Since I was a child I was enthralled with the idea of using a computer to solve problems that I couldn't with pencil and paper. There is a good chance if you are reading this that you have had a similar experience with the augmented problem solving ability that computers allow. Most people in computational science have, at one point or another, been frustrated with the limited applicability of the typical toolbox used to solve partial differential equations (e.g., integral transforms, eigenvector expansions, etc.). The beauty of computer simulation is that any problem can be solved provided you can cast the continuum equations in terms of finite quantities and you have enough computer horsepower at your disposal.

Beyond the fact that computation allows the solution of problems that are intractable by other means, simulation also allows us to probe areas that experimental measurements cannot. No experiment could give you the temperature profile at every point on the surface of a space re entry vehicles or the distribution of neutrons in a nuclear reactor. Solving the equations on a computer gives you this information at the scale one desires, and can give insight into the mechanism of a phenomenon in ways that experiments can only suggest. Indeed, if the goal of the simulation is to show whether an experiment will be in a particular regime, it is likely that a simulation-based prediction that is within an order of magnitude of an actual values.

The ability to show what is going on in an experiment can be extrapolated. It is reasonable to suggest, and difficult to perform, that a computational simulation could tell a researcher what will happen in an experiment that has yet to be performed. Such a request occurs often in terms of design, asking how a new system will perform when it is built. Typically, this exercise uses computation to rule out certain designs, and the candidate designs that pass the computational pass are then tested in small scale experiments, before production of the new system takes place. Eliminating some designs will cut down on the possible number of prototypes that need to be built and tested, leading to significant cost and time savings. Using computation in this way is entirely justifiable and reasonable, especially when there is operation history and previous experimental results for systems that are "nearby" the new candidate design. Take the example of an airplane. From my (admittedly) outsider's perspective, the jets produced in the 2000's are not significantly different than those produced in the 1980's in terms of basic aeronautics. A new aircraft design probably shares a lot in common with previous designs and if computational models could adequately predict what happens with those systems, then there is hope that the new designs could have their performance adequately simulated. Of course, this begs the questions: "What does it mean to have a nearby design" and "How do we quantify adequate simulation performance?" We will revisit these topics later.

If simulation for evolutionary design makes sense, provided we define our goal clearly, can we go one step further: can we predict the behavior of a system in conditions where we cannot do a full scale experiment either for cost, safety, or regulatory reasons. These are the questions that are often the thorniest to answer and have the highest impact. When the space shuttle was damaged by falling debris on a launch, how can we use computation to make statements about the reliability of the craft? The people making decisions about the mission want an answer, but

we also need to quantify the our uncertainty in our answer. Another case worth mentioning is the question of long term reliability of a system. Consider a nuclear reactor that was initially designed to last 30 years of continuous operation. What can we say about the safety of the system if its license is extended another 50 years? We obviously cannot do an experiment where a reactor system undergoes 80 years of irradiation at operating conditions without actually operating the system for that long (and even if we did, that would be one sample from the distribution of possible outcomes). We can do small scale experiments where certain components receive an equivalent dose to decades of radiation exposure, but how can we assemble the experimental data to say that the entire system is safe. How do we state our sense of the risks/uncertainty in any result.

Both of these scenarios are high-impact decisions that must be made without full knowledge of how the system will respond. Lives could be potentially on the line, and we need to make the best decision given the experimental, computational, and theoretical data at hand. We could be "safe" and always answer the question in the negative. In such a case almost no new technologies would be fielded and arguably life as a whole would suffer. To a large extent economic growth depends on the development of technology, this has been true since the advent of agriculture; to stop the progress of technology in the midst of the tools at our disposal would almost be criminal. A person earning a subsistence wage in a first-world country today has a life that would be envied by the monarchs even two centuries in the past (if only for antibiotics alone).

I contend that having the default "safe" option is not an option at all. I imagine you feel the same way: you have most likely made the decision to travel somewhere by automobile. It is not possible to guarantee that such a trip was safe, but you, perhaps unconsciously, decided the risks were worth the reward. The best answer is to balance the uncertainty in the outcome with the benefit of the risk.

This work *does not* deal with the process of making decisions under uncertainty. That is, we will not deal with policy matters of what is an acceptable risk in a situation. What we will discuss is how to assess the amount of uncertainty in a prediction based on a simulation. We will call the process of making a credible prediction based on computer simulation and available experimental data **predictive science**.

1.1 The limits of prediction

Before embarking on a journey to make predictions with simulations, we should a look at what are the theoretical limits of how we make predictions and the path that scientific progress has taken to get to the current state of affairs where we believe we can make predictions with an understanding of the uncertainty in those predictions.

The height of predictability, i.e., determinism, can be expressed through a thought experiment. In 1814 Pierre Simon Laplace proposed his "demon"

- If it knew the position and momentum of every atom in the universe,

- It could, using Newton's laws, determine the future state of the universe.
- Obviously, quantum mechanics makes the power of the demon impossible.
- Even in the classical sense, its task is impossible.
- Recent proof [Collins(2009), Wolpert(2008)].
- Of course this is a rather strong form of determinism.

The early 20th century was in a sense the zenith of the confidence in Enlightenment. Humanity believed that the solution to the problems of nature were at hand:

- Hilbert's problems and program
- Classical physics
- Russell and Whitehead's "Principia Mathematica" tried to derive mathematical truth from a set of basic axioms.
  - If you want to know for sure that $1 + 1 = 2$, they prove it on page 379.
  - With the added comment "The above proposition is occasionally useful."

  That was soon to change.
- Heisenberg uncertainty principle
  - By being a part of the system you change it
  - Related to the Nyquist-Shannon sampling theorem.
- Gödel's incompleteness theorem(s)
  - A complete and consistent set of axioms for all of mathematics is impossible
  - Liar's paradox: Gödel proved that basic arithmetic (as defined by a finite set of axioms) will lead to the statement "True proposition G cannot be proved"
- Church-Turing Theorem - The Entscheidungsproblem
  - If I give you a statement about mathematics, it is not possible to guarantee that you can prove or disprove that statement in a finite number of steps.

The confidence in science and progress was set back in the West by the World Wars. Nearly concurrently, the work was done that showed we cannot know everything: Gödel, Turing, Quantum mechanics, etc. Of course all is not lost though:

- Bridges get built and they (mostly) work
- Technology has marched on.

Certainly we as scientists rely on some, perhaps ineffable, weak form of determinism.

1.1.1 Simulation v. Experiment

For the most part computer simulations are guilty until proven innocent, in that the burden of proving that a simulation does represent reality lies in the hands of the one doing the simulation. On the other hand, an experimental result is often widely

accepted as being an accurate picture of reality. Few will question whether the team who completed the experiment properly characterized and accounted for all sources of error. Parphrasing Roache [Roache(1998)], that state of play is such that nobody believes the result of a simulation, except the person who performed the simulation, and everbody believes the result of an experiment, except the person who ran the experiment.

The outlook of the experimenter is often proper, that is, it is naive to assume that the result of a single experiment is the final word on a specific phenomenon or system. This should also be the outlook of the computational scientist that is attempting to validate a particular model: one number from one experiment should not make or break a model.

1.2 Verification and Validation

Verification and validation (V&V) are two processes in computational science that give confidence in the results of a simulation. Successful V&V are essential to performing uncertainty quantification. Approaches to performing V&V are the subject of several books. Three useful examples are [Roache(1998), Oberkampf and Roy(2010), Knupp and Salari(2002)].

1.2.1 Code and Solution Verification

Interrelated with V&V

Verification is the process of demonstrating that a simulation code solves the underlying mathematical model equations and the characterization of the numerical error. In simpler terms, verification answers the questions:

- Does the code solve the equations it claims to,
- How big is the error, and
- How do I expect that error to change as the mesh, time step, etc. changes?

The verification exercise is often a computer science and mathematics exercise. The computer science aspect is evident in the fact that errors, i.e., bugs, in the code slightly alter the solutions and the numerical error in the simulation. In this regard, a code bug code make the code solve a different set of equations than intended. The mathematics aspect of verification is showing that the code has numerical error that behaves as expected, base on the knowledge of the accuracy, stability, and other properties of the underlying numerical methods. To demonstrate that the error behaves as expected, one often compares to the code to exact solutions to the underlying equations, and shows that the error in the calculation goes to zero in an expect way as the resolution of the simulation is increased.

Also included in the verification process is the exercise known as solution verification. Solution verification attempts to bound and perhaps quantify the numerical error in a calculation. One might imagine that when simulating a large system, calculations may only be done at a handful of resolutions (due to, perhaps, the difficulties of mesh generation or the amount of available time on a machine). Quantifying the error in such a situation may be difficult. In these instances one may turn to convergence acceleration techniques, or other estimates such as Richardson extrapolation or single-grid error estimators. Solution verification is important component of studying uncertainty because the numerical error in a calculation is a source of uncertainty. One needs to know the magnitude of this error to account for its impact.

1.2.2 Validation

Validation attempts to answer the question of whether the underlying mathematical model is appropriate for the system of interest. In the parlance of our times, validation can be said to answer the question: "Am I solving the correct equations?" Validation is thought to be an endeavor in physics and engineering. This is due to the fact that to perform validation one needs to compare numerical solutions to experiments, and, if the system of interest does not have experimental data, expert judgment to decide if the experimental data available is applicable. Beyond the physics/engineering questions, there is an element of the philosophy of science to answer the question of whether the mathematical model is predictive. Validation is necessarily situation dependent: a code that is valid for one system will be not valid for another. That being said there is no such thing as a "validated" code, because one can always come up with a situation where the mathematical model will fail[1]. The best that validation can do is make concrete and narrow statements about the applicability of a mathematical model for a system under a particular circumstance. The range of scenarios where a model has been shown to be valid is its domain of validity.

Here is where the opt-quoted aphorism by George Box, "All models are wrong, but some are useful," could be mentioned. Validation answers the question of where a given mathematical model, that is a simplification of reality, is useful describing physical phenomena.

In bears mentioning that one cannot perform validation without having done thorough verification of the code because it is not possible to make statements about a code's validity unless we know something about the numerical error. This also connects validation to uncertainty quantification because we cannot measure the agreement between simulation and experiment without knowledge of how uncertain the simulation result is.

[1] This is not just a handwaving argument. There is no unified theory of all the forces in the universe, i.e., we have not uncovered the equations that underly the universe at all scales. Therefore, any single mathematical model will not be accurate for every problem.

Where verification is math, validation, it can be said, is a scientific endeavor. We have a theory that a particular model or system of equations can explain the phenomena of a real world situation; proving that the theory is applicable is by no means trivial. Also, because validation answers a scientific question, the methodology of validation differs significantly between the scientific branches. The contrasts with the fact that mathematics is a generic construct, a property that makes the process of verification the same across disciplines: only the equations change. In a validation exercise one needs to leverage knowledge of the underlying scientific branch, be it physics, engineering, chemistry, biology, economics or sociology.

Even a fledgling science student knows that lynchpin of the scientific process is the use of experiments to confirm hypotheses. Confirming the theory that a given mathematical model describes a phenomena is no different. The problem is the process of comparing experiments to numerical results is not as simple as computing a number and then seeing if agrees with the experimental measurement.

Unfortunately, experimental data is often lacking or impossible to gather. This predicament is not uncommon. The problem of geologic disposal of nuclear waste is a prime example. We can model the behavior of a repository for nuclear waste using geology, hydrology, and nuclear engineering considerations in an attempt to say whether the waste will contaminate the groundwater, but we cannot do an experiment unless we want to wait 10,000 years (!) for the result of the experiment. In such a situation often the best we can do is to forthrightly state the assumptions in our model and point by point justifying each of these assumptions.

1.2.3 Experiments for Validation

Using experimental measurements to compare with simulation results is the bedrock of model validation. Nevertheless, comparing numerical results with experiment can be exceedingly difficult. Specifying the problem for the numerical code to solve is not a straightforward task. For example, the boundary and initial conditions that are needed to mock up the experimental set up may not be known to enough precision, or may not fit into the framework of the code. Care must be taken and a large amount of detail is needed to simulate a given experiment.

The large amount of detail needed to properly specify the experimental configuration in order to simulate the experiment on a computer will often mean that "old" experiments are not suitable for the validation task. Generally, there is not enough detail archived about the experiments, especially in journal publications where economy of space is favored over detailed descriptions of the experiment and long lists of data. It is best to use experiments explicitly designed to validate computational models. That way care can be taken to precisely characterize the experimental set up, provide large amounts of data, provide detailed estimates of the measurement error.

Another fact that should be considered when thinking about experiments is the fact that experiments rarely report raw data. Rather, experiments often use some

conceptual model to process the raw data and produce a result. For example, consider an experiment measures the yield stress of a novel material. That experiment assumes that the materials properties are such that there is one particular value of the yield stress. Also, the yield stress is not measured; other parameters of the measured and then a yield stress is inferred.

By far the most important consideration in interpreting experimental data is the notion of measurement error. There are two types of errors in experiments, systematic error, also known as bias, and random errors, also known as precision errors. Random errors contribute to the scatter in the data of repeated measurements. Bias or systematic error is basically all the other error in the problem. For example, errors are often characterized as being a Gaussian distribution with a mean and variance. In this

1.2.4 Small-Scale Experiments

The small-scale experiments mentioned above test the simulation performance on a particular aspect of the system simulation. For example, if the system under consideration couples heat transfer, fluid flow flow, and chemical reactions, there are several small-scale experiments possible. One type is a single-physics experiment, where a particular physics phenomena is observed and measured in isolation, for example, one could field a heat-transfer experiment and a separate fluid-flow experiment. Then the simulation for that single "physics", is compared with the experiment. If each of the single-physics experiments can be reproduced with the simulation code, we at least have the hope that the simulation will perform in the coupled case.

The other type of small-scale experiments involve a system similar to the full system but modified to make the experiment possible. For instance the system could be made smaller, the heating rate could be lower, the materials used could be surrogates for the actual materials. These experiments test the simulation's ability to reproduce phenomena in a coupled system as similar to the full system as possible. One example of a small scale experiment involves the heat transfer in a nuclear reactor. In a typical reactor each fuel assembly could be generating megawatts of power. A small-scale experiment may involve a fuel assembly containing non-nuclear material (a surrogate material) and heated electrically with kilowatts of energy (a scaled-down system load). These choices are made for several reasons, building the experiment with nuclear material like uranium would require much more effort in terms of safety, regulatory approval, and might limit the types of diagnostics available. The lower power level is required because megawatts of electricity is difficult to get except at specialized facilities.

1.3 What is Uncertainty Quantification

Uncertainty quantification (UQ) attempts to answer the question of how uncertain is the result of the computation [National Academy of Science(2012)]. Every simulation has inherent uncertainties in the input. These could be the dimensions of pieces of the system due to manufacturing tolerances, the constitutive properties of materials, or a lack of knowledge of the ambient conditions. Propagating these uncertainties to the result of the simulation is one aspect of UQ. Beyond such propagation of input, also called parametric, uncertainties, UQ attempts to include knowledge of numerical error (perhaps from solution verification) and the mathematical model error. UQ is often considered an exercise in statistics because of the probabilistic nature of uncertainty and the typically large number of uncertainties in a simulation. It should not be a purely statistical exercise, however, because the results of statistical models should respect the physics of the problem. Including physical knowledge in the statistics of the problem provides better estimates of uncertainty due to the fact that the physics considerations can constrain the distribution of quantities of interest.

UQ is not just a single step process, there are several stages that must be completed to defensibly, reliably, and accurately quantify the uncertainty in the simulation of a physical system. Each step could be the subject of a book on its own and is the subject of active research. The steps of uncertainty quantification are

1. Identifying quantities of interest,
2. Identifying and modeling the uncertainties in the problem inputs,
3. A down-selection of the uncertain inputs,
4. The propagation of uncertain inputs through the simulation, and
5. Determining how the uncertainties will affect predictions.

The first step in UQ, though often not thought of as a difficult task, is the selection of the quantities of interest (QoIs), that is, what are the metrics by which we will assess a given system or design. Typically, a QoI is a single scalar number, such as, for example, the maximum temperature in the system, the failure stress of the structure, etc. These quantities are typically expressed in terms of a function of the solution of a set of model equations (e.g., partial differential equations, algebraic relations, etc.). For example, the maximum temperature in the system could be determined by applying a maximum function to the solution of the heat equation. Another common situation requires taking an integral over time and/or space of the solution of the model equations. This might be the case if the QoI was the average of the temperature in the system during a certain time.

To proceed with uncertainty quantification, we need to be able to make statements about the input uncertainties. We cannot simply say that an input x is uncertain, we need to say how it is uncertain. Can we give it a probability distribution? Is it correlated with another input? Do we know basic statistics of the input, e.g. mean and variance? Answering these questions can be difficult. For instance, if we have 1000 observations of an input parameter x that all fall in the range $[a,b]$, does that

mean x can never take a value outside of this range? The answer to this question will have an impact on the resulting uncertainty in the QoI.

Uncovering and identifying the uncertainties in a simulation often asks questions that have not been asked before. Some of the data used in simulations is of unknown origin or based on approximate models. In these cases it may be difficult to characterize the uncertainty. Additionally, question about numerical error and model error, can also have an impact on a simulation, and should be considered.

The task of identifying uncertainties will often uncover many uncertain parameters in the simulation. Typically, as there are more uncertain parameters, more simulation results are required to quantify the uncertainty due to each parameter. Furthermore, in situations where the simulations are expensive in terms of computer time, one cannot afford to run a large number of simulations to characterize the impact of all of the uncertain parameters. In this scenario it can be useful to judiciously remove some of the uncertain parameters if they will have a small impact on the QoIs. This can be done by estimating local sensitivities and other approximations, such as active subspace projection. Of course, the selection of parameters to remove from the simulation is a source of uncertainty in the uncertainty quantification.

Given the uncertain parameters that one wants to analyze, the next step is to actually quantify the uncertainty in the QoIs from them. There are several approaches to do this that span the spectrum from simple methods that are quick and approximate or slow, but bulletproof methods. The choice will often depend on what the analysis will be used for. If one wants to judge whether a standard safety factor is applicable to a system, reliability methods can quickly assess how close a system is to "failure". If the distribution and extreme values of the QoI are important, for instance if there a potentially low-frequency but high-impact events possible in the system as in, for example, nuclear reactor safety, then more sophisticated methods may be required, such as polynomial chaos or Monte Carlo methods.

At this point, many would consider the uncertainty quantification process complete, yet the application of the knowledge learned about the system is an important consideration. Given that one knows how the QoI can vary with respect to inputs, what does that mean for making predictions? Additionally, we need to re-evaluate past experimental data in light of the uncertainties, as well as determine how likely the simulations are to be accurate for a prediction on a different system. Addressing these issues requires deeper understanding of the simulations and the input uncertainties. To answer these questions, when there is a limited budget of time or other resources for performing more simulations, it is often necessary to construct and approximation to the simulation, known as an emulator or a reduced order model. Furthermore, the types of input uncertainties affect the interpretation of the prediction.

The process and science of UQ is more than just putting error bars on the simulation. It requires inquisitiveness to ask questions about the impact of results, physical and engineering intuition to know how to interpret results, and the humility to understand that not all questions can be answered with certainty.

1.4 Selecting Quantities of Interest (QoIs)

As mentioned above, one of the necessary steps in the UQ process is selecting quantities of interest. These QoIs are necessary to perform many of the the subsequent UQ tasks. The fact that the QoIs are a finite number of scalar values may be counterintuitive to many computational scientists. One of the benefits of computational simulation is that we can usually get the solution "everywhere" in the problem. That is, we can know the solutions to the underlying model equations throughout the problem domain in space, times, etc. Nevertheless, the problem of discussing the uncertainty of a function, or in more technical terms a distribution of functions, is a much more difficult proposition. In a sense, the uncertainty in a function is the same as having an infinite number of QoIs. As we shall see, handling a small number of QoIs is difficult enough.

To illustrate the selection of a QoI, we will introduce a model problem that we will use throughout our study. This is the advection-diffusion-reaction equation. Here we are interested in a quantity $u(\mathbf{r},t)$ that is governing by the following partial differential equation on the spatial domain V

$$\frac{\partial u}{\partial t} + \mathbf{v}\cdot\nabla u = -\nabla\cdot\omega\nabla u + R(u), \qquad \mathbf{r}\in V,\ t>0. \tag{1.1}$$

with boundary and initial conditions given by

$$u(\mathbf{r},t) = g(\mathbf{r},t) \quad \mathbf{r}\in\partial V, \qquad u(\mathbf{r},0) = f(\mathbf{r}). \tag{1.2}$$

In this model, $\mathbf{v}$ is the speed of advection in each direction of u, ω is a diffusion coefficient, and $R(u)$ is a reaction function. We choose this model because it is a simplified model for many physical processes. For instance, if u is a temperature and $R(u)=0$, then we have a heat equation that includes convection via the $\mathbf{v}\cdot\nabla$ term, and heat conduction via the diffusion term. Other possible ways to use this model is to treat simplified problems in particle transport, contaminate dispersion, or fluid-flow problems.

In the situation where Eq. (1.1) is an adequate model for our physical system we might be interested in the following quantities:

- The maximum value of inside a given time range $[a,b]$:

$$\max_{\mathbf{r},t\in[a,b]} u(\mathbf{r},t),$$

- The average value over a particular region of space, D, and time range $[a,b]$:

$$\frac{1}{b-a}\frac{1}{|D|}\int_D d\mathbf{r}\int_a^b dt\, u(\mathbf{r},t).$$

  Here, $|D|$ is the volume of the region D.

- The total reaction rate in the system over a given time range $[a,b]$:

$$\int_V d\mathbf{r} \int_a^b dt\, R(u(\mathbf{r},t)).$$

- The outflow of from the system over a given time range $[a,b]$:

$$\int_V dA \int_a^b dt\, (\mathbf{v}\cdot\nabla - \omega\nabla)\, u(\mathbf{r},t).$$

These examples can be readily generalized for many problems and scenarios. As a generic way of writing a QoI, it is often possible to write a QoI, Q, as

$$Q = s(u) + \int_V d\mathbf{r} \int_0^T dt\, w(u,\mathbf{r},t). \tag{1.3}$$

Here, $s(u)$ is a function that maps the output of $u(\mathbf{r},t)$ to a scalar, such as the max function we saw above, and $w(u,\mathbf{r},t)$ is a weight function. As an example, if the QoI is the reaction rate over a range of time then $s(u) = 0$ and

$$w(u,\mathbf{r},t) = \begin{cases} R(u) & t \in [a,b] \\ 0 & \text{otherwise} \end{cases}.$$

Alternatively, we were interested in the maximum value of u, then $s(u)$ would be a a maximum function.

The upshot of this discussion is that QoIs can be very general and can typically written in the form given in Eq. (1.3). If the domain of u includes more than just space and time, then the integrals will include those added dimensions.

1.5 Identifying Uncertainties

Finding all the uncertainties in a computer simulation is a harder task than it might seem at first blush. For a large, simulation code there is typically an input file that controls how the code with many possible options and all of those options could be a source of uncertainty. Additionally, there are sources of uncertainty that could arise from what the code is doing behind the scenes. In this section we will give a list of the potential sources of uncertainty, how to classify them, and what they mean.

1.5.1 Types of uncertainties

There are two main classes of uncertainties in a problem. These are not necessarily two distinct classes as some uncertainties could be classified into either category.

1.5.2 Aleatory Uncertainties

1. These are uncertainties due to inherent randomness.
2. Comes from the latin for dice player: *aleator*.
3. Examples include manufacturing tolerances, ambient conditions (i.e., weather), the distribution of constituents in the mixture, measurement error.
4. Aleatory uncertainties have a distribution or you can infer one based on the results.

1.5.3 Epistemic Uncertainties

1. These are uncertainties due to lack of knowledge
2. Can be due to an approximate model, numerical error, unknown unknowns.
3. Not distributional, uncertainties are interval quantities, rather than probabilities.
4. Many times these are parameters that have no "correct value".
5. Affect how we interpret results.

As an example, consider a car braking system where there is a "0.1% chance of failure." The implications of this 0.1% number depend on the type of uncertainties in the estimate. In one case, the uncertainties are due to aleatory uncertainy: the system performance is determined by variability in manufactured parts. The analysis indicates that 0.1% of manufactured parts made will fall outside the tolerable range and will fail due to inherent uncertainties in the manufacturing process. The result is the 0.1% of the systems will fail.

If, however, there are uncertainties in the failure temperature in the brake system. Based on the possible range of the failure temperature, about 0.1% of that range will lead to the system failure. What this means is that if the failure temperature for the system is in the 0.1% range, then all the brakes will fail. In other words, the 0.1% chance of failure, means that there is a 0.1% chance that all the brakes fail.

As this example makes clear, the types of uncertainties affect the interpretation of the output. Also, because epistemic uncertainties do not have an associated probability distribution, there are fewer mathematical tools to deal with the lack of knowledge. We will show, however, how we can take them into account in real systems using specialized techniques.

As mentioned above, there are instances where a uncertainty could be considered epistemic or aleatory depending on the context. For example, when we speak of the

uncertainty in a physical quantity, there may be a model implied in the idea of that quantity to begin with. One example is an equation of state model that assumes a gamma-law gas. The parameter γ may have a distribution depending on some ambient conditions, but there is no correct γ because it is derived from a simplified model of how gas molecules behave. In this case we may say that part of the uncertainty is aleatory (the value of γ that we use based on experimental data for the gas) and part is epistemic (the uncertainty due to using the simplified model).

1.6 Physics-based uncertainty quantification

An important consideration in uncertainty quantification is where uncertain data comes from. This will be a part of the process of identifying uncertainties, but it is important to think about the origin of data. For example, in some simulations the equation of state of the material (e.g., a relation between pressure, temperature, and internal energy of the material). As used in codes, the equation of state can be represented by a large look-up table in the code that could have thousands of entries. That does not mean there are thousands of uncertain parameters, however. The equation of state table is likely to be generated by a combination of experimental measurements, theoretical models, or simulations. Each of these components of the equation of state table will have it's own uncertainties. For example, the experiments will have uncertainties in the measurement, the theoretical models have parameters that will be uncertain, such as a gas-constant, and the simulations will also have uncertainties. The sum total of the uncertainties in these components is likely to be much smaller than the number of parameters in the table. Therefore, the true dimension of the uncertainty is not based on the equation of state table, but on the physics behind the table.

This is an example of physics-based uncertainty quantification, and it is an important illustration of the power that knowledge of the simulation and the processes behind it are useful to the uncertainty quantification practitioner. There are also many other ways that domain expertise can inform a UQ study. With knowledge of the properties of the inputs and QoIs, the UQ process can be tailored to the situation and be more efficient and more accurate. For instance if a parameter is known to be strictly positive, that will influence the type of distribution it can be. Also, if a QoI cannot be larger than a given amount, the UQ procedure should respect that.

These examples of physics-based uncertainty quantification indicate that the expertise that the scientist has cannot be forgotten when performing a UQ study, or, to put it the other way, the UQ expert is most effective when expert knowledge is combined with domain expertise from a scientist or engineer. Furthermore, this type of domain knowledge is not limited to physics, one could easily speak of chemistry-based or biology-based UQ or any number of other technical fields.

1.7 From simulation to prediction

Given the results of a UQ study, that is, knowledge of the QoI and its uncertainty, the next question what does one do with that information. There are several scenarios that serve as the bridge between understanding of parametric uncertainties and the application of that knowledge to making a prediction. As a way to highlight how this might be used we will detail some examples of predictive science in action.

1.7.1 Best Estimate plus Uncertainty

The term Best Estimate plus Uncertainty is used by regulators in nuclear reactor certification around the world. The term refers to the use of simulation codes and models that have been demonstrated to be applicable to the system and conditions under question. The values of the QoI (typically the probability of failure) are quoted at the most likely values of the uncertain parameters, this is the best estimate part of the equation, and then a confidence interval around that estimate, the uncertainty part. This confidence interval is estimated by sampling uncertain parameters, running a simulation, looking at the distribution of the outputs, or building an approximate model based on the outputs.

1.7.2 Quantification of Margins and Uncertainties

Quantification of Margins and Uncertainty (QMU) is concerned making decisions on whether a system will perform as expected given the a performance margin built into the system and the uncertainties in a simulation. In the simplest illustration we have a system where a performance metric can be in the range $[q, q+M]$, i.e., that is there is a margin of M. Then from a best estimate plus uncertainty study, we have a range of simulated system outputs, U. Then, based on the ratio M/U, as well as subject matter expertise (i.e., expert judgment), and small-scale experiments that test system components, the decision is made whether there probability of failure of the system is tolerable.

1.7.3 Optimization under uncertainty

Designing a system, while considering the uncertainties in simulations, is known as optimization under uncertainty. In this exercise, one wants to tune a system's performance by adjusting inputs, but taking into consideration the fact that one does not know precisely what the system performance will be. Unique problems arise in this type of optimization. It is possible that the global maximum (i.e., nominally

the best design) has a larger uncertainty than a lesser maximum that has a smaller uncertainty. In other words, if two designs give quantities of interest of q_1 and q_2 with $q_1 > q_2$, but the range of performance when considering uncertainty is $\pm 10\%$ for the design giving q_1 and $\pm 1\%$ for the other design, the second design may be a better choice if the worst-case performance is more important than the optimal performance.

1.7.4 Data-driven Experimental Design

One of the outcomes of an uncertainty quantification study could be which uncertainties in the inputs or models lead to the largest fraction of the uncertainty in the quantities of interest. With this knowledge the investment in additional experiments, higher fidelity models, additional computer resources, etc. can be prioritized. Making quantifiable statements about where uncertainty in our QoIs comes from and how to reduce them is a powerful result. If an engineer can say, 80% of the uncertainty in our system's performance is due to uncertainty in the melting point of a component, then we know that improving the knowledge of that melting point will result in a large decrease in uncertainty in the system performance. This is an important, but often overlooked, benefit of a rigorous uncertainty study.