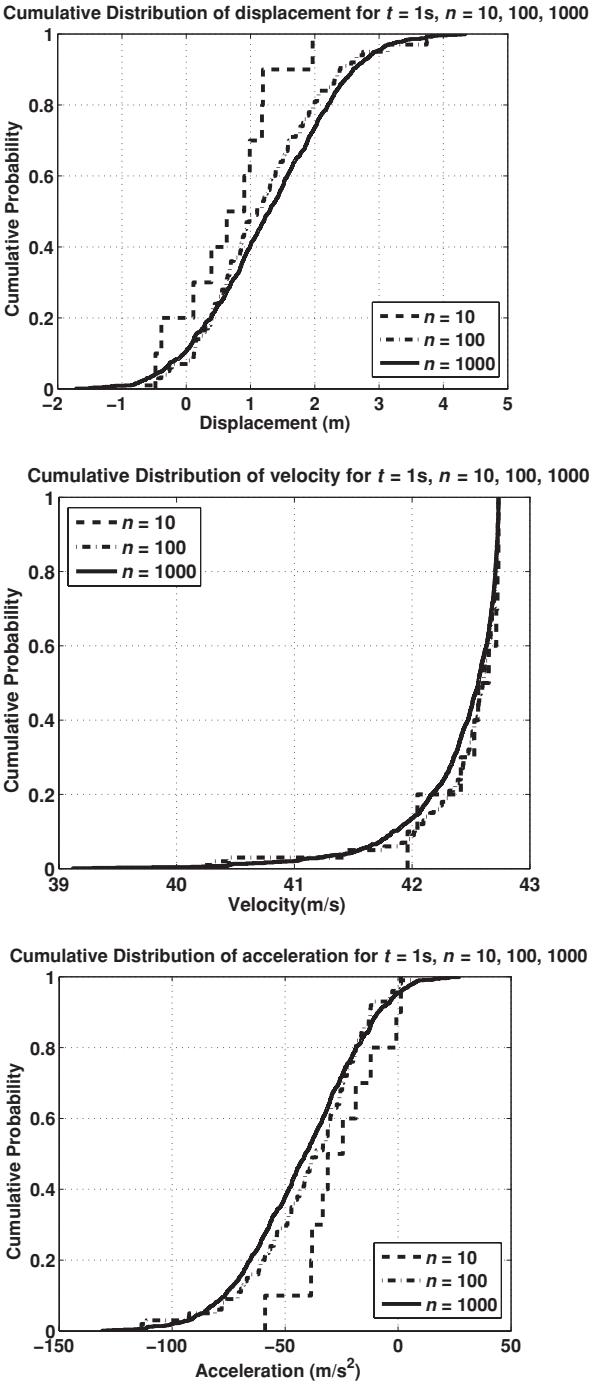

Figure 3.9 Empirical distribution functions for $t = 1$ s for aleatory uncertainty.

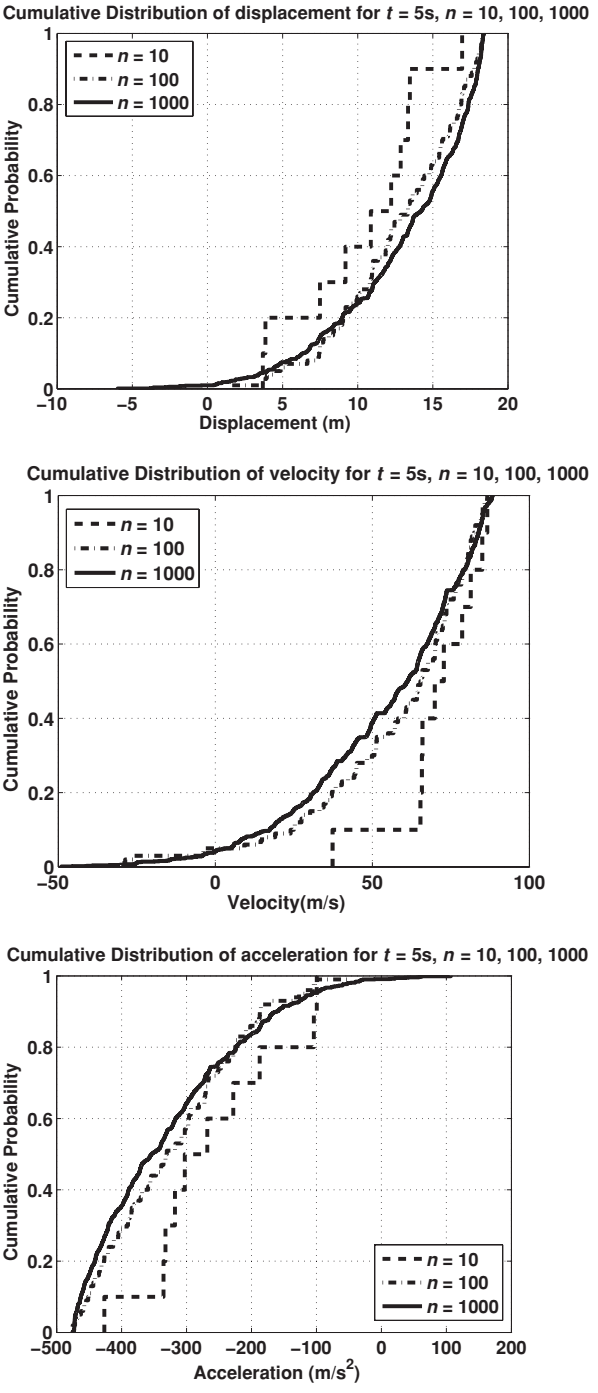

Figure 3.10 Empirical distribution functions for $t = 5$ s for aleatory uncertainty.

Table 3.3 *Maximum and minimum value of displacement, velocity, and acceleration for $t = 1$ s.*

	Max value	Min value
Displacement x		
Samples (n)		
10	3.69	−0.82
100	3.86	−1.01
1000	3.85	−1.11
10 000	3.87	−1.11
Velocity $\dot{x}$		
Samples (n)		
10	42.7	40.5
100	42.7	40.2
1000	42.7	40.2
10 000	42.7	40.2
Acceleration $\ddot{x}$		
Samples (n)		
10	8.7	−110
100	12.7	−115
1000	14.7	−115
10 000	14.8	−116

system with the masses provided by the new supplier, they could only justify an interval-valued quantity for the mass as [3.7, 4.7] kg. Stated differently, they had *no knowledge* to justify *any* PDF for the masses within the interval. Consequently, in the analysis of the system, the customer considered the uncertainty in the mass as a purely epistemic uncertainty.

With the epistemic uncertainty for the mass of the system, a probability bounds analysis (PBA) is required. Since there is no aleatory uncertainty in the system, the PBA reduces to an interval analysis. That is, if only interval-valued uncertainties exist on the inputs, then only interval-values can result for the SRQs. We used Monte Carlo sampling incorporated in MATLAB's random number generator *rand* to obtain samples over the interval-valued range of the mass. Since *rand* produces a random number scaled between 0 and 1, we shifted the output of *rand* so that it would produce a sequence of random numbers in the interval [3.7, 4.7]. The number of samples, n, was again set to 10, 100, and 1000. Note that we only used Monte Carlo sampling as a vehicle to sample over the interval-valued uncertainty. None of the samples are associated with a probability, i.e., each sample is simply considered as a *possible realization* that could occur in the system and no probability is assigned to it.

We will present the results for the interval-valued uncertainty in the mass in terms of tables of results, similar to Table 3.1 and Table 3.2, and in terms of plots of CDF,

Table 3.4 *Maximum and minimum value of displacement, velocity, and acceleration for $t = 5$ s.*

	Max value	Min value
Displacement x		
Samples (n)		
10	18.3	0.51
100	18.4	−0.15
1000	18.4	−1.48
10 000	18.4	−1.50
Velocity $\dot{x}$		
Samples (n)		
10	87.5	−23.9
100	88.3	−31.4
1000	89.3	−30.3
10 000	89.3	−31.6
Acceleration $\ddot{x}$		
Samples (n)		
10	−29.9	−470
100	−15.2	−475
1000	13.0	−475
10 000	13.4	−475

similar to Figure 3.9 and Figure 3.10. Table 3.3 and Table 3.4 show the maximum and minimum value of $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$ at $t = 1$ and 5 s, respectively, as a function of the number of samples computed, including 10 000 samples. Since we are only dealing with an interval-valued uncertain input, Table 3.3 and Table 3.4 show the maximum and minimum values of the various SRQs, based on the number of samples obtained. By comparing the results in Table 3.3 and Table 3.4 with the aleatory results shown in Table 3.1 and Table 3.2 (all for $n = 10\,000$), it is seen that the maximum and minimum values for $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$ are bounded by the aleatory results using $\mu \pm 3\sigma$. However, depending on the nature of the system and the input uncertainties, the SRQ uncertainty due to aleatory input uncertainty can be quite different compared to epistemic input uncertainty.

An important computational point should be mentioned with regard to the sampling results shown in Table 3.3 and Table 3.4. For each of the three different numbers of samples shown, a *different* random number seed was used. For $n = 10$, 100, 1000, and 10 000, seed values of 0, 1, 2, and 3, respectively, were used in MATLAB. Using different seed values for the random number generator is referred to as *replicated Monte Carlo sampling*. Using different seeds will result in a different random number sequence for sampling the input. Consequently, each ensemble of output results will be different, i.e., each ensemble result, which is composed of n samples, is a different set of computed samples. Of course,

in the limit as n becomes large, the interval-value bounds from each replicated Monte Carlo sample will become equivalent. For the case of sampling over an interval, replicated sampling is more important than for aleatory uncertainties because each sample over the interval is treated as a possible value instead of a value associated with a probability. In a probabilistic view, the probability of the sample is related to (a) the PDF characterizing the uncertainty, and (b) the number of samples obtained. As mentioned earlier in the chapter, what we mean by a *possible value* is to say that *every value* sampled can be considered to have a probability of *unity*. This feature of n sampled values, each with a probability of unity, is a source of disagreement with the Bayesian perspective.

Figure 3.11 shows the EDF of $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$ at $t = 1$ for each of the number of samples computed. These graphs are strikingly different than those shown for aleatory uncertainty, Figure 3.9 and Figure 3.10. The reason, of course, is that the EDFs for epistemic uncertainty portray an interval-valued quantity for each SRQ. Even though Figure 3.11 presents the same information as given in Table 3.3, it is worthwhile to consider what an interval-valued quantity looks like in terms of an EDF. For an uncertainty that is characterized as an epistemic uncertainty, i.e., an interval-valued quantity, the EDF will be a p-box. The p-boxes shown in Figure 3.11 are a degenerate case of general p-boxes because there is only epistemic uncertainty and no aleatory uncertainty. The general case of p-boxes for mixed epistemic and aleatory uncertainty will be briefly addressed in Section 3.5.6, with a more detailed discussion given in Chapter 12, Model accuracy assessment, and Chapter 13.

Consider the interpretation of the p-boxes shown in Figure 3.11. For values of the SRQ less than the minimum value observed, the probability is considered to be zero, because no smaller values were computed. For values of the SRQ larger than the maximum value observed, the probability is considered to be unity, because no larger values were computed. For values of the SRQ between the minimum and maximum values observed, the probability can range from zero to unity. That is, given the epistemic uncertainty in the simulation, all that can be stated about the probability within the range of observed values is that the probability itself is an interval-valued quantity, i.e., [0, 1]. Stated another way, Figure 3.11 is simply the graphical depiction of an interval in terms of an empirical distribution function. When this interpretation is explained to a traditional frequency-based statistician or a Bayesian statistician, their reaction typically is "That's not saying *anything*!" Our response is: given the poor knowledge that is stated for the input, that is *all* that can be stated. Or equivalently: all values within the observed range of values are possible, but there is *no evidence* to claim any likelihood of outcomes within the range.

As a final comment on this example, if readers attempt to reproduce the results given in either the aleatory or epistemic uncertainty examples, they may not be able to reproduce exactly the same numerical results shown. If the reader uses MATLAB and all of the same numerical input values given here, one should be able to accurately repeat the results shown. However, if a different software package is used, particularly a different random number generator, then the results could vary noticeably because each random number generator will produce its own unique sequence of pseudo-random numbers. The differences in the present results and a reader's results will be most noticeable for a low numbers of samples.

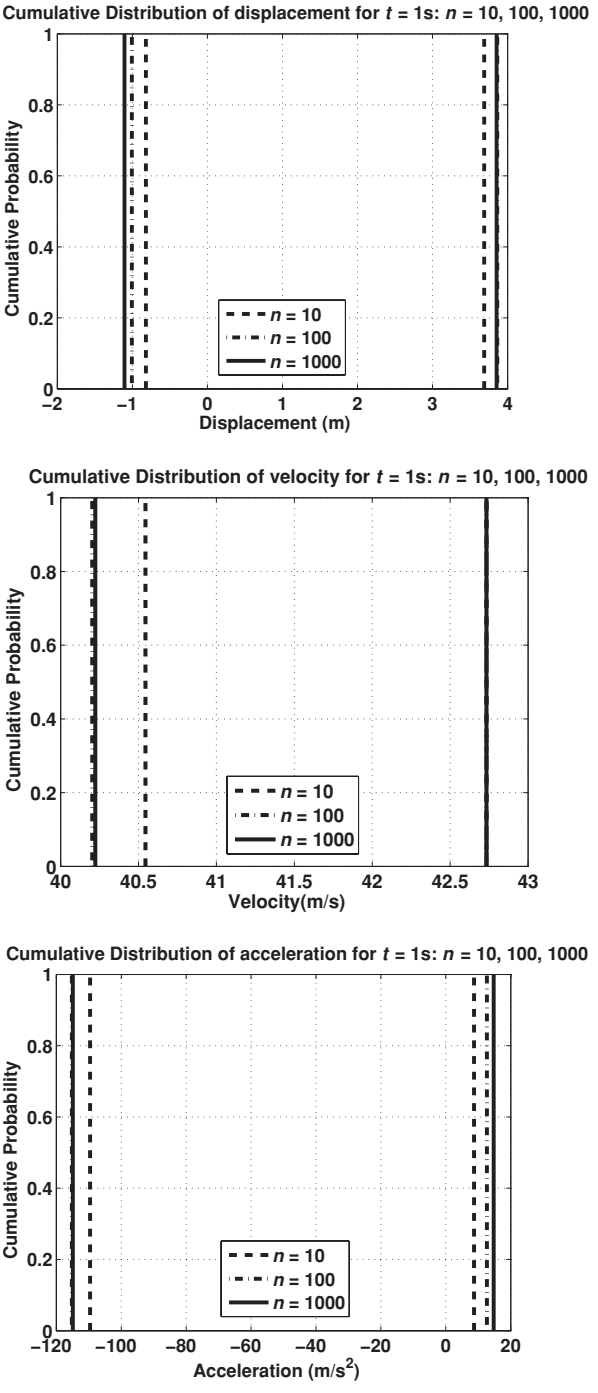

Figure 3.11 Empirical distribution functions for $t = 1$ s for epistemic uncertainty.

3.3 Risk and failure

We have referred to *risk* in various contexts, but the concept is so important, it must be defined more precisely. Essentially all modern quantitative risk analyses use the foundational concepts formulated by Kaplan and Garrick in their classic paper "On the quantitative definition of risk" (Kaplan and Garrick, 1981). They formulated the issue of quantitative risk in terms of asking three failure-related questions.

- What can go wrong?
- What is the likelihood that it will go wrong?
- If it does go wrong, what are the consequences?

They answer these questions in terms of the risk triplet

$$\langle s_i, p_i, x_i \rangle, \quad i = 1, 2, \ldots, n, \tag{3.6}$$

where s_i is the specific failure scenario being considered, p_i is the probability that the failure scenario occurs, x_i is the consequence or damage-measure from that failure scenario, and n is the number of failure scenarios being considered. s_i is simply an index for identifying a specific scenario, and x_i is a scalar that attempts to quantify in terms of dimensional units the magnitude of the consequence of a failure. For the remainder of the text we will simply use the term *failure* instead of failure scenario. The probability p_i could be stated in different ways; for example, for each use of the system, over a fixed period of a time, or over the lifetime of the system.

The conceptual framework of the risk triplet is very useful, but it is mathematically clumsy to deal with in a quantitative risk assessment (QRA) or probabilistic risk assessment (PRA) analysis. Many alternatives could be defined so that the risk could be computed by combining the three terms in Eq. (3.6). The most common method of defining risk is simply to take the product of the probability of failure occurring and the consequence of failure (Modarres *et al.*, 1999; Haimes, 2009). One has

$$\text{Risk} = p_i \bullet x_i, \quad i = 1, 2, \ldots, n. \tag{3.7}$$

The value of risk computed in Eq. (3.7) is a dimensional quantity, measured in terms of the units of the consequence of the failure x_i. The units most commonly used are monetary value, e.g., dollars or euros. For example, Eq. (3.7) could be used to estimate the total financial liability or expected total damages incurred over time as a result of a system failure. Consequences, however, can be very difficult to quantify because they can have a multitude of aspects; e.g., lost future business, environmental impact, societal impact, decrease in military security, or political impact. In addition, there are almost always unintended consequences; some of which have short-term and long-term negative effects (Tenner, 1996). The ability to identify short-term and long-term detrimental unintended consequences is extremely difficult for many reasons. High among them is that individuals, organizations, and governments tend to focus on the near-term benefits, as opposed to potential long-term risks or required changes in behavior.

Haimes (2009) and Lawson (2005) give an in-depth discussion of the three fundamental sources of failure in systems. They identify these as technical failure, individual human failure, and organizational failure. Technical failure occurs within the system, from either hardware or software, and commonly initiates some type of larger-scale system failure that has noticeable consequences. Hardware failures are commonly a result of inadequate system maintenance or inspection, or lack of needed repairs or improvements. Human failure can be of many types; e.g., system operator error, lack of proper safety training, human misuse of the system, or the system operator ignoring warning alarms. Organizational failure can occur due to gross negligence or malfeasance, but it is commonly caused by neglect or omission. As pointed out by Haimes (2009) and Lawson (2005), organizational failure is caused by humans in the organization, but the dominant feature of the failure is due to the culture and traditions within an organization. Some examples of organization failure are (a) overlooking system weaknesses or delaying maintenance because of competitive or schedule pressures; (b) poor or filtered communication between management and staff; (c) competition between groups within an organization such that a group remains silent concerning weaknesses in a competing group's design; (d) lack of incentives for a project group manager to identify weaknesses in his system's design; or (e) external political pressures on an organization causing impaired judgment of management. Each of these three fundamental sources of failure is interconnected to the others in direct and indirect ways. These connections can occur during the proposal preparation phase for a new system, design phase, testing, and operation and maintenance of the system.

A number of researchers and recent investigations of root causes of high-visibility system failures effectively argue that organizational failures are the dominant cause of most complex system failures (Dorner, 1989; Paté-Cornell, 1990; Petroski, 1994; Vaughan, 1996; Reason, 1997; Gehman *et al.*, 2003; Lawson, 2005; Mosey, 2006). Organizational failures are typically difficult to identify carefully and isolate, particularly for large organizations or the government. Organizational failure is usually connected to unintentional or intentional avoidance of an issue in some way, commonly brought on by competitive, schedule, budgetary, cultural, political, or face-saving pressures. News media coverage of failures tends to focus on the technical or human failure that initiated the event, such as "The captain of the ship was intoxicated while on duty." However, most complex systems require multiple contributing failures to occur before some type of disaster is precipitated. Many of these multiple contributors can be traced back to organizational failures.

3.4 Phases of computational simulation

The operations research (OR) and systems engineering (SE) communities have developed many of the general principles and procedures for M&S. Researchers in this field have made significant progress in defining and categorizing the various activities and phases of M&S. For recent texts in this field, see Bossel (1994); Zeigler *et al.* (2000); Severance (2001); Law (2006); and Raczynski (2006). The areas of emphasis in OR and SE include

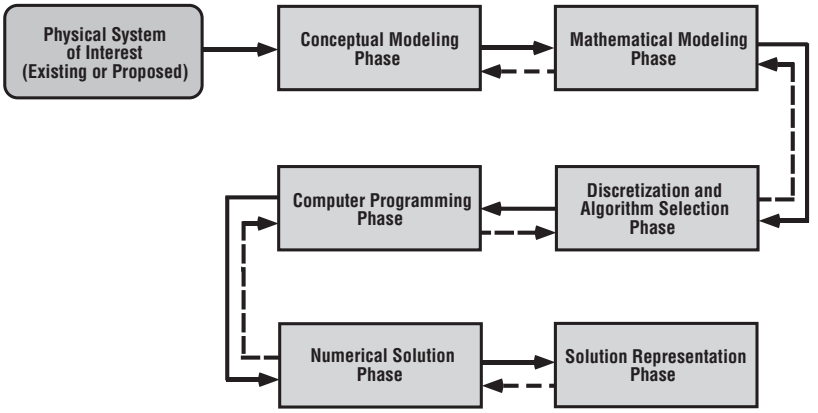

Figure 3.12 Phases for a computational simulation (Oberkampf *et al*., 2000; 2002).

definition of the problem entity, definition of the conceptual model, assessment of data and information quality, discrete event simulation, and methods for using simulation results as an aid in decision making. From a computational simulation perspective, this work is not focused on models that are specified by PDEs. However, the OR and SE work is very helpful in providing a useful philosophical approach for identifying sources of uncertainty, as well as developing some of the basic terminology and model development procedures.

Based on the work of Jacoby and Kowalik (1980), Oberkampf *et al.* (2000, 2002) developed a comprehensive, new framework of the general phases of computational simulation. This structure is composed of six phases that represent a synthesis of the tasks recognized in the operations research, risk assessment, and numerical methods communities. Figure 3.12 depicts the phases of computational simulation appropriate to systems analyzed by the numerical solution of PDEs. The physical system can be an existing system or process, or it can be a system or process that is being proposed. The phases represent collections of activities required in a generic large-scale computational analysis. The ordering of the phases implies an information and data flow indicating which tasks are likely to impact analyses, decisions, and methodologies occurring in later phases. Each succeeding phase could be described as a *mapping of the preceding phase into a new phase* of activities. Any assumptions, approximations, aleatory uncertainties, recognized epistemic uncertainties, or blind epistemic uncertainties introduced in one phase are then propagated to *all* succeeding phases. Suppose, for example, it is discovered in some latter phase that an assumption or approximation was inappropriate, or a blind uncertainty (e.g., oversight or mistake) was introduced at some earlier phase. Then one must return to that phase and re-evaluate all subsequent phases to determine what changes must be made. This type of feedback interaction between the phases is shown by the dashed lines in Figure 3.12.

In the following sections, characteristics and activities of each of the phases are discussed. The emphasis in the discussion is on the identification and propagation of different types of uncertainty through the computational simulation process. In Section 2.3, taxonomy of

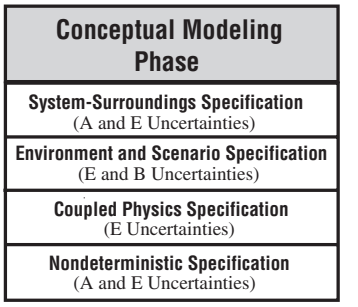

Figure 3.13 Conceptual modeling phase and primary sources of uncertainty.

uncertainties was constructed with the primary separation made between aleatory uncertainties and epistemic uncertainties. Epistemic uncertainties were further divided into (a) recognized uncertainty, i.e., uncertainty due to incomplete knowledge in which a conscious decision has been made to either characterize it in some way, or to ignore it with justification; and (b) blind uncertainty, i.e., uncertainty due to incomplete knowledge, but where it is *not* recognized that the knowledge is incomplete and relevant. The distinction between aleatory and epistemic uncertainty is important not only in assessing how each contributes to an estimate of total predictive uncertainty, but also how each should be mathematically represented and propagated through each phase.

3.4.1 Conceptual modeling phase

Activities conducted in the conceptual modeling phase are shown in Figure 3.13, along with the primary sources of uncertainty introduced in each activity. Note that in Figure 3.13, and all subsequent graphics of a similar nature, the text in brackets indicates the primary types of uncertainty that occur in that activity. *A Uncertainty* is aleatory uncertainty. *E Uncertainty* is recognized epistemic uncertainty. *B Uncertainty* is blind epistemic uncertainty, which is commonly referred to as unknown-unknowns.

The first activity is the specification of the physical system of interest and its surroundings. The primary conceptualization that must be specified in this activity is the demarcation between the system and the surroundings. As discussed in Section 3.1.1 above, the surroundings are not modeled as part of the system. The system responds to the surroundings. The uncertainties associated with the system and surroundings specification consist primarily of epistemic uncertainties that arise in defining the scope of the problem. For a complex engineered system, epistemic uncertainties commonly occur because of factors such as the following. Was the system incorrectly manufactured or assembled? How well was the system maintained? Was the system damaged in the past and not recorded? The system and surroundings specification would also contain aleatory uncertainties, such as variability due to manufacturing, systems exposed to weather conditions, and random excitation of a system by its surroundings. Unless a large number of empirical samples are available to

characterize an aleatory uncertainty, then it is likely that the uncertainty would be a mixture of aleatory and epistemic uncertainty.

The second activity is the determination of the environments and the scenarios that will be considered in the computational simulation. As discussed earlier in this chapter, the three classes of environment are normal, abnormal, and hostile. Different conceptual models are almost always required if more than one class of environment is considered. Scenario specification identifies physical events or sequences of events that could *possibly* be considered for a given environment, see Figure 3.1. Identifying possible scenarios or event sequences is similar to developing an event-tree or fault-tree structure in the probabilistic risk assessment of high consequence systems, such as in nuclear reactor safety analyses. Event and fault trees include not only technical (hardware and software) failures, but also human actions that could be taken either within or outside the system, i.e., as part of the surroundings. Even if a certain sequence of events is considered extremely remote, it should still be included as a possible event sequence in the fault tree. Whether or not the event sequence will eventually be analyzed is not a factor that impacts its inclusion in the conceptual modeling phase. In this activity both epistemic and blind (epistemic) uncertainties are the most likely to occur. This is particularly true in identifying possible scenarios within abnormal and hostile environments. Creativity and imagination are especially useful qualities of individuals involved in analyzing abnormal and hostile environments.

The third activity is the specification of the possible types of coupling of different physical processes that will be incorporated in the modeling. During the coupled physics specification, no mathematical equations are written. After the system and surroundings are specified, options for various levels of possible physics couplings should be identified, even if it is considered unlikely that all such couplings will be considered subsequently in the analysis. If a physics coupling is not considered in this phase, it cannot be addressed later in the process.

The fourth activity is the specification of all aspects in the modeling that will be considered as nondeterministic. The nondeterministic specification applies to every aspect of the first three activities considered in conceptual modeling, assuming that the activity can be characterized as an aleatory uncertainty or a recognized epistemic uncertainty. Blind uncertainties, of course, are not characterized because they are not recognized as an uncertainty. These determinations must be based on the general requirements for the computational simulation effort. The question of what representation will be used for the uncertainty is deferred until later phases.

3.4.2 Mathematical modeling phase

The primary task in this phase is to develop a precisely stated mathematical model, i.e., analytical statements based on the conceptual model formulated in the previous phase. The four activities in mathematical modeling are shown in Figure 3.14. The number of analyses

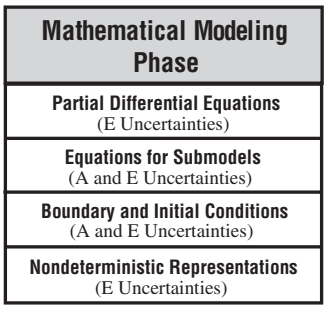

Figure 3.14 Mathematical modeling phase and primary sources of uncertainty.

to be conducted depends on how many combinations of environments and scenarios were identified in the conceptual model phase. For large-scale analyses, the number could be quite large and, as a result, prioritization of the more important analyses needs to be conducted. Typically, this prioritization is based on the risk (i.e., estimated probability multiplied by estimated consequence) that the environment–scenario pair represents to the success of the system of interest.

The complexity of the PDE models depends on the physical complexity of each phenomenon being considered, the number of physical phenomena being considered, and the level of coupling of different types of physics. The system-surroundings specification and the physics coupling specification should have been completed in the conceptual modeling phase. Some examples of epistemic uncertainties that occur in physics modeling are (a) fracture dynamics in any type of material; (b) coupling of the liquid, solid, and fluid phases in multiphase flow; (c) phase change of materials that are not in equilibrium; and (d) choosing to model a problem in 2-D instead of 3-D.

A complex mathematical model given by a set of PDEs is usually complemented by a number of mathematical submodels. Examples of submodels are (a) analytical equations or tabular data for mechanical, thermodynamic, electrical, and optical properties of materials; (b) ODEs and PDEs for constitutive properties of materials; and (c) PDEs for fluid dynamic turbulence modeling. The submodels, along with the boundary conditions, initial conditions, and any system excitation equations, complete the equation set for the system. BCs, ICs, and system excitation commonly exhibit aleatory and epistemic uncertainties. For abnormal and hostile environments, BCs, ICs, and system excitation are almost always dominated by epistemic uncertainties.

Any mathematical model, regardless of its physical level of detail, is *by definition* a simplification of reality. Any complex engineering system, or even an individual physical process, contains phenomena that are not represented in the model. As a result, specification of the mathematical models involves approximations and assumptions. These both result in epistemic uncertainties being introduced into the modeling process. Sometimes, in large-scale computational simulation projects, one can hear statements such as "The project will use such large-scale, massively parallel computers, that full physics simulations will be

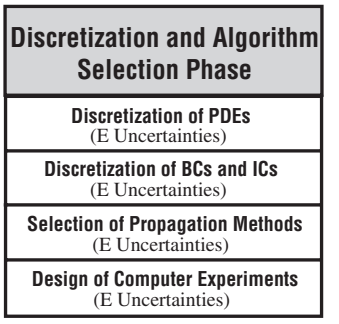

Figure 3.15 Discretization and algorithm selection phase and primary sources of uncertainty.

computed." These kinds of statements can only be considered as advertising hyperbole. The enduring truth of modeling was succinctly stated many years ago by George Box (1980): "All models are wrong, some are useful."

Another function addressed during this phase of analysis is selecting appropriate representations for the nondeterministic elements of the problem. Several considerations might drive these selections. Restrictions set forth in the conceptual modeling phase of the analyses may put constraints on the range of values or types of representations that might be used in the analysis. Within these constraints the quantity and/or limitations of available or obtainable data will play an important role. If sufficient sampling data is available for aleatory uncertainties, then a PDF or CDF can be constructed. In the absence of data, expert opinion or a similar type of information may be used. For this type of information, one could either represent the information as an interval, with no likelihood information claimed over the interval, or use a p-box. It would be highly suspect if an expert claimed that they could specify a precise probability distribution, i.e., a distribution with fixed values for the parameters of the distribution.

3.4.3 Discretization and algorithm selection phase

The discretization and algorithm selection phase maps the mathematical model developed in the previous phase into a fully discrete mathematical model. Figure 3.15 shows the four activities that are completed in this phase. These activities are grouped into two general tasks related to converting the mathematical model into a form that can be addressed through computational analysis. The first task involves conversion of the PDEs from the mathematical model into a discrete, or numerical, model. Simply stated, the mathematics is translated from a calculus problem into an arithmetic problem. In this phase, all of the spatial and temporal discretization methods are specified for discretization of the domain of the PDEs, including the geometric features, mathematical submodels, BCs, ICs, and system excitation. The discrete form of the PDEs is typically given by finite element, finite volume, or finite difference equations. In this task the discretization algorithms and methods are prescribed, but the spatial and temporal step sizes are simply given as quantities to be

specified. The discretization phase focuses on the conversion of the mathematical model from continuum mathematics, i.e., derivatives and integrals, to discrete mathematics. The methods for the numerical solution of the discretized equations are discussed in a later phase.

Although we may not consciously specify all of the discretization methods in some computational analyses, such as when using commercial software packages, we strongly believe it is an important step because it can greatly assist in detecting certain types of numerical error. This conversion process is the root cause of certain difficulties in the numerical solution of PDEs. If the mathematical model contains singularities, or the solution is near or in a chaotic state, then much more care is required when choosing the numerical algorithms. Singularities commonly exist in the mathematical models, but they rarely exist in discrete mathematics. Yee and Sweby (1995; 1996; 1998); Yee *et al.* (1997) and others have investigated the numerical solution of nonlinear ODEs and PDEs that are near chaotic behavior. They have clearly shown that the numerical solution of these equations can be quite different from exact analytical solutions of the mathematical model even when using established methods that are well within their numerical stability limits, and using what is believed to be a mesh-resolved solution. Although it is beyond the scope of this book, much more research is needed in the simulation of chaotic solutions.

The third task of this phase of the analysis is the specification of uncertainty propagation methods and the design of computer experiments in order to accommodate the nondeterministic aspects of the problem. Both activities address conversion of the nondeterministic elements of the analysis into multiple runs, or solutions, of a deterministic computational simulation code. Selection of an uncertainty propagation method involves the determination of an approach, or approaches, to propagating uncertainties through the model. Examples of propagation methods include reliability methods (Melchers, 1999; Haldar and Mahadevan, 2000a; Ang and Tang, 2007; Choi *et al.*, 2007) and sampling methods such as Monte Carlo or Latin Hypercube (Cullen and Frey, 1999; Ross, 2006; Ang and Tang, 2007; Dimov, 2008; Rubinstein and Kroese, 2008). Traditional emphasis in uncertainty quantification and risk assessment is on propagation of parametric uncertainties, but in many complex physics simulations, model-form uncertainties tend to be the dominant contributor to uncertainty in SRQs. In this phase, methods for propagating model-form uncertainties are also specified. If any methods are used that approximate the propagation of input to output uncertainties, then they should also be specified in this phase. A very common approximation method is the use of response surface methods to reduce the number of numerical solutions of the discrete model that are needed to propagate uncertainties.

The design of computer experiments, i.e., statistical experiments, is driven to a large extent by the availability of computer resources and by the requirements of the analysis. Establishing an experimental design often involves more than just implementation of the propagation method specified above (Mason *et al.*, 2003; Box *et al.*, 2005). 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

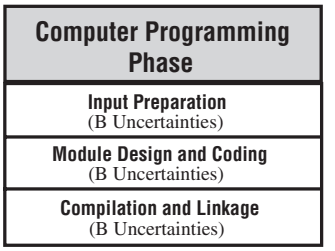

Figure 3.16 Computer programming phase and primary sources of uncertainty.

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.

As noted in Figure 3.15, the primary type of uncertainty introduced in this phase is epistemic uncertainty. These uncertainties are a specific type of epistemic uncertainty, i.e., they are due to the fidelity and accuracy of the numerical approximations. These numerical approximations are due to the *choice* of numerical methods to execute the mapping from continuum mathematics to discrete mathematics. As a result, they are analogous to choices for constructing mathematical models of physics processes. These numerical approximations are *not* analogous to numerical solution errors, such as mesh resolution errors, because numerical solution errors can usually be ordered in terms of accuracy. Choices of numerical methods and algorithms cannot always be ordered with respect to anticipated accuracy, reliability, and robustness.

3.4.4 Computer programming phase

The computer programming phase maps the discrete mathematical model formulated in the previous phase into software instructions executable on a digital computer. Figure 3.16 lists the activities conducted in this phase, as well as the primary sources of uncertainty introduced in this phase. These activities are divided into two groups: preparation of input for the computer code and computer programming activities. Preparation of input is part of this phase because it sets all of the numerical values of the input quantities that will be used in the actual computation, which occurs in the next phase. The dominant uncertainty that occurs in the preparation of input activity is the introduction of blind uncertainties, e.g., mistakes or blunders in the preparation of the input. Some researchers and analysts experienced only with relatively simple model problems do not appreciate the concern with input errors due to humans. This is, however, an important source of blind uncertainties when one is dealing with a complex and wide range of physical, modeling, and numerical details in a large code, multiple codes that are sequentially coupled, simulations that heavily rely

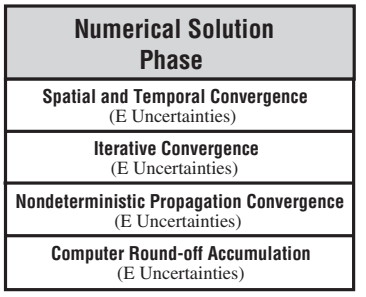

Figure 3.17 Numerical solution phase and primary sources of uncertainty.

on geometries specified by sophisticated solid modeling software, and tens or hundreds of material models needed for input (Reason, 1997). The complexity of the input data and the resulting opportunity for error with such codes is extraordinary. The importance of input preparation has been recognized for some time in the thermal-hydraulics field concerned with the safety analyses for nuclear power reactors. Formal, structured, and rigorous procedures have been developed in this field to ensure the input data accurately reflects the intended input.

The second and third activities relate to all of the software elements used in the simulation, but here we concentrate on the application code itself. In the application code the computer program modules are designed and implemented through a high-level programming language. This high-level source 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. These activities are becoming more prone to blind uncertainties due to massively parallel computers, including elements such as (a) optimizing compilers, (b) message passing and memory sharing, and (c) the effect of individual processors or memory units failing during a computer simulation. The correctness of the computer-programming activities is most influenced by blind uncertainties. In addition to programming errors, there is the subtler problem of undefined variables. This occurs when particular code syntax is undefined within the programming language, leading to executable code whose behavior is compiler-dependent. Compilation and linkage introduce the potential for further errors unbeknownst to the developer. Primary among these are bugs and errors in the numerous libraries of object code linked to the application. Such libraries allow the developer to reuse previously developed data handling and numerical analysis algorithms. Unfortunately, the developer also inherits the undiscovered or undocumented errors in these libraries. There is also the possibility that the developer misunderstands how the library routines should be used, or he makes an error in the values that are needed by the library routines.

3.4.5 Numerical solution phase

This phase maps the software programmed in the previous phase into a set of numerical solutions using a digital computer. Figure 3.17 shows the various activities that are

conducted during this phase. During the computation of the solution, *no* quantities are left arithmetically undefined or continuous; only discrete values of all quantities exist, all with finite precision. For example: (a) geometries only exist as a collection of points, (b) all independent and dependent variables that exist in the PDEs now only exist at discrete points, and (c) nondeterministic solutions only exist as an ensemble of individual discrete solutions. As stated by Raczynski (2006), "In the digital computer nothing is continuous, so continuous simulation using this hardware is an illusion."

The primary uncertainty introduced during this phase is epistemic uncertainty, specifically numerical solution errors. Roache categorizes these types of numerical solution errors as *ordered errors* (Roache, 1998). Most of the contributing errors in the four activities shown in Figure 3.17 can be ordered in terms of magnitude of their effect on the simulation outcomes; e.g., discretization size in terms of space or time, number of iterations in an implicit numerical procedure, number of computed samples in the propagation of input-to-out uncertainties, and word length of the computer system. The numerical solution errors introduced in the four activities shown in Figure 3.17 can be divided into three categories. The first category contains those that are due to the spatial and temporal discretization of the PDEs. The second category contains those that are due to the approximate solution of the discrete equations. Iterative convergence using an implicit method and computer round-off errors are of this type and they account for the difference between the exact solution to the discrete equations and the computed solution. Iterative solution errors can be due to, for example, iterative solution of a nonlinear matrix equation, or the iterative solution of a nonlinear time-dependent solution. The third category contains those that are due to the finite number of individual deterministic solutions obtained. The difference between using a finite number of solutions computed and the exact nondeterministic solution, for whatever probability distribution or statistic that is of interest, is the nondeterministic solution error. The most common example is the error due to a finite number of Monte Carlo samples used to approximate a nondeterministic solution. If stochastic expansion methods are used for uncertainty propagation, e.g., polynomial chaos expansions and the Karhunen–Loeve transform, then the nondeterministic solution error depends on the number of solutions computed to the discrete equations (Haldar and Mahadevan, 2000b; Ghanem and Spanos, 2003; Choi *et al.*, 2007). Multiple solutions can also be required if the mathematical modeling phase includes the nondeterministic effect of alternative mathematical model forms in order to estimate model-form uncertainty.

3.4.6 Solution representation phase

This final phase maps the raw numerical solutions, i.e., the numbers themselves, which are computed in the previous phase, into numerical results usable by humans. Figure 3.18 shows the activities and the primary source of uncertainty introduced in each activity. The solution representation phase is included in the six phases of computational simulation because of the sophisticated post-processing that is increasingly done to comprehend complex simulations, as well as the recognition that this phase can introduce unique types of uncertainties. Input

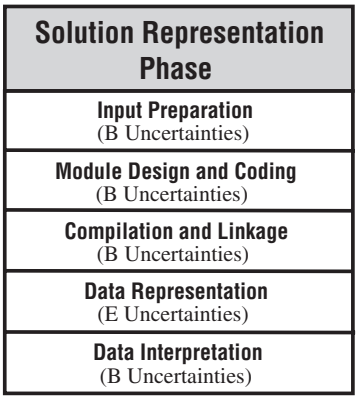

Figure 3.18 Solution representation phase and primary sources of uncertainty.

preparation, module design and coding, and compilation and linkage refer to the same types of activities listed in the computer programming phase, but here they refer to all of the post-processing software that is used. As before, all three of these activities have uncertainty contributions primarily from blind uncertainties.

Data representation is concerned with the construction of the functions that are intended to represent the dependent variables from the PDEs, as well as post-processing of the dependent variables to obtain other SRQs of interest. Post-processing includes three-dimensional graphical visualization of solutions, animation of solutions, use of sound for improved interpretation, and use of virtual reality tools that allow analysts to *go into the solution space*. Epistemic uncertainties are introduced in data representation, primarily ordered numerical errors, which can result in the inaccurate or inappropriate construction of either the dependent variables or other SRQs of interest. Some examples of numerical errors are (a) oscillations of the function in-between discrete solution points due to the use of a high-order polynomial function in the post-processor, (b) inappropriate interpolation of the discrete solution between multi-block grids, (c) inappropriate interpolation of the discrete solution when the solution to the PDEs is a discontinuous function, and (d) excessive amplification or damping of the interpolation function for the dependent variables that are used to compute other SRQs of interest. Concern for errors in data representation can be better understood by posing the question: what is the mathematically correct reconstruction of the functions using the discrete solution points, given that these point values are intended to represent a solution to a PDE? When viewed from this perspective, one recognizes the potential reconstruction errors better because this is *not* the perspective taken in modern data visualization packages. The view of these general-purpose packages is that the reconstruction is based on ease of use, speed, convenience, and robustness of the package. Stated differently, in data visualization packages there is no interest or concern with respect to insuring that the interpolation function conserves mass, momentum, or energy.

Data interpretation errors are made by the interpreter of the computational results, based on observations of the representation of the solution and the computed SRQs. The interpreter of the results could be, for example, the computational analysts using the code or a decision maker relying on the results. Data interpretation errors are blind uncertainties introduced by individuals or groups of individuals. Two examples of interpretation error are (a) concluding that a computed solution is chaotic when it is not (and vice versa); and (b) not recognizing the important frequency content in a complex SRQ. Importantly, our definition of data interpretation errors does *not* include poor decisions made by the user based on the simulation, such as incorrect design choices or inept policy decisions.

Individual deterministic solution results are typically used by researchers, physical scientists, and numerical analysts; whereas the collective nondeterministic results are more commonly used by system designers, decision makers, or policy makers. Each of these audiences usually has very different interests and requirements. The individual solutions provide detailed information on deterministic issues such as (a) the coupled physics occurring in the system; (b) the adequacy of the numerical methods and the mesh resolution needed to compute an accurate solution; and (c) how the SRQs vary as a function of the independent variables, the physical parameters in the model, and the boundary and initial conditions. The collective nondeterministic results are used, for example, to (a) understand the magnitude of the effect of aleatory and epistemic uncertainties on the SRQs of interest, particularly model form uncertainty; and (b) examine the results of a sensitivity analysis. A sensitivity analysis is commonly the most useful result to system designers and decision makers because it helps guide their thinking with regard to issues such as (a) changes needed to obtain a more robust design, (b) tradeoffs between design parameters or various operating conditions, and (c) allocation of resources to reduce the dominant uncertainties in system performance, safety, or reliability.

3.5 Example problem: missile flight dynamics

To demonstrate each of the phases of computational simulation, a system-level example is given of the flight dynamics of a rocket-boosted, aircraft-launched missile. This example is adapted from Oberkampf *et al.*, 2000; 2002. For a detailed discussion of this example, see these references. Figure 3.19 shows all six phases of computational simulation and the activities conducted in each. The missile is a short range, unguided, air-to-ground rocket. The missile is powered by a solid fuel rocket motor during the initial portion of its flight, and is unpowered during the remainder of its flight. The analysis considers the missile flight to be in the unspecified future. Thus, we are attempting to simulate future plausible flights, not analyze an event in the past (such as an accident investigation), or update models based on past observations of the system.

An additional example of a system in an abnormal, i.e., an accident, environment is given in Oberkampf *et al.* (2000).

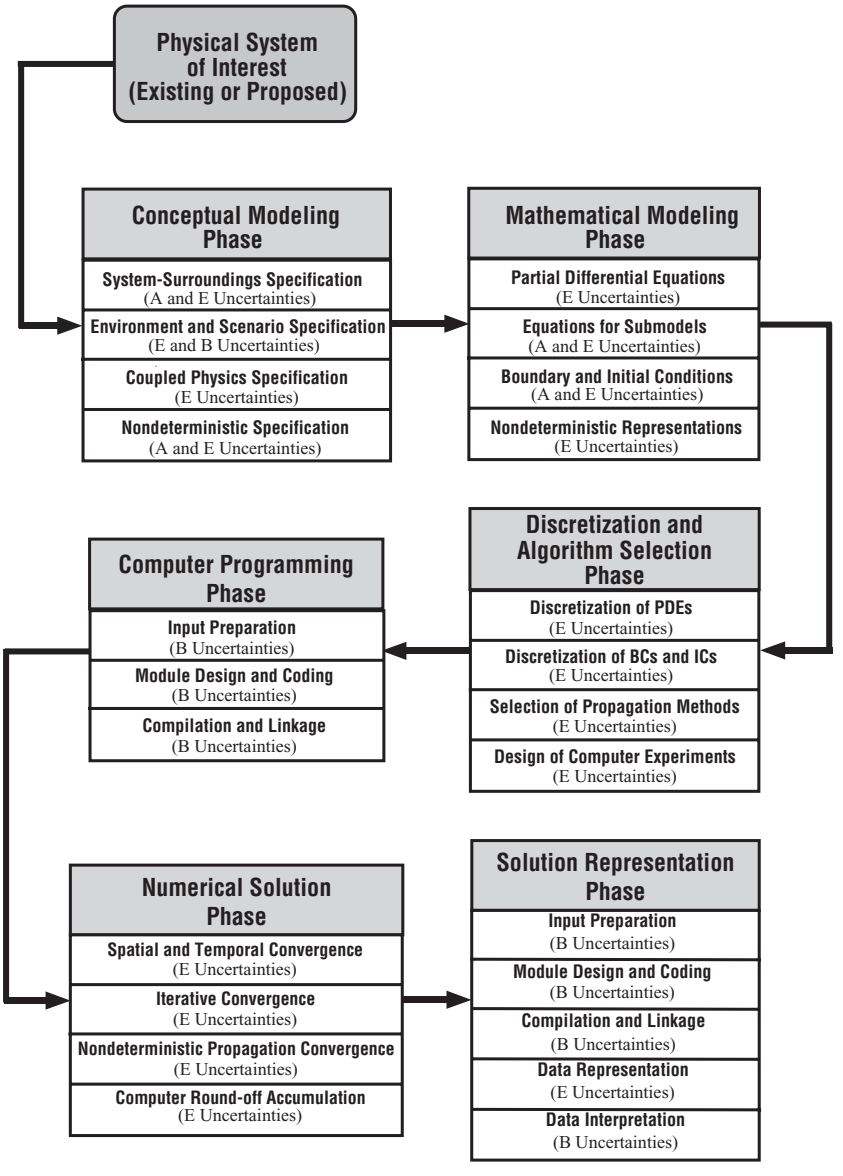

Figure 3.19 Phases and activities in computational simulation.

3.5.1 Conceptual modeling phase

Figure 3.20 shows three possible system-surroundings specifications for the missile flight example. Other specifications could be made, but these give a wide range of options that could be used for various types of simulation. The specifications are listed from the most physically inclusive, with regard to the system specification and the physics that

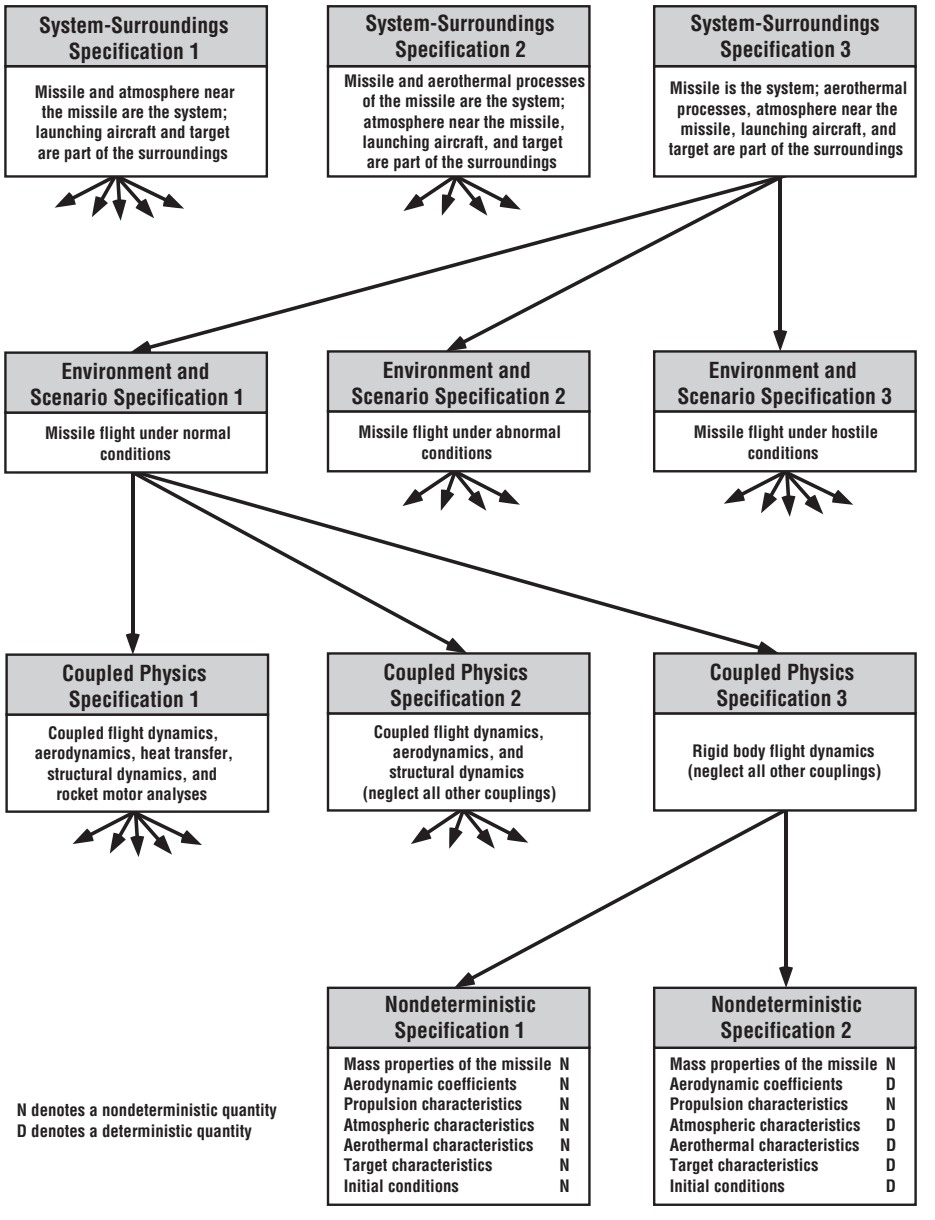

Figure 3.20 Conceptual modeling activities for the missile flight example.

could be coupled, to the least inclusive. For each row of blocks shown in Figure 3.20, the most physically inclusive are given on the left with decreasing physical complexity moving toward the right. System-surroundings 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 to be part of the surroundings. An example of an analysis that would be allowed with this specification is where 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 the rocket exhaust could be influenced by the presence of the aircraft and its flow field, but the aircraft structure, for example, could not change its geometry or deform due to the rocket exhaust. Another example allowed by this specification would be the analysis of the missile flight inside a structure, e.g., launch from inside of a structure; or a flight inside of a tunnel, e.g., a target is inside a tunnel.

System-surroundings specification 2 considers the missile and the aerothermal processes occurring near the surface of 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 surroundings. This specification allows analyses that couple the missile and the aerothermal effects on the missile. For example, one could consider the structural deformation of the missile due to aerodynamic loading and thermal heating of the structure. Then one could couple the missile deformation and the flow field so that the aerodynamic loading and thermal heating could be simulated on the deformed structure.

System-surroundings specification 3 considers the missile to be the system, whereas the aerothermal processes external to the missile, the atmosphere near the missile, the launching aircraft, and the target are considered part of the surroundings. Even though this is the simplest specification considered here, it still allows for significant complexities in the analysis. Note that the missile flight example presented here will only pursue system-surroundings Specification 3.

The environment specification (Figure 3.20) identifies three general environments: normal, abnormal, and hostile. For each of these environments one then identifies all possible scenarios, physical events, or sequences of events that may affect the goals of the simulation. For relatively simple systems, isolated systems, or systems with very controlled surroundings or operational conditions, this activity can be straightforward. Complex engineered systems, however, are commonly exposed to a myriad of scenarios within each of the normal, abnormal, and hostile environments. Constructing environment and scenario specifications for these complex systems is a mammoth undertaking. A many-branched event tree and/or fault tree can be constructed with each scenario having a wide range of likelihoods and consequences. Even though the risk (probability times consequence) for many scenarios may be low, these scenarios should be identified at this phase. Often, when various scenarios are identified, other scenarios are discovered that would not have been discovered otherwise. The decision of which scenarios to pursue should be made *after* a very wide range of scenarios has been identified.

Normal environment conditions are those that can be reasonably expected during nominal operations of the aircraft and missile system. Some examples are (a) typical launch conditions from various types of aircraft that are expected to carry the missile, (b) nominal operation of the propulsion and electrical system, and (c) reasonably expected weather conditions while the missile is attached to the aircraft and during flight to the target. Examples of flight under abnormal conditions are (a) improperly assembled missile components or subsystems; (b) explosive failure of the propulsion system during operation, particularly while still attached or very near the aircraft; and (c) flight through adverse weather conditions, like hail or lightning. Examples of flight under hostile conditions are (a) detonation of nearby defensive weapon systems; (b) damage to missile components or subsystems resulting from small-arms fire; and (c) damage, either structural or electrical, from laser or microwave defensive systems. Note that the missile flight example will only pursue environment specification 1, normal environment. Furthermore, no unusual conditions will be considered within the realm of normal conditions.

Figure 3.20 identifies three levels of physics coupling, although more alternatives could be identified. Coupled physics specification 1 couples essentially all of the physics that could exist in this decision-thread of the analysis, i.e., system-surroundings specification 3 and environment and scenario specification 1. 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, and the effect of 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. One could then, for example, recompute 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 within the missile, but the missile structure is assumed rigid. The missile is allowed to respond only to inputs or forcing functions from the surroundings. 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, a few comments should be made concerning the possible sources of epistemic and blind uncertainty that could occur in the three activities discussed so far. Epistemic uncertainties arise primarily because of (a) situations, conditions, or physics that are poorly known or understood; (b) situations or conditions that are consciously excluded from the analysis; and (c) approximations made in situations or conditions considered. Blind uncertainties arise primarily because of situations or conditions that are *not* imagined or recognized, but are possible. The more complex the system, the more possibilities exist for blind uncertainties to occur.

Indeed, a common weakness of modern technological risk analyses is overlooking, either by oversight or negligence, unusual events, effects, possibilities, or unintended consequences. For example, automatic control systems designed to ensure safe operation of complex systems can fail (either hardware or software failures) in unexpected ways, or the safety systems are overridden during safety testing or maintenance. For systems in abnormal or hostile environments, the likelihood of blind uncertainties increases greatly compared to normal environments.

For the missile flight example we list only two alternative nondeterministic specifications, as shown in Figure 3.20. Nondeterministic Specification 1 includes the following (indicated by an N at the bottom of Figure 3.20): mass properties of the missile, aerodynamic force and moment coefficients, propulsion characteristics, atmospheric characteristics, aerothermal heating characteristics, target characteristics, and initial conditions at missile launch. Nondeterministic Specification 2 considers only two parameters as uncertain; the mass properties of the missile and the propulsion characteristics of the motor. All other parameters are considered as deterministic (indicated by a D in Figure 3.20). The missile flight example will only pursue nondeterministic specification 2.

3.5.2 Mathematical modeling phase

In the mathematical modeling phase, the PDEs, equations and data for submodels, BCs, ICs, and forcing functions are specified. Even with the specifications made in the conceptual model phase, there is always a wide range of mathematical models that one can choose from. Typically, the range of modeling choices can be arranged in order of increasing fidelity of the physics being considered.

For the missile flight example, two mathematical models are chosen; a six-degree-of-freedom (6-DOF) model and a three-degree-of-freedom (3-DOF) model (Figure 3.21). Both models are consistent with the conceptual model being analyzed: system-surroundings specification 3, environment specification 1, coupled physics specification 3, and nondeterministic specification 2 (Figure 3.20). For the 3-DOF and 6-DOF mathematical models of flight dynamics, one can unequivocally order these two models in terms of physics fidelity. The ability to clearly order the physics fidelity of multiple models can be used to great advantage in the following situations. First, there are often conditions where multiple models of physics should give very similar results for certain SRQs. By comparing the results from multiple models one can use this as an informal check between the models. Second, there are sometimes conditions where we expect multiple models of physics to compare well, but they don't. If we conclude that both models are correct, given their assumptions, these conditions can lead to a deeper understanding of the physics, particularly coupled physics. And third, by exercising multiple models of physics we can develop confidence in where and why the lower fidelity model gives essentially the same results as the higher fidelity model. If the higher fidelity model is much more computationally demanding, we can use the lower fidelity model for nondeterministic simulations over the range of conditions where we have developed trust in the model.

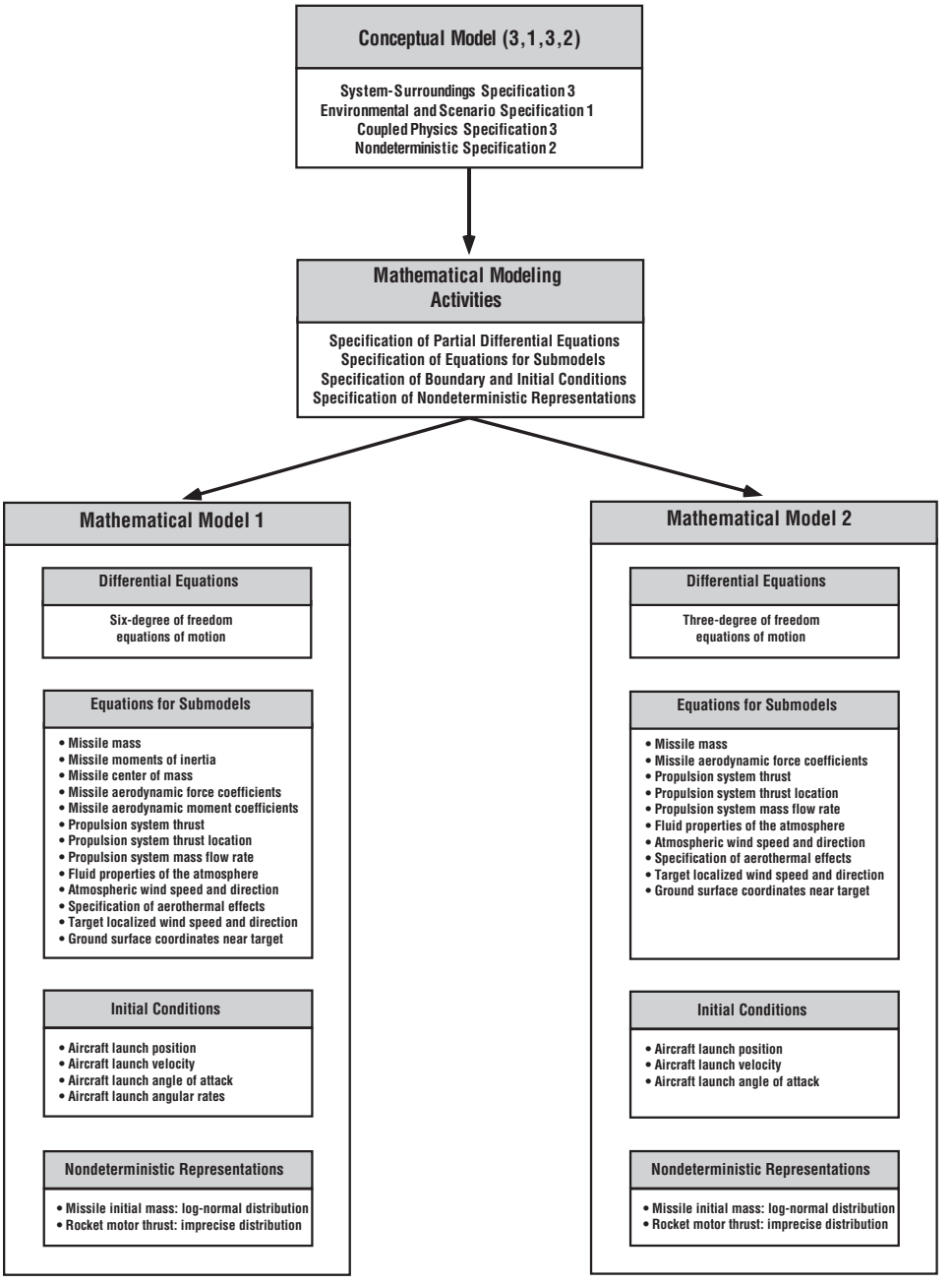

Figure 3.21 Mathematical modeling activities for the missile flight example.

The translational equations of motion can be written as

$$m\frac{\mathrm{d}\vec{V}}{\mathrm{d}t} = \sum \vec{F}, \tag{3.8}$$

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

$$[I]\frac{\mathrm{d}\vec{\omega}}{\mathrm{d}t} = \sum \vec{M} + \vec{\omega} \times \{[I] \bullet \vec{\omega}\}, \tag{3.9}$$

where $[I]$ is the inertia tensor of the vehicle, $\vec{\omega}$ is the angular velocity, and $\sum \vec{M}$ is the sum of all moments acting on the vehicle. Eq. (3.8) represents the 3-DOF equations of motion, and the coupling of Eq. (3.8) and Eq. (3.9) represent the 6-DOF equations of motion. Although the 3-DOF and 6-DOF equations are ODE models instead of the PDE models stressed in the present work, key aspects of the present framework can still be exercised.

Figure 3.21 lists all of the submodels and initial conditions 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. In some situations, the increase in predictive capability from higher fidelity physics models can be offset by the increase in information that is needed to characterize the uncertainties required as input to the high fidelity model. That is, unless the additional uncertainty information that is needed in higher fidelity models is available, the poorer characterization of uncertainty can overwhelm the increase in physics fidelity as compared to the lower fidelity model. As a result, higher fidelity physics models *may yield poorer predictive capability* than lower fidelity models, a seemingly counterintuitive conclusion.

The two nondeterministic parameters considered in the missile flight example are the initial mass of the missile and the rocket motor thrust characteristics (Figure 3.21). Both parameters appear in each of the mathematical models chosen so that direct comparisons of their effect on each model can be made. For the initial mass, it is assumed that sufficient inspection data of manufactured missiles is available so that a probability distribution could be computed. Suppose that, after constructing either a histogram or an EDF of the measurement data, it was found that a log-normal distribution with precisely known mean and standard deviation could be used (Bury, 1999; Krishnamoorthy, 2006).

For the thrust of the solid rocket motor, suppose that a number of newly manufactured motors have been fired so that variability in thrust can be well represented by a two-parameter Gamma distribution (Bury, 1999; Krishnamoorthy, 2006). It is well known that the propulsion characteristics can vary substantially with the age of the solid propellant. Suppose that a number of motors with various ages have also been fired. For each age grouping of motors, it is found that a Gamma distribution can be used, but each group has a different set of distribution parameters. As a result, the uncertainty in thrust characteristics could be represented as a mixture of aleatory and epistemic uncertainty. The aleatory portion of the uncertainty is due to manufacturing variability of the motor and the epistemic uncertainty is due to the age of the motor. The representation of the thrust characteristics

is given by a two-parameter gamma distribution, where the parameters of the distribution are given as interval-valued quantities. In the flight dynamics simulation, it is clear that the epistemic uncertainty in thrust can be reduced if information is added concerning the age of the motors of interest. For example, if all of the missiles that may be launched are known to be from a single production lot, then the epistemic uncertainty could be eliminated because it would be known when the production lot was manufactured. The two parameters of the gamma distribution would then become precisely known values.

3.5.3 Discretization and algorithm selection phase

The discretization method chosen to solve the ODEs of both mathematical models was a Runge-Kutta 4(5) method (Press *et al.*, 2007). The RK method is fifth-order accurate at each time step, and the integrator coefficients of Cash and Karp (1990) were used. The method provides an estimate of the local truncation error, i.e., truncation error at each step, so that adjusting the step size as the solution progresses can directly control the estimated numerical solution error. The local truncation error is computed by comparing the fourth-order accurate solution with the fifth-order accurate solution.

The method chosen for propagation of the uncertainties through the model was probability bounds analysis (PBA). As previously discussed in the mass–spring–damper example, a sampling procedure was used in which the aleatory and epistemic uncertainties are separated during the sampling. The particular sampling procedure used was Latin Hypercube Sampling (LHS) (Ross, 2006; Dimov, 2008; Rubinstein and Kroese, 2008). LHS employs stratified random sampling for choosing discrete values from the probability distribution specified for the aleatory uncertainties. For propagation of the epistemic uncertainty, samples are chosen from the two parameters of the gamma distribution characterizing the uncertainty in thrust of the solid rocket motor. Samples chosen over the two intervals are assigned a probability of unity. The method of obtaining samples over the two intervals can, in principle, be any method that obtains samples over the entire interval. The usual procedure used is to assign a uniform probability distribution over the interval and then use the same sampling procedure that is used for the aleatory uncertainties. It should be noted that the seeds for sampling the two interval-valued parameters were assigned different values so that there is no correlation between the random draws from each interval. The experimental design calls for performing the same number of LHS calculations for both the 3-DOF and 6-DOF models. An alternative procedure commonly used in complex analyses is to include a method to mix computer runs between the two models to maximize the accuracy and efficiency of the computations.

3.5.4 Computer programming phase

A computer code (TAOS) developed at Sandia National Laboratories was used to compute the trajectories of the missile flight example (Salguero, 1999). This general-purpose flight dynamics code has been used for a wide variety of guidance, control, navigation, and

optimization problems for flight vehicles. We used only the ballistic flight option to solve both the 3-DOF and 6-DOF equations of motion.

3.5.5 Numerical solution phase

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 model there are six 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, then the step size is increased for the next time step.

The LHS method often provides an advantage in sampling convergence rate over traditional Monte Carlo sampling. However, that advantage is somewhat degraded because estimates of sampling error cannot be computed without replicating the LHS runs.

3.5.6 Solution representation phase

For this relatively simple example, the representation of solution results is rather straightforward. The primary SRQ of interest for the example was the range of the missile. The most common method of showing nondeterministic results is to plot the CDF for the SRQ of interest. If only aleatory uncertainties exist in a nondeterministic analysis, then only one CDF exists for any given SRQ. If epistemic uncertainty also exists, as it does in this simulation, then an ensemble of CDFs must be computed. One CDF results from *each sample* of all of the epistemic uncertainties. To compute the p-box of the SRQ, one determines the minimum and maximum probability from all of the CDFs that were computed at each value of the SRQ. If alternative mathematical models are used, as in the present case, then a p-box is shown for each model.

Figure 3.22 shows a representative result for the CDF of the range of the missile given as a p-box resulting from one of the mathematical models. The p-box shows that epistemic uncertainty due to the age of the solid propellant rocket motor is a major contributor to the uncertainty in the range of the missile. For example, at the median of the range (probability = 0.5), the range can vary by about 1 km depending on the age of the motor. A different way of interpreting the p-box is to pick a value of range, and then read the interval-valued probability. For example, the probability of attaining a range of 34 km, or less, can vary from 0.12 to 0.52, depending on the age of the motor.

Recall that the Gamma distribution represents the variability in thrust due to manufacturing processes and the epistemic uncertainty due to the age of the motor is represented by the two interval-valued parameters in the distribution. Some analysts would argue that

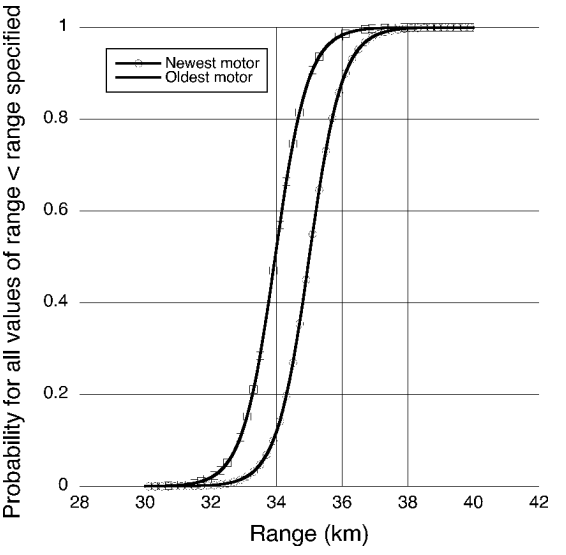

Figure 3.22 Representative p-box for range of the missile as a function of rocket motor age.

an alternative method of representing the uncertainty due to age is to replace the characterization of the parameters with two uniform PDFs over the same range of the intervals. They would argue that, if the age of the motors is uniformly distributed over time, then a uniform distribution should represent the age. The fallacy of this argument is that once a motor is selected for firing, the age of motor is fixed, but the variability of the thrust still exists, which is characterized by the gamma distribution. That is, once a motor is selected, the previously unknown age of the motor is now a number that identifies a single, precise gamma distribution. If this were done, then a *single* CDF would replace the p-box shown in Figure 3.22. If the uniform PDF approach were taken, the representation of the uncertainty in range would present a very different picture to the decision maker than what is shown in Figure 3.22. There would be *one* CDF that was within the p-box, disguising the true uncertainty in the range.

3.6 References

AEC (1966). *AEC Atomic Weapon Safety Program*. Memorandum No. 0560, Washington, DC, US Atomic Energy Commission.

Almond, R. G. (1995). *Graphical Belief Modeling*. 1st edn., London, Chapman & Hall.

Andrews, J. D. and T. R. Moss (2002). *Reliability and Risk Assessment*. 2nd edn., New York, NY, ASME Press.

Ang, A. H.-S. and W. H. Tang (2007). *Probability Concepts in Engineering: Emphasis on Applications to Civil and Environmental Engineering*. 2nd edn., New York, John Wiley.

Aughenbaugh, J. M. and C. J. J. Paredis (2006). The value of using imprecise probabilities in engineering design. *Journal of Mechanical Design*. **128**, 969–979.

Aven, T. (2005). *Foundations of Risk Analysis: a Knowledge and Decision-Oriented Perspective*, New York, John Wiley.
Ayyub, B. M. (1994). The nature of uncertainty in structural engineering. In *Uncertainty Modelling and Analysis: Theory and Applications*. B. M. Ayyub and M. M. Gupta, eds. New York, Elsevier: 195–210.
Ayyub, B. M., ed. (1998). *Uncertainty Modeling and Analysis in Civil Engineering*. Boca Raton, FL, CRC Press.
Ayyub, B. M. and G. J. Klir (2006). *Uncertainty Modeling and Analysis in Engineering and the Sciences*, Boca Raton, FL, Chapman & Hall.
Bae, H.-R., R. V. Grandhi, and R. A. Canfield (2006). Sensitivity analysis of structural response uncertainty propagation using evidence theory. *Structural and Multidisciplinary Optimization*. **31**(4), 270–279.
Bardossy, G. and J. Fodor (2004). *Evaluation of Uncertainties and Risks in Geology: New Mathematical Approaches for their Handling*, Berlin, Springer-Verlag.
Baudrit, C. and D. Dubois (2006). Practical representations of incomplete probabilistic knowledge. *Computational Statistics and Data Analysis*. **51**, 86–108.
Beck, M. B. (1987). Water quality modeling: a review of the analysis of uncertainty. *Water Resources Research*. **23**(8), 1393–1442.
Bedford, T. and R. Cooke (2001). *Probabilistic Risk Analysis: Foundations and Methods*, Cambridge, UK, Cambridge University Press.
Ben-Haim, Y. (1999). Robust reliability of structures with severely uncertain loads. *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, AIAA Paper 99-1605*, St. Louis, MO, American Institute of Aeronautics and Astronautics, 3035–3039.
Bogen, K. T. and R. C. Spear (1987). Integrating uncertainty and interindividual variability in environmental risk assessment. *Risk Analysis*. **7**(4), 427–436.
Bossel, H. (1994). *Modeling and Simulation*. 1st edn., Wellesley, MA, A. K. Peters.
Box, G. E. P. (1980). Sampling and Bayes' inference in scientific modeling and robustness. *Journal of the Royal Statistical Society: Series A*. **143**(A), 383–430.
Box, G. E. P., J. S. Hunter, and W. G. Hunter (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. 2nd edn., New York, John Wiley.
Breeding, R. J., J. C. Helton, E. D. Gorham, and F. T. Harper (1992). Summary description of the methods used in the probabilistic risk assessments for NUREG-1150. *Nuclear Engineering and Design*. **135**, 1–27.
Bury, K. (1999). *Statistical Distributions in Engineering*, Cambridge, UK, Cambridge University Press.
Cacuci, D. G. (2003). *Sensitivity and Uncertainty Analysis: Theory*, Boca Raton, FL, Chapman & Hall/CRC.
Cash, J. R. and A. H. Karp (1990). A variable order Runge-Kutta method for initial-value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software*. **16**(3), 201–222.
Choi, S.-K., R. V. Grandhi, and R. A. Canfield (2007). *Reliability-based Structural Design*, London, Springer-Verlag.
Cullen, A. C. and H. C. Frey (1999). *Probabilistic Techniques in Exposure Assessment: a Handbook for Dealing with Variability and Uncertainty in Models and Inputs*, New York, Plenum Press.
Dimov, I. T. (2008). *Monte Carlo Methods for Applied Scientists*. 2nd edn., Singapore, World Scientific Publishing.

Dorner, D. (1989). *The Logic of Failure, Recognizing and Avoiding Error in Complex Situations*, Cambridge, MA, Perseus Books.

Fellin, W., H. Lessmann, M. Oberguggenberger, and R. Vieider, eds. (2005). *Analyzing Uncertainty in Civil Engineering*. New York, Springer.

Ferson, S. (1996). What Monte Carlo methods cannot do. *Human and Ecological Risk Assessment*. **2**(4), 990–1007.

Ferson, S. (2002). *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*. Setauket, NY, Applied Biomathematics Corp.

Ferson, S. and L. R. Ginzburg (1996). Different methods are needed to propagate ignorance and variability. *Reliability Engineering and System Safety*. **54**, 133–144.

Ferson, S. and J. G. Hajagos (2004). Arithmetic with uncertain numbers: rigorous and (often) best possible answers. *Reliability Engineering and System Safety*. **85**(1–3), 135–152.

Ferson, S., V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz (2003). *Constructing Probability Boxes and Dempster–Shafer Structures*. SAND2003-4015, Albuquerque, NM, Sandia National Laboratories.

Ferson, S., R. B. Nelsen, J. Hajagos, D. J. Berleant, J. Zhang, W. T. Tucker, L. R. Ginzburg, and W. L. Oberkampf (2004). *Dependence in Probabilistic Modeling, Dempster–Shafer Theory, and Probability Bounds Analysis*. SAND2004-3072, Albuquerque, NM, Sandia National Laboratories.

Fetz, T., M. Oberguggenberger, and S. Pittschmann (2000). Applications of possibility and evidence theory in civil engineering. *International Journal of Uncertainty*. **8**(3), 295–309.

Frank, M. V. (1999). Treatment of uncertainties in space: nuclear risk assessment with examples from Cassini Mission applications. *Reliability Engineering and System Safety*. **66**, 203–221.

Gehman, H. W., J. L. Barry, D. W. Deal, J. N. Hallock, K. W. Hess, G. S. Hubbard, J. M. Logsdon, D. D. Osheroff, S. K. Ride, R. E. Tetrault, S. A. Turcotte, S. B. Wallace, and S. E. Widnall (2003). *Columbia Accident Investigation Board Report Volume I*. Washington, DC, National Aeronautics and Space Administration Government Printing Office.

Ghanem, R. G. and P. D. Spanos (2003). *Stochastic Finite Elements: a Spectral Approach*. Revised edn., Mineda, NY, Dover Publications.

Haimes, Y. Y. (2009). *Risk Modeling, Assessment, and Management*. 3rd edn., New York, John Wiley.

Haldar, A. and S. Mahadevan (2000a). *Probability, Reliability, and Statistical Methods in Engineering Design*, New York, John Wiley.

Haldar, A. and S. Mahadevan (2000b). *Reliability Assessment Using Stochastic Finite Element Analysis*, New York, John Wiley.

Hauptmanns, U. and W. Werner (1991). *Engineering Risks Evaluation and Valuation*. 1st edn., Berlin, Springer-Verlag.

Helton, J. C. (1993). Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. *Reliability Engineering and System Safety*. **42**(2–3), 327–367.

Helton, J. C. (1994). Treatment of uncertainty in performance assessments for complex systems. *Risk Analysis*. **14**(4), 483–511.

Helton, J. C. (1997). Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. *Journal of Statistical Computation and Simulation*. **57**, 3–76.

Helton, J. C. (1999). Uncertainty and sensitivity analysis in performance assessment for the waste isolation pilot plant. *Computer Physics Communications*. **117**(1–2), 156–180.

Helton, J. C., D. R. Anderson, H.-N. Jow, M. G. Marietta, and G. Basabilvazo (1999). Performance assessment in support of the 1996 compliance certification application for the Waste Isolation Pilot Plant. *Risk Analysis*. **19**(5), 959–986.

Helton, J. C., J. D. Johnson, and W. L. Oberkampf (2004). An exploration of alternative approaches to the representation of uncertainty in model predictions. *Reliability Engineering and System Safety*. **85**(1–3), 39–71.

Helton, J. C., W. L. Oberkampf, and J. D. Johnson (2005). Competing failure risk analysis using evidence theory. *Risk Analysis*. **25**(4), 973–995.

Helton, J. C., J. D. Johnson, C. J. Sallaberry, and C. B. Storlie (2006). Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering and System Safety*. **91**(10–11), 1175–1209.

Hoffman, F. O. and J. S. Hammonds (1994). Propagation of uncertainty in risk assessments: the need to distinguish between uncertainty due to lack of knowledge and uncertainty due to variability. *Risk Analysis*. **14**(5), 707–712.

Hora, S. C. and R. L. Iman (1989). Expert opinion in risk analysis: the NUREG-1150 methodology. *Nuclear Science and Engineering*. **102**, 323–331.

Jacoby, S. L. S. and J. S. Kowalik (1980). *Mathematical Modeling with Computers*, Englewood Cliffs, NJ, Prentice-Hall.

Kaplan, S. and B. J. Garrick (1981). On the quantitative definition of risk. *Risk Analysis*. **1**(1), 11–27.

Kleijnen, J. P. C. (1998). Chapter 6: Experimental design for sensitivity analysis, optimization, and validation of simulation models. In *Handbook of Simulation: Principles, Methodology, Advances, Application, and Practice*. J. Banks, ed. New York, John Wiley: 173–223.

Klir, G. J. and M. J. Wierman (1998). *Uncertainty-Based Information: Elements of Generalized Information Theory*, Heidelberg, Physica-Verlag.

Kloeden, P. E. and E. Platen (2000). *Numerical Solution of Stochastic Differential Equations*, New York, Springer.

Kohlas, J. and P.-A. Monney (1995). *A Mathematical Theory of Hints – an Approach to the Dempster–Shafer Theory of Evidence*, Berlin, Springer-Verlag.

Krause, P. and D. Clark (1993). *Representing Uncertain Knowledge: an Artificial Intelligence Approach*, Dordrecht, The Netherlands, Kluwer Academic Publishers.

Kriegler, E. and H. Held (2005). Utilizing belief functions for the estimation of future climate change. *International Journal for Approximate Reasoning*. **39**, 185–209.

Krishnamoorthy, K. (2006). *Handbook of Statistical Distribution with Applications*, Boca Raton, FL, Chapman and Hall.

Kumamoto, H. (2007). *Satisfying Safety Goals by Probabilistic Risk Assessment*, Berlin, Springer-Verlag.

Kumamoto, H. and E. J. Henley (1996). *Probabilistic Risk Assessment and Management for Engineers and Scientists*. 2nd edn., New York, IEEE Press.

Law, A. M. (2006). *Simulation Modeling and Analysis*. 4th edn., New York, McGraw-Hill.

Lawson, D. (2005). *Engineering Disasters – Lessons to be Learned*, New York, ASME Press.

LeGore, T. (1990). Predictive software validation methodology for use with experiments having limited replicability. In *Benchmark Test Cases for Computational Fluid*

Dynamics. I. Celik and C. J. Freitas, eds. New York, American Society of Mechanical Engineers. FED-Vol. 93: 21–27.
Leijnse, A. and S. M. Hassanizadeh (1994). Model definition and model validation. *Advances in Water Resources*. **17**, 197–200.
Mason, R. L., R. F. Gunst, and J. L. Hess (2003). *Statistical Design and Analysis of Experiments, with Applications to Engineering and Science*. 2nd edn., Hoboken, NJ, Wiley Interscience.
Melchers, R. E. (1999). *Structural Reliability Analysis and Prediction*. 2nd edn., New York, John Wiley.
Modarres, M., M. Kaminskiy, and V. Krivtsov (1999). *Reliability Engineering and Risk Analysis; a Practical Guide*, Boca Raton, FL, CRC Press.
Moller, B. and M. Beer (2004). *Fuzz Randomness: Uncertainty in Civil Engineering and Computational Mechanics*, Berlin, Springer-Verlag.
Morgan, M. G. and M. Henrion (1990). *Uncertainty: a Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. 1st edn., Cambridge, UK, Cambridge University Press.
Mosey, D. (2006). *Reactor Accidents: Institutional Failure in the Nuclear Industry*. 2nd edn., Sidcup, Kent, UK, Nuclear Engineering International.
NASA (2002). *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners.* Washington, DC, NASA.
Neelamkavil, F. (1987). *Computer Simulation and Modelling*. 1st edn., New York, John Wiley.
Nikolaidis, E., D. M. Ghiocel, and S. Singhal, eds. (2005). *Engineering Design Reliability Handbook*. Boca Raton, FL, CRC Press.
NRC (1990). *Severe Accident Risks: an Assessment for Five U.S. Nuclear Power Plants*. NUREG-1150, Washington, DC, US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Division of Systems Research.
NRC (2009). *Guidance on the Treatment of Uncertainties Assoicated with PRAs in Risk-Informed Decision Making.* Washington, DC, Nuclear Regulator Commission.
Oberkampf, W. L. and J. C. Helton (2005). Evidence theory for engineering applications. In *Engineering Design Reliability Handbook*. E. Nikolaidis, D. M. Ghiocel and S. Singhal, eds. New York, NY, CRC Press: 29.
Oberkampf, W. L., S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin (2000). *Estimation of Total Uncertainty in Computational Simulation.* SAND2000-0824, Albuquerque, NM, Sandia National Laboratories.
Oberkampf, W. L., S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin (2002). Error and uncertainty in modeling and simulation. *Reliability Engineering and System Safety*. **75**(3), 333–357.
Oksendal, B. (2003). *Stochastic Differential Equations: an Introduction with Applications*. 6th edn., Berlin, Springer-Verlag.
Paté-Cornell, M. E. (1990). Organizational aspects of engineering system failures: the case of offshore platforms. *Science*. **250**, 1210–1217.
Pegden, C. D., R. E. Shannon, and R. P. Sadowski (1990). *Introduction to Simulation Using SIMAN*. 1st edn., New York, McGraw-Hill.
Petroski, H. (1994). *Design Paradigms: Case Histories of Error and Judgment in Engineering*, Cambridge, UK, Cambridge University Press.
Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (2007). *Numerical Recipes in FORTRAN*. 3rd edn., New York, Cambridge University Press.

Raczynski, S. (2006). *Modeling and Simulation: the Computer Science of Illusion*, New York, Wiley.
Reason, J. (1997). *Managing the Risks of Organizational Accidents*, Burlington, VT, Ashgate Publishing Limited.
Roache, P. J. (1998). *Verification and Validation in Computational Science and Engineering*, Albuquerque, NM, Hermosa Publishers.
Ross, S. M. (2006). *Simulation*. 4th edn., Burlington, MA, Academic Press.
Ross, T. J. (2004). *Fuzzy Logic with Engineering Applications*. 2nd edn., New York, Wiley.
Rubinstein, R. Y. and D. P. Kroese (2008). *Simulation and the Monte Carlo Method*. 2nd edn., Hoboken, NJ, John Wiley.
Salguero, D. E. (1999). *Trajectory Analysis and Optimization Software (TAOS)*. SAND99-0811, Albuquerque, NM, Sandia National Laboratories.
Saltelli, A., S. Tarantola, F. Campolongo, and M. Ratto (2004). *Sensitivity Analysis in Practice: a Guide to Assessing Scientific Models*, Chichester, England, John Wiley.
Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola (2008). *Global Sensitivity Analysis: the Primer*, Hoboken, NJ, Wiley.
Schrage, M. (1999). *Serious Play: How the World's Best Companies Simulate to Innovate*, Boston, MA, Harvard Business School Press.
Serrano, S. E. (2001). *Engineering Uncertainty and Risk Analysis: a Balanced Approach to Probability, Statistics, Stochastic Modeling, and Stochastic Differential Equations*, Lexington, KY, HydroScience Inc.
Severance, F. L. (2001). *System Modeling and Simulation: an Introduction*, New York, Wiley.
Singh, V. P., S. K. Jain, and A. Tyagi (2007). *Risk and Reliability Analysis: a Handbook for Civil and Environmental Engineers*, New York, American Society of Civil Engineers.
Singpurwalla, N. D. (2006). *Reliability and Risk: a Bayesian Perspective*, New York, NY, Wiley.
Stockman, C. T., J. W. Garner, J. C. Helton, J. D. Johnson, A. Shinta, and L. N. Smith (2000). Radionuclide transport in the vicinity of the repository and associated complementary cumulative distribution functions in the 1996 performance assessment for the Waste Isolation Pilot Plant. *Reliability Engineering and System Safety*. **69**(1–3), 369–396.
Storlie, C. B. and J. C. Helton (2008). Multiple predictor smoothing methods for sensitivity analysis: description of techniques. *Reliability Engineering and System Safety*. **93**(1), 28–54.
Suter, G. W. (2007). *Ecological Risk Assessment*. 2nd edn., Boca Raton, FL, CRC Press.
Taylor, H. M. and S. Karlin (1998). *An Introduction to Stochastic Modeling*. 3rd edn., Boston, Academic Press.
Tenner, E. (1996). *Why Things Bite Back*, New York, Alfred A. Knopf.
Tung, Y.-K. and B.-C. Yen (2005). *Hydrosystems Engineering Uncertainty Analysis*, New York, McGraw-Hill.
Vaughan, D. (1996). *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA*, Chicago, IL, The University of Chicago Press.
Vinnem, J. E. (2007). *Offshore Risk Assessment: Principles, Modelling and Applications of QRA Studies*, Berlin, Springer-Verlag.
Vose, D. (2008). *Risk Analysis: a Quantitative Guide*. 3rd edn., New York, Wiley.

Yee, H. C. and P. K. Sweby (1995). Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations II. Global asymptotic behavior of time discretizations. *Computational Fluid Dynamics*. **4**, 219–283.

Yee, H. C. and P. K. Sweby (1996). *Nonlinear Dynamics & Numerical Uncertainties in CFD.* Rept. No. 110398, Moffett Field, CA, NASA/Ames Research Center.

Yee, H. C. and P. K. Sweby (1998). Aspects of numerical uncertainties in time marching to steady-state numerical solutions. *AIAA Journal*. **36**(5), 712–724.

Yee, H. C., J. R. Torczynski, S. A. Morton, M. R. Visbal, and P. K. Sweby (1997). On spurious behavior of CFD simulations. *13th AIAA Computational Fluid Dynamics Conference, AIAA Paper 97-1869*, Snowmass, CO, American Institute of Aeronautics and Astronautics.

Zeigler, B. P., H. Praehofer, and T. G. Kim (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. 2nd edn., San Diego, CA, Academic Press.