

Being Sensitive to Uncertainty

Predictive modeling's effectiveness is hindered by inherent uncertainties in the input parameters. Sensitivity and uncertainty analysis quantify these uncertainties and identify the relationships between input and output variations, leading to the construction of a more accurate model. This survey introduces the application, implementation, and underlying principles of sensitivity and uncertainty quantification.

In the 1960s, Richard Hamming captured the spirit and essence of modern computational sciences in a profound statement: “The purpose of computing is insight, not numbers.” In this survey, “insight” means assessing the effects of uncertainties on input parameters and the subsequent effect on the simulation’s output. The basic tenet in *sensitivity analysis* (SA) and *uncertainty quantification* (UQ) is that variability in the input leads to variability in the output, thus the primary purpose of SA/UQ is to quantify these uncertainties. In doing so, we can place greater confidence in the model’s validity and prioritize the importance of the assumptions made about the model’s parameters, as well as make accurate predictions about system behavior.

We typically use *forward sensitivity analysis* (FSA) when the number of system responses, or outputs of interest, greatly exceeds the number of input variables. Figure 1a shows how small changes in a few input parameters can cause broad changes in the output. In contrast, we use *adjoint sensitivity*

analysis (ASA) when this ratio is reversed—for a given allowed variation in the output, ASA quantitatively determines the allowed variation in the inputs (see Figure 1b). Both the forward and adjoint methods are local to a particular solution, where the derivatives defining the sensitivities are evaluated at some fixed nominal values. Both methods form the basis for SA.

Global methods, such as UQ, use sampling and statistical techniques to assign distributions to inputs to determine the output’s resulting distributions. Suppose a model has two important input parameters, p_1 and p_2 , where each parameter has an associated *probability density function* (PDF), as Figure 2 shows. Uncertainties in the input parameters enter the model and produce uncertainty in the output. In other words, the output isn’t a single value but a PDF as well.

We can divide SA into two major categories: *analytic methods* aim to obtain explicit expressions for the desired derivatives, and *algorithmic methods* calculate the desired derivatives as a sequence of intermediate derivatives. This sequence arises as the algorithm evolves either forward or backward in time.

Alternatively, UQ quantitatively determines various measures of uncertainty in outputs as a result of input variability. Because UQ’s underpinnings are derived from statistical methods, its main techniques are based on sampling methods, and correlation and variance measurements. The inherent problem of

computational costs leads to the need for an efficient means of obtaining small samples of input values.

Spread of an Infectious Disease

Let's look more closely at SA/UQ in terms of a real-world scenario: the spread of a disease in a large, stratified population of people, most of whom are susceptible to the emerging contagion. Assume that the disease transmits itself when an infectious person has contact with a susceptible individual.

A detailed epidemiological model would incorporate important aspects of disease transmission, including incubation period, transmission probability, an individual's age, average mortality rate with and without the disease, intervention strategies, and so forth. Observational data helps us assign reasonable numerical distributions to these parameters, so once we specify the parameter distributions and the initial number of infectious individuals, the model's computer simulations can predict the complex system's behavior. An important response function for this problem would be a measure of how rapidly the disease spreads throughout the population.

Because very few of the parameters in this model are known precisely, an important aspect of the investigation is to determine how changes, or perturbations, to a generic parameter p (such as the disease's incubation period) affect the infectious class $I(t)$ at time t . In essence, we're trying to evaluate the derivative of I with respect to a particular parameter p —that is, the quantity $\partial I / \partial p$. This SA derivative quantifies how small perturbations in p will propagate forward in time.

We can now determine which parameters have the biggest effect on the disease's transmissibility. Because external intervention (such as vaccines, diagnosis, quarantine, and so on) can change only a particular subset of the parameters, calculating the *sensitivity indices* (SIs), $(\partial I / \partial p)(p/I)$, gives us a normalized measure of each option's efficacy. SIs also provide an equitable way to determine which parameters have the biggest effect on the disease's spread—Figure 3, for example, shows SIs for the infection level with respect to the effective contact rate I_r and the recovery rate I_r . In this figure, SIs are time dependent, and the effective contact rate has the biggest effect on the epidemic's spread 10 days into the outbreak. Because each intervention option has an associated cost, we can determine which combinations should be implemented to minimize the disease's spread for a fixed amount of money.

Industrial Air Pollution

Now let's consider a different scenario: a new industry is coming to a populated area, and we need

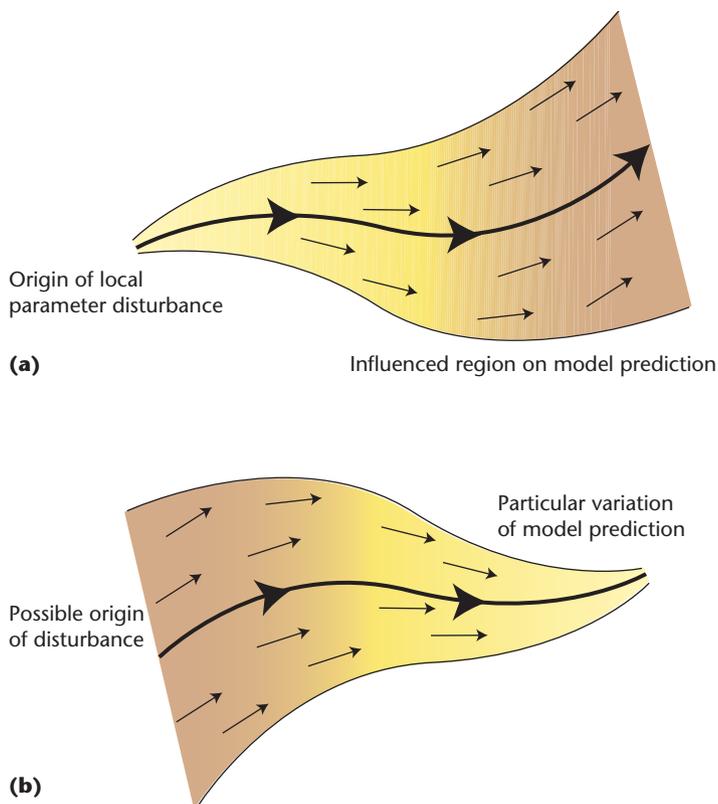


Figure 1. Forward and adjoint sensitivity. (a) Forward sensitivity analysis quantitatively determines how local perturbations to input parameters evolve forward in time. (b) Adjoint sensitivity analysis quantitatively determines how much perturbation is permitted to input parameters for a specified variation in the output. Although it looks like the perturbations are made to the initial conditions, the variations are in fact made to the defining parameters.

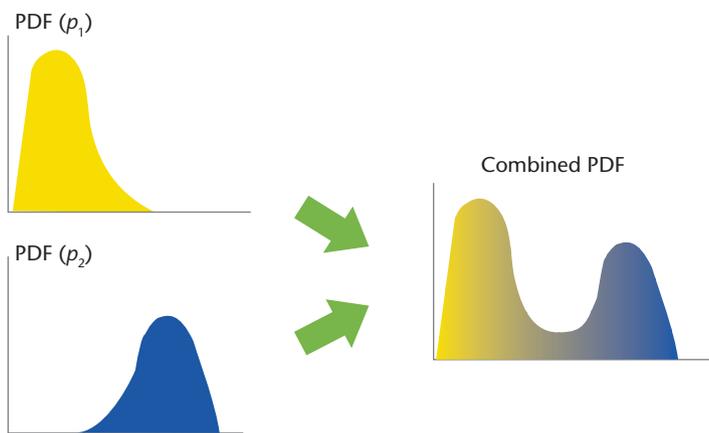


Figure 2. Uncertainty quantification. The probability density functions (PDFs) of two input parameters result in a multi-peaked PDF for the output variable.

a mathematical model to help guide the decision on the factory's location. The available sites are in

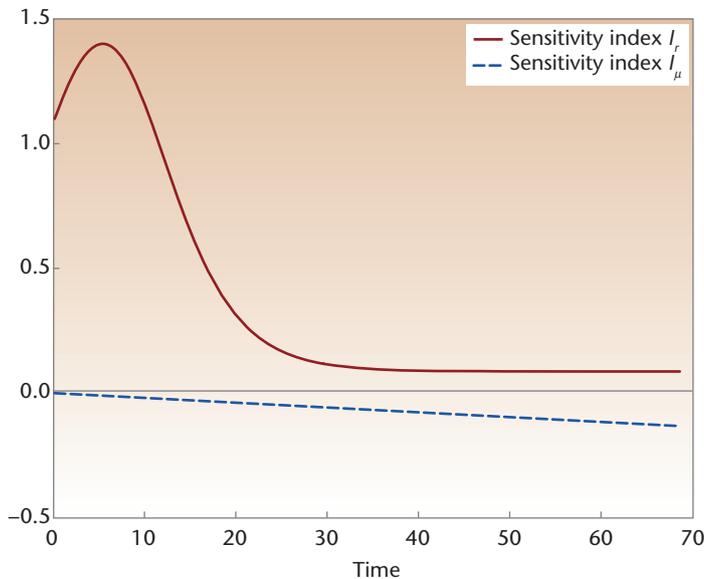


Figure 3. Sensitivity indices' effective transmission rate SI_r vs. recovery rate SI_μ . The number of people infected is most sensitive in the early stages of the epidemic, whereas the infected population's sensitivity with respect to the recovery rate is small and almost independent of time.

close proximity to businesses and recreational and residential areas, and this factory has a history of emitting toxic fumes. It would be a hazard to the community if toxic levels exceed US Environmental Protection Agency (EPA) standards.

The city planners want to know where to put the industry so as not to exceed EPA standards as well as determine how sensitive their assumptions are about the weather. An appropriate mathematical model for this application would be a nonlinear transport equation, with specified boundary conditions defined by local weather patterns. Assuming we have the pollutant source as an explicit forcing function and initial weather conditions, then we can use any of several existing numerical weather simulation codes to define and solve the associated dynamical atmospheric model to predict pollutant levels (see Figure 4). However, this solution doesn't directly answer where the factory should be located under a wide range of potential weather patterns. What we need here is an appropriate response function.

Because the planners want to meet EPA standards, these standards become the appropriate response functions—for example, one EPA standard might be that the average amount of an aerosol pollutant at a specified altitude shouldn't exceed a specified amount over a six-hour period. Or, if the aerosol precipitates to the ground, deposits on outdoor playground equipment, and accumulates, then an ap-

propriate response function might be the total accumulated toxic substance over a designated area and time period. In these cases, the response function isn't the solution itself, but a functional of the solution. Specifically, the functional takes the form of a double integral: one integration corresponds to the spatial component and the other to the temporal component. Evaluation of this functional over available sites provides a list of acceptable sites.

Suppose a particular site meets EPA standards. Before the industry accepts the designated site, company analysts will want to determine how much variation in pollutant production, wind, and so on is allowable, given that it's below the EPA standard. In other words, the solution to the transport problem, as given by the amount of pollution, can change, thus the associated response function also changes.

ASA can help answer questions about how much change in the defining parameters is allowable for a given acceptable increase in the response function. The way to answer this question is to construct another associated, or *adjoint*, problem. This associated problem can provide a way to calculate derivatives in a reverse or adjoint mode.

Nuclear Waste Repository

A common application of UQ is when input parameters don't have fixed or specific values but take on a range of values. Consider a study of proposed repository sites for nuclear waste. Clearly, the underlying geological structure will have a major effect on the unwanted diffusion of radioactive materials in the ground. At best, geologists can broadly estimate the underlying geological structure's diffusive properties—one region of the proposed repository might have diffusive properties that follow a normal distribution, for example, and others might follow logistic, lognormal, or uniform distributions. The precise location of interfaces between boundary layers of materials with differing densities, porosity, and so forth also causes uncertainty in the input variables.

Naturally, the inevitability of these inherent uncertainties in the inputs will produce uncertainty in the model's output variables. Moreover, certain inputs, such as the geological strata, might be correlated and have the tendency to aggregate locally. The model must reflect these correlations between input parameters because they affect the output's UQ.

In this example, UQ's primary goal is to determine whether certain parameters—and their variation—result in waste outputs that exceed the levels imposed by regulatory agencies. UQ's methodology is based on standard statistical methods and numerical estimates (given as distributions and

confidence intervals) and can help determine a proposed site's suitability. The information UQ provides also helps identify which assumptions have the biggest effect on unwanted nuclear waste diffusion, which is useful in allocating funding for expensive geological surveys (which, in turn, help reduce the model's uncertainty).

Forward and Adjoint Sensitivity

A common but extremely inefficient way to determine how a model's solution is affected by perturbations to input parameters is to choose nominal values for the parameters, run a simulation, and get the solution. We continue in this manner until each individual input parameter is perturbed and the subsequent solution calculated. This brute-force method produces numerical estimates of derivatives, but it's computationally intensive. A better approach is to introduce an additional problem—specifically, an adjoint problem. It eliminates the need to use brute-force methods and yields the desired derivatives by only having to solve—once each—the forward and adjoint problems.

System of Linear Equations

To introduce the concept of an associated adjoint problem, let's consider the ubiquitous system of linear equations $Au = b$, where A is a nonsingular and nonsymmetric $n \times n$ matrix. SA, as applied to this problem, can help determine how the solution vector u changes as small perturbations are made to the matrix entries a_{ij} , or b_i . Assuming that the solution is sufficiently far from any singularities, SA attempts to find explicit expressions for the derivatives $\partial u / \partial q$, where q denotes any of the parameters a_{ij} , or b_i .

Differentiating the linear system gives the forward sensitivity equation

$$A \frac{\partial u}{\partial q} = \frac{\partial b}{\partial q} - \frac{\partial A}{\partial q} u. \quad (1)$$

An extremely poor way of solving for the desired derivative $\partial u / \partial q$ would be to exploit the fact that A is assumed to be nonsingular and then premultiply the equation by the matrix inverse. As any experienced modeler knows, explicitly finding the inverse should, in most cases, be vehemently avoided.

A more elegant way of solving for $\partial u / \partial q$ is to introduce an associated adjoint problem

$$A^T v = c, \quad (2)$$

where v is the associated adjoint variable, and c is unspecified. Leaving the adjoint constraint vector c unspecified—for now—is the key to isolating the

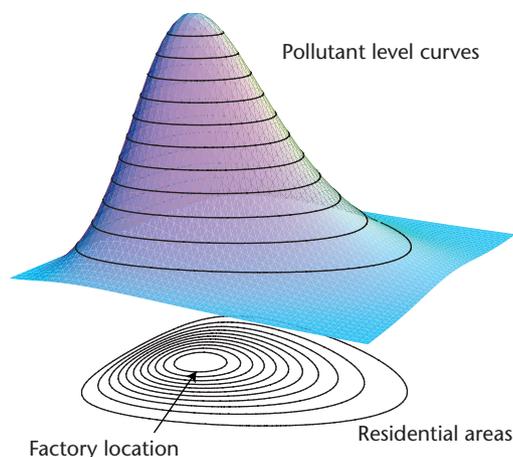


Figure 4. Pollutant levels indicated by level curves. These levels are greatest near the factory and downwind from the pollution source. Sensitivity analysis can measure the change in pollution levels in the residential areas as a function of changes in weather-pattern assumptions.

desired derivatives. Taking the transpose of the adjoint Equation 2 and premultiplying the forward sensitivity Equation 1 by v^T yields

$$v^T A \frac{\partial u}{\partial q} = v^T \left(\frac{\partial b}{\partial q} - \frac{\partial A}{\partial q} u \right).$$

Now we can cleverly choose a set of adjoint constraint vectors to be

$$c_i^T = (0 \ 0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0), \quad (3)$$

where the 1 is located in the i th entry for $i = 1, 2, \dots, n$. This particular choice effectively extracts the i th component $\partial u_i / \partial q$.

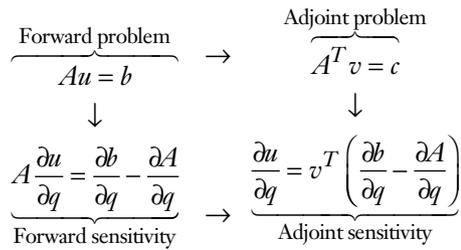
Notice that the adjoint solution has—in effect—solved the linear system, component by component. This observation reflects the intimate relationship between the adjoint solution and the inverse matrix. To show that this isn't a mere coincidence, we can form the adjoint solution matrix, whose rows are the adjoint vectors:

$$V := \begin{pmatrix} v_1^T & v_2^T & \dots & v_n^T \\ v_{11} & v_{21} & \dots & v_{n1} \\ v_{12} & v_{22} & \dots & v_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1n} & v_{2n} & \dots & v_{nn} \end{pmatrix},$$

in which case $A^T V = I := \text{identity matrix}$. In other

words, the adjoint solutions are directly related to the inverse matrix—specifically, $V^T = A^{-1}$.

We can use a diagram to demonstrate:



We can apply this elegant and powerful technique for constructing an appropriate adjoint to other static problems, including spectral, linear and quadratic programming, nonlinear optimization, and nonlinear equation problems.

In our example, the crucial aspects of finding the derivatives were twofold:

- to construct an appropriate adjoint problem $A^T v = c$, and
- to construct appropriate response functionals $\mathcal{J}_i(u) := c_i^T \cdot u$.

A clever choice of the response function \mathcal{J} provides an appropriate and useful associated problem: the adjoint problem. In this case, we could formulate the response functional $c_i^T \cdot u$ as a projection operator.

You're probably wondering if these fortuitous choices were a fluke—we hope to convince you otherwise in the next section.

Adjoint Operator and Problem

The types of problems that are amenable to the adjoint methodology are those that we can express as

$$F(u) = f,$$

where F is a linear/nonlinear operator $F : X \rightarrow Y$, and f is the forward forcing function. We assume the domain and range X and Y are sufficiently nice topological spaces—for example, both X and Y could be Hilbert or Sobolev spaces. Also associated with the forward problem is the task of determining the sensitivity of a desired response functional $\mathcal{J}(u)$.

The adjoint problem arises naturally with the introduction of an adjoint variable $v \in X$, through the calculation of the Gâteaux derivative defined as

$$F'(u)v := \lim_{\varepsilon \rightarrow 0} \frac{F(u + \varepsilon v) - F(u)}{\varepsilon}.$$

Think of this definition as a directional derivative of

operator F at point u , and in the direction of adjoint variable v . The somewhat awkward notation $F'(u)v$ is intended to suggest that operator F takes the forward variable u and maps it to operator F' , which now depends on both u as well as the adjoint variable v .

The next piece of necessary machinery is to formulate an extended representation of operator F , which we accomplish by assuming that F is sufficiently Gâteaux-differentiable. Application of the intermediate value theorem for operators permits us to rewrite the forward operator F in extended form as

$$\Phi(u)u = F(u),$$

where the residual operator Φ is defined in integral form as

$$\Phi(u) := \int_{\tau=0}^1 F'(\tau u) d\tau.$$

Given that an appropriate inner product exists, let's consider the adjoint operation

$$\langle \Phi(u)v, w \rangle = \text{SC1} + \langle v, \Phi^\dagger(u)w \rangle,$$

where SC1 denotes the first *solvability condition*, and Φ^\dagger denotes the adjoint operator associated with the forward operator. (A simple example here is in order. Consider the process of integration by parts, where the associated inner product is a definite integral. In essence, a derivative is shifted off the forward variable onto the adjoint variable, and in the process, we must formulate appropriate *boundary conditions* (BCs) so that the adjoint problem is correctly defined. In the more abstract operator setting, the SC1 is, in fact, the analogy of the BCs. Appropriately choosing the adjoint variable's boundary values will ensure that the SC1 is annihilated.) When SC1 = 0, we refer to the result as the *Lagrange identity*.

The associated generalized adjoint problem is

$$\Phi^\dagger(u)v = g,$$

where the adjoint forcing function g hasn't yet been specified. As in the linear system problem, not specifying g at this time is advantageous because it might be cleverly defined later in terms of the response functional \mathcal{J} to ensure that the solvability conditions are satisfied.

A second solvability condition SC2 occurs when the forward and adjoint problems are related. Assuming that the Lagrange identity is satisfied—that is, SC1 = 0—then taking the inner product of the forward problem with the adjoint solution gives

$$\langle \Phi(u)u, v \rangle = \langle f, v \rangle,$$

while taking the inner product of the adjoint problem with the forward solution gives

$$\langle \Phi^\dagger(u)v, u \rangle = \langle g, u \rangle$$

$$\langle v, \Phi(u)u \rangle = \langle g, u \rangle$$

$$\langle v, f \rangle = \langle g, u \rangle.$$

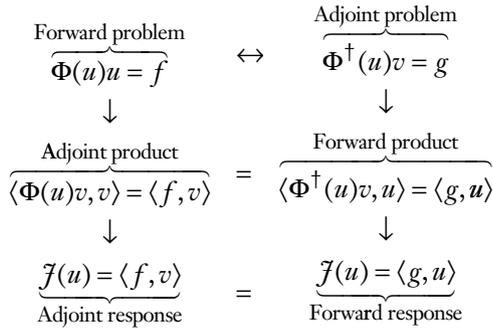
This invariance condition, or second solvability condition SC2, relates the forward and adjoint solutions and forcing functions by the condition

$$\langle g, u \rangle = \langle f, v \rangle,$$

in which case, we define the adjoint forcing function g so that

$$\langle g, u \rangle = \mathcal{J}(u).$$

We can use a diagram to visualize the adjoint problem's construction:



For the linear system, the adjoint methodology produces the adjoint problem $A^T v = c$ when the operator equation $F(u) = f$ is constructed from Equation 1. Specifically, the results follow when

$$F(u) := A \frac{\partial u}{\partial q}, \quad f := \frac{\partial b}{\partial q} - \frac{\partial A}{\partial q} u, \quad \text{and}$$

$$\mathcal{J}(u) := \left\langle \frac{\partial u}{\partial q}, c \right\rangle.$$

Forward and Reverse Modes

In contrast to the static model in the previous section, other commonly occurring models evolve temporally as well as spatially. Moreover, the temporal and spatial components can be continuous or discrete.

Initial Value Problem

We use nonlinear systems of *ordinary differential equations* (ODEs) to describe a dynamical system's evolution. In theory, a solution that's differentiable in time exists, but we usually can't find the explicit solution. In this case, we apply numerical algorithms to produce a sequence $\{u_n\}$ that approximates the actual solution $u(t_n)$ at time t_n .

Consider the scalar, autonomous, nonlinear ODE

$$\frac{du}{dt} = F(u; p), \quad (4)$$

with given initial condition u_0 and parameter p , and where $t \in [0, b]$. Associated with this dynamical system is a response functional $\mathcal{J} = \mathcal{J}(u)$, which depends on the forward solution u . SA requires calculation of the SI

$$\frac{q}{\mathcal{J}} \frac{d\mathcal{J}}{dq} = \frac{q}{\mathcal{J}} \frac{\partial \mathcal{J}}{\partial q} \left(\frac{\partial u}{\partial u_0} \frac{du_0}{dq} + \frac{\partial u}{\partial p} \frac{dp}{dq} \right),$$

where q denotes either u_0 or p .

To calculate this SI, we must evaluate the time-varying derivatives $\partial u / \partial u_0$ and $\partial u / \partial p$, which we accomplish by solving the forward sensitivity equations

$$\frac{d}{dt} \left[\frac{\partial u}{\partial u_0} \right] = \frac{\partial F}{\partial u} \frac{\partial u}{\partial u_0}, \quad \frac{d}{dt} \left[\frac{\partial u}{\partial p} \right] = \frac{\partial F}{\partial u} \frac{\partial u}{\partial p} + \frac{\partial F}{\partial p},$$

simultaneously with the forward problem in Equation 4. If $q = u_0$, the initial conditions are $\partial u / \partial u_0|_{t=0} = 1$ and $\partial u / \partial p|_{t=0} = 0$, whereas if $q = p$, the initial conditions are reversed.

Because—in general—we can't solve this dynamical system in closed form, we apply a robust numerical algorithm to obtain a sufficiently accurate approximation over some specified grid. These algorithms frequently take the form of an iterative scheme—for example, when we initially apply a single-step algorithm, $u_1 \approx u(t_1)$ is constructed from the given initial condition u_0 as

$$u_1 := \Psi_1[u_0; p].$$

Subsequent approximations are generated by repeated compositions of the operators Ψ_k :

$$\begin{aligned} u_2 &:= \Psi_2[u_1; p] \\ &= \Psi_2[\Psi_1[u_0; p]] \\ &= \Psi_2 \circ \Psi_1[u_0; p] \\ &\vdots \end{aligned}$$

$$u_{n+1} := \Psi_{n+1} \circ \Psi_n \circ \dots \circ \Psi_1[u_0; p].$$

The variables arising in the forward and adjoint modes are categorized as input, intermediate, and output variables. In this example, the input variables are the initial condition u_0 and the parameter value p . The intermediate variables are the approximations u_n , and the final output variables are response functionals of the solution—namely, $\mathcal{J}(u_{\text{end}})$. Execution of a single-step algorithm, in the forward mode, calculates the derivatives in order:

$$\frac{du_1}{dq} \rightarrow \dots \frac{du_n}{dq} \rightarrow \dots \frac{du_{\text{end}}}{dq}.$$

We calculate the specific differentiations by using compositions of the chain rule:

$$\begin{aligned} \frac{du_1}{dq} &= \frac{\partial \Psi_1}{\partial u_0} \frac{du_0}{dq} + \frac{\partial \Psi_1}{\partial p} \frac{dp}{dq} \\ \frac{du_2}{dq} &= \frac{\partial \Psi_2}{\partial u_1} \frac{du_1}{dq} + \frac{\partial \Psi_2}{\partial p} \frac{dp}{dq} \\ &= \frac{\partial \Psi_2}{\partial u_1} \left(\frac{\partial \Psi_1}{\partial u_0} \frac{du_0}{dq} + \frac{\partial \Psi_1}{\partial p} \frac{dp}{dq} \right) + \frac{\partial \Psi_2}{\partial p} \frac{dp}{dq} \end{aligned}$$

:

$$\frac{du_{n+1}}{dq} = \frac{\partial \Psi_{n+1}}{\partial u_n} \frac{du_n}{dq} + \frac{\partial \Psi_{n+1}}{\partial p} \frac{dp}{dq}$$

We call these calculations the *forward mode* because the derivatives du_n/dq are calculated successively—that is, recursively forward in time. Each intermediate variable u_n is differentiated here with respect to input variable q .

The *adjoint mode*, however, reverses these evaluations in the sense that we calculate and evaluate the output variable's derivative with respect to the intermediate variables in a reverse order. Let $u_{\text{end}} = u$; the adjoint derivatives are

$$\frac{\partial u}{\partial u_{n+1}} \rightarrow \frac{\partial u}{\partial u_n} \rightarrow \dots \frac{\partial u}{\partial u_1} \rightarrow \frac{\partial u}{\partial q},$$

where they're evaluated in reverse mode:

$$\frac{\partial u}{\partial u_{\text{end}}} = 1$$

:

$$\frac{\partial u}{\partial u_n} = \frac{\partial u}{\partial u_{n+1}} \frac{\partial u_{n+1}}{\partial u_n}$$

:

$$\frac{\partial u}{\partial u_1} = \frac{\partial u}{\partial u_2} \frac{\partial u_2}{\partial u_1}.$$

In the more general case—in which the numerical algorithm isn't simply a single-step method but a sequence of function evaluations—the reverse mode leads to a linear system. The coefficient matrix of the associated matrix equation is the transpose of the Jacobian matrix. The adjoint, disguised as the matrix transpose, naturally appears in the reverse mode's execution.

Computational Issues

When deriving the adjoint equations for discrete approximations of differential equations, the question arises whether it's better to first discretize the equations and then form the adjoint equations, or to form the adjoint equations of the differential equations and then discretize the equations independently. The first approach is aided by automatic differentiation software tools¹ and gives an accurate SA of the finite difference approximation, even when it's not an accurate approximation of the original differential equation. The second approach requires both the forward and adjoint equations to be accurately approximated for the results to be valid. Naturally, there are strong arguments in favor of both approaches.

When estimating the sensitivities of finite difference methods, we must be especially careful to avoid potentially large truncation or round-off errors. Even though the numerical solution of the differential equations are computed accurately, it doesn't necessarily mean that the numerical approximations of the sensitivities are accurate.

Uncertainty Quantification

The defining parameters of models in environmental, financial, industrial, and risk analysis settings are often known only within a specified interval—namely, $p_i \in [\alpha_i, \beta_i]$. The basic assumption is that each input parameter is viewed as a random variable, with an associated PDF and a *cumulative distribution function* (CDF). In this situation, calculating sensitivity indices by explicitly finding derivatives is inappropriate. (In discrete event systems, we can obtain gradient estimates in essentially two broad ways: infinitesimal perturbation analysis or likelihood ratio/score function methods. Strictly speaking, these methods do indeed estimate derivatives.) Instead, the measure of choice is based on estimating the variance of the Monte Carlo method's output. The goals here are usually priority setting: determine which of the many parameters must be precisely measured and which parameters produce the most output variance. If we use UQ in a forecasting mode, then the parameters' PDFs take on an assumed character, and thus determine the output or response's character. If we use UQ in a calibration mode, then we can use the

output's character to change the assumptions of the input parameters' PDFs. UQ execution generally follows this procedure:

- Assign an appropriate PDF to each input parameter.
- Select a representative set of samples from each PDF.
- Evaluate the model and obtain the associated set of outputs.
- Define and evaluate suitable measures for the SA and UQ.

In any UQ discussed here, the mathematical model F is assumed to be deterministic, but the inputs are treated as stochastic quantities. (This is in sharp contrast to the model structure's sensitivity. In this arena, we frequently use principal component analysis to identify the model's strongly correlated and uncorrelated components. Using this information, we can break the model into submodules, each of which we can analyze individually.)

Sampling Methods

Consider the single output model, defined as

$$u := F(p),$$

where $p := (p_1, \dots, p_N)$, and each input parameter is assumed to be $p_i \in [\alpha_i, \beta_i]$ intervals. We generate a sample of M parameter vectors from the designated PDF and evaluate the output function. The way in which a sample is constructed is extremely important in determining the output's perceived uncertainty as well as the inputs' relative importance. Let's review some of the more commonly used sampling methods.

Random Sampling

The simplest approach is to select random samples of numbers x in the interval $[0, 1]$. For the chosen number x , we determine the associated sample value p_i in the interval $[\alpha_i, \beta_i]$, by finding the inverse of the CDF—that is, $p = \text{CDF}^{-1}(x)$. This sampling strategy is easy to implement, and for large samples, it produces unbiased estimates of the mean and variance of output U . However, if we divide the sample interval $[\alpha_i, \beta_i]$ into a large number of equally sized subintervals, and a relatively small number of samples is taken, we have no guarantee that all the subintervals will be sampled.

Stratified Sampling

A more sophisticated sampling strategy is stratified sampling. This variance reduction method first divides the input parameter interval into (pos-

sibly nonuniform) strata or subintervals. Next, a random sample is chosen from each of the subintervals. The motivation behind stratified sampling is to ensure that we obtain samples from each particular subinterval.

Another commonly used stratified sampling method is Latin hypercube sampling. To visualize this process, imagine a square with M equally divided rows and columns. A Latin square has the geometric property that in each row and column, one and only one cell is occupied. This sampling method's strategy is to replace the concept of a particular cell being occupied with getting a sample from that particular associated subinterval. Orthogonal sampling adds the additional requirement that the distribution of samples be evenly distributed from the sample subspaces.

Latin hypercube sampling is slightly easier to implement than orthogonal sampling. Both methods return a sample that gives a good representation of variability, as well as reduce the variance in the output's evaluation. The main advantage to using either method is that we can obtain smaller sample sizes without a major subsequent loss of quantification of the output's variance. In comparison, random sampling generates new samples without taking into account previous ones, which means it doesn't require advanced knowledge of how many sample points are needed (the reverse is true of Latin hypercube sampling and orthogonal sampling).

If the underlying input parameters are correlated, choosing samples can be more difficult. Another caveat is that it's possible to artificially introduce correlations between parameters when none actually exist.

Measuring Sensitivity and Uncertainty

The next step is to statistically analyze each of the M realizations u_i to determine the expected value and variance. We can choose other, more informative measures depending on whether the relationship between the input and output uncertainties appears to be linear or nonlinear.

Linear Regression Analysis

If the relationship between input and output uncertainties is essentially linear, the standard effective methods include regression, correlation, and partial correlation analysis. Standardized regression coefficients provide quantitative measures of the specific, individual importance of how closely linked the output is with the input. Essentially, these coefficients measure the effect of individually changing each input parameter's value by a fixed portion of its standard deviation.

Partial correlation coefficients quantify the strength of the linear correlations between the input and output while removing correlations between other inputs.

Rank Transformation

When the relationship between input and output data appears to be nonlinear but monotonic, we typically use the rank transformation methodology. As the name suggests, input data is ranked in increasing order, and the associated output is rearranged accordingly. We then perform the usual regression analysis on the ranked data set to produce the Spearman rank coefficient. This method produces a measure of how strong a correlation there is in the monotonicity between the input and output data.

These measures assume that the input variables are independent—that is, that they aren't correlated. When this isn't true, these methods can give erroneous results, and more sophisticated methods might have to be applied.

All the methods discussed here rely on some form of regression. A more in-depth analysis of these methods reveals that we can derive them from the concept of variance $\text{Var}_P[\mathbb{E}[U|P]]$, where P and U are the input and output spaces, respectively, and \mathbb{E} is the expected value. More sophisticated methods include Sobol' and the Fourier amplitude sensitivity test.

Sobol' Method

To intuitively understand the Sobol' method, recall the essence of a Fourier series—a series of mutually orthogonal and harmonic functions can represent an arbitrary function. The Sobol' method is similar in that it decomposes the output model function into a unique sum of orthogonal functions of increasing dimension.

Consider the three-input, single-output model

$$u = F(p_1, p_2, p_3).$$

The associated Sobol' decomposition is of the form

$$F(p_1, p_2, p_3) = F_0 + F_1(p_1) + F_2(p_2) + F_3(p_3) + F_{12}(p_1, p_2) + F_{13}(p_1, p_3) + F_{23}(p_2, p_3) + F_{123}(p_1, p_2, p_3),$$

where the individual decomposition functions are mutually orthogonal with respect to integration over the input sample space and are defined inductively.

Once this has been accomplished, the variance is similarly decomposed as

$$D = D_1 + D_2 + D_3 + D_{12} + D_{13} + D_{23} + D_{123},$$

from which the first-order sensitivity coefficients are defined as

$$S_1 := \frac{D_1}{D}, \quad S_2 := \frac{D_2}{D}, \quad \text{and} \quad S_3 = \frac{D_3}{D}.$$

Higher-order sensitivities such as S_{12} , S_{13} , S_{23} , and S_{123} are similarly defined. The Sobol' sensitivity coefficients (sometimes called total SIs) quantify the fraction of the variance contributed to the total variance by individual or combinations of inputs. In other words, D_2 provides a quantitative measure of the fraction of the total variance that p_2 contributes, whereas D_{13} describes the combined contribution from p_1 and p_3 .

This method's advantage is that higher-order sensitivities give additional measures of how multiple parameters can affect the output's overall variance. However, its disadvantage is the amount of computation needed to calculate all these coefficients—namely, $M \times 2^N$ evaluations. If the number of parameters N is large, clearly the computational cost becomes prohibitive.

Fourier Amplitude Sensitivity Test

The Fourier amplitude sensitivity test's main goal is to evaluate the expected value of the output's k th moment—namely, the multidimensional integral

$$\mathbb{E}[U^k] := \int_P F^K(p) PDF(P) dp.$$

This integral's evaluation provides the basis for various sensitivity measures. The methodology for evaluating this integral is to construct an appropriate sequence of parameterized frequency transformations

$$p_i = G_i(\omega_i, s), \quad s \in (-\pi, \pi), \quad \text{for } i = 1, \dots, N,$$

where the G_i and ω_i are as of yet unspecified functions and frequencies, respectively. When properly chosen, the expected value $\mathbb{E}[U]$ is now approximated by

$$\mathbb{E}[U] \approx \int_{s=-\pi}^{\pi} F(s) ds.$$

Additionally, we can approximate the variance $\mathbb{V}[U]$ with a series of Fourier coefficients—specifically,

$$\mathbb{V}[U] \approx 2 \sum_{j=1}^{\infty} (A_j^2 + B_j^2),$$

where

$$A_j := \frac{1}{2\pi} \int_{s=-\pi}^{\pi} F(s) \cos(js) ds,$$

and

$$B_j := \frac{1}{2\pi} \int_{s=-\pi}^{\pi} F(s) \sin(js) ds.$$

The main challenges are to

- define appropriate transformations G_i and frequencies ω_i , and
- sample the parameter space on a sufficiently fine mesh to accurately evaluate Fourier coefficients.

We frequently use the arcsine function as the transformation, where the frequencies are required to be incommensurate. (A set of frequencies is incommensurate, provided any arbitrary frequency in that set can't be written as a linear combination—with integer coefficients—of any other frequency.) These particular choices create a sequence of space-like-filling curves that consist of oscillating straight lines. Because the frequencies are incommensurate, these lines are guaranteed to come arbitrarily close to any element in the input space. In other words, these transformations attempt to foliate the input parameter space.

One advantage of using the Fourier amplitude sensitivity test is that it works well for both monotonic and non-monotonic outputs. Furthermore, it has no restrictions such as fixing the values of all the parameters except one to determine sensitivity. This beneficial aspect allows for wide ranges of parameter sampling and the possibility of identifying extreme events. Finally, the extended Fourier amplitude sensitivity test method briefly described here is also significantly faster than Monte Carlo methods. However, if we have a large number of inputs, we can encounter significant computational complexity.

Discrete Event Simulations

Many modern problems of interest contain two fundamentally distinct aspects from previous models: the system's state space is a discrete set, and the state transitions are event driven. We call these types of models *discrete event simulations*.

Let's consider the standard queueing problem: clients or customers must wait in line to use a limited resource. Customers arrive and wait in a queue, and once a server has completed their request, they depart. We can thus define the event space as arriving/waiting, processing, and departing. Associated with the event space are the state variables—the number of customers waiting to be serviced in a given queue, the waiting time, and the server state are all quantities of interest here. A typical sensitivity concern is how the servers' processing time affects the mean wait time or number of clients in a queue. In this case, we assume the parameter varies continuously—that is, the process-

ing time can improve or degrade continuously over time. A different situation occurs when the parameter can only take on discrete variables—if we increase or decrease the queueing capacity, for example, then by definition those changes will occur only by integer increments.

UQ, using the previously discussed sampling methods, requires multiple runs to generate the associated output. Taking a different approach, two methods that don't require multiple runs are the score function method and infinitesimal perturbation analysis. The elegance of both methods is that we can determine sensitivity with a single sample path—that is, a single simulation. This elegance, however, comes with a price: we must have some knowledge of the PDF and CDF for the decision variables.

Score Function Method

To illustrate the idea of the score function method, let $u \in U$ denote a random variable with CDF $y = F(u; p)$, where p denotes some parameter. Suppose the measure of interest is the expectation value of $\mathcal{L}[u]$ —that is,

$$\mathcal{J}(U) := \mathbb{E}[\mathcal{L}[U]] = \int_U.$$

We find the sensitivity by assuming that the differentiation and integration operators commute, in which case

$$\frac{\partial \mathcal{J}(U)}{\partial p} = \int_U \mathcal{L}[u] \eta(u; p) dF(u; p),$$

where the likelihood ratio is defined as $\eta(u; p) := (\partial/\partial p)[\ln[dF(u; p)]]$. In other words, we can calculate the moments' sensitivity as the expectation of another function—namely, $\mathcal{L}[u] \eta(u; p)$. Because only a discrete sample is available, the estimate reduces to

$$\frac{\partial \mathcal{J}(U)}{\partial p} \approx \frac{1}{M} \sum_{i=1}^M \mathcal{L}[u_i] \eta(u_i; p).$$

The score function method's advantage is that the estimate is relatively easy to calculate. However, its downside is that this sum of positive terms increases proportionately to the simulation's length, in which case the associated variance is an increasing function. This has the effect of yielding poor variance estimates.

Infinitesimal Perturbation Analysis

The motivation behind infinitesimal perturbation analysis is to calculate the expectation value using the inverse transform method. Because the CDF's

inverse is given by $u = F^{-1}(y; p)$, we can rewrite the expected value of \mathcal{L} in the inverse form

$$\mathcal{F}(p) := \int_{y=0}^1 \mathcal{L}[y(p; u)] dy.$$

Once again, let's assume that the differentiation and integration operators commute, so we can now write the derivative sensitivity as

$$\frac{\partial \mathcal{F}(p)}{\partial p} = \int_{y=0}^1 \frac{\partial \mathcal{L}[y(p; u)]}{\partial y} \frac{\partial y(p; u)}{\partial p} dy.$$

Notice how the two derivative expressions reflect the fact that perturbations to parameter p change the sample function and the sample.

The main areas for applying this method are in queueing theory—specifically, when timed events aren't significantly affected by perturbations to the underlying parameters. The biggest disadvantage of infinitesimal perturbation analysis is its limited applicability.

For those readers who want to perform SA or UQ on a particular application, we encourage you not to write your own computer code but to take advantage of existing packages. When doing an Internet search, use the keyword automatic differentiation (AD). A helpful site is www.autodiff.org; it offers links to packages such as ADIC (C/C++), ADIFOR (Fortran 77), or AdiMAT (Matlab). The Joint Research Center for the European Commission's site on SA and UQ (<http://sensitivity-analysis.jrc.cec.eu.int/>) offers tutorials, conference proceedings, forums, and other helpful information.

We hope the discussions and examples in this survey will also whet your appetite for further investigation of the applications, implementation, and theoretical foundations of SA and UQ. To help motivate you, we've provided a nonexhaustive list of references to specific topics of interest. These references discuss, in much greater detail, the ideas presented in this article.

The authors of a good, basic introduction to UQ numerically examine several test cases via a downloadable program called Simlab, which is available at www.jrc.cec.eu.int/uasa/primer-SA.asp.¹¹ The same site also has downloadable scripts in Matlab and Gluewin. The authors of a more in-depth and broader coverage of UQ offer a list of comprehensive references; their bibliography is 21 pages long!¹⁰

Two recent monographs cover adjoint formalism in great depth and sophistication,^{7,8} and both are very thorough. Two other studies extensively

cover the FSA and ASA methods in two separate volumes.^{2,3} Another study focuses on the computational issues that arise when implementing algorithmic methods and offers a healthy dose of warnings and discussion about computational efficiency.¹ Lastly, separate researchers extensively discuss infinitesimal perturbation analysis and the score function method and thoroughly cover applications and theory.^{4-6,9}

Acknowledgments

This work was performed under the auspices of the US National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under contract number DE-AC52-06NA25396 and was partially funded by the DOE Office of Science ASCR program in applied mathematics.

References

1. A. Griewank, *Evaluating Derivatives: Principles and Techniques of Automatic Differentiation*, SIAM Press, 2000.
2. D. Cacuci, *Sensitivity and Uncertainty Analysis: Theory*, vol. 1, Chapman & Hall/CRC, 2003.
3. D. Cacuci, M. Ionescu-Bujor, and I. Navon, *Sensitivity and Uncertainty Analysis: Applications to Large-Scale Systems*, vol. 2, Chapman & Hall/CRC, 2005.
4. M. Fu and J. Hu, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*, Kluwer Academic, 1997.
5. P. Glasserman, *Gradient Estimation via Perturbation Analysis*, Kluwer Academic, 1991.
6. Y. Ho and X. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer Academic, 1991.
7. G. Marchuk, *Adjoint Equations and Analysis of Complex Systems*, Kluwer Academic, 1995.
8. G. Marchuk, V. Agoshkov, and V. Shutyayev, *Adjoint Equations and Perturbation Algorithms*, CRC Press, 1996.
9. R. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, John Wiley & Sons, 1993.
10. A. Saltelli, K. Chan, and E. Scott, *Sensitivity Analysis*, John Wiley & Sons, 2000.
11. A. Saltelli et al., *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, John Wiley & Sons, 2004.

Leon M. Arriola is a professor of mathematics at the University of Wisconsin–Whitewater. His research interests include sensitivity analysis, functional equations, and modeling the spread of epidemics. Arriola has a PhD in mathematics from Old Dominion University. Contact him at arriolal@uww.edu.

James M. Hyman is group leader of the applied mathematics group at Los Alamos National Laboratory and past president of SIAM. His research interests include the numerical solution and analysis of partial differential equations. Hyman has a PhD in mathematics from the Courant Institute of Mathematical Sciences. Contact him at hyman@lanl.gov.