20

Stochastic Simulation Methods for Engineering Predictions

Dan M. Ghiocel
Ghiocel Predictive Technologies, Inc.

20.1 Introduction

The term "simulation" comes from the Latin word *simulatio*, which means to imitate a phenomenon or a generic process. In the language of mathematics, the term "simulation" was used for the first time during the World War II period at the Los Alamos National Laboratory by the renowned mathematicians Von Neumann, Ulam, Metropolis and physicist Fermi, in the context of their nuclear physics research for the atomic bomb. About the same time they also introduced an exotic term in mathematics, namely "Monte Carlo" methods. These were defined as numerical methods that use artificially generated statistical selections for reproducing complex random phenomena and for solving multidimensional integral problems. The name "Monte Carlo" was inspired by the famous Monte Carlo casino roulette, in France, that was the best available generator of uniform random numbers.

In scientific applications, stochastic simulation methods based on random sampling algorithms, or Monte Carlo methods, are used to solve two types of problems: (1) to generate random samples that belong to a given stochastic model, or (2) to compute expectations (integrals) with respect to a given distribution. The expectations might be probabilities, or discrete vectors or continuous fields of probabilities or in other words, probability distributions. It should be noted that if the problem is solved,

0-8493-1180-X/04/$0.00+$1.50

so that random samples are available, then the solution to the problem, to compute expectations, becomes trivial because expectations can be approximated by the statistical averaging of the random samples.

The power of Monte Carlo methods manifests visibly for multidimensional probability integration problems, for which the typical deterministic integration algorithms are extremely inefficient. For very low-dimensional problems, in one dimension or two, Monte Carlo methods are too slowly convergent and therefore there is no practical interest in employing them. For a given number of solution points N, the Monte Carlo estimator converges to the exact solution with rate of the order $O(N^{-1/2})$ in comparison with some deterministic methods that converge much faster with rates up to the order $O(N^{-4})$ or even $O(\exp(-N))$. Unfortunately, for multidimensional problems, the classical deterministic integration schemes based on regular discretization grids fail fatally. The key difference between stochastic and deterministic methods is that the Monte Carlo methods are, by their nature, meshless methods, while the deterministic integration methods are regular grid-based methods. The problem is that the classical grid-based integration methods scale very poorly with the space dimensionality.

The fact that the Monte Carlo estimator convergence rate is independent of the input space dimensionality makes the standard Monte Carlo method very popular for scientific computing applications. Although the statistical convergence rate of the Monte Carlo method is independent of the input space dimensionality, there are still two difficulties to address that are dependent on the input space dimensionality: (1) how to generate uniformly distributed random samples in the prescribed stochastic input domain, and (2) how to control the variance of Monte Carlo estimator for highly "nonuniform" stochastic variations of the integrand in the prescribed input domain. These two difficulties are becoming increasingly important as the input space dimensionality increases. One can imagine a Monte Carlo integration scheme as a sort of a random fractional factorial sampling scheme or random quadrature that uses a refined cartesian grid of the discretized multidimensional input domain with many unfilled nodes with data, vs. the deterministic integration schemes or deterministic quadratures that can be viewed as complete factorial sampling schemes using cartesian grids with fully filled nodes with data. Thus, a very important aspect to get good results while using Monte Carlo is to ensure as much as possible a *uniform filling* of the input space domain grid with statistically independent solution points so that the potential local subspaces that can be important are not missed. The hurting problem of the standard Monte Carlo method is that for high-dimensional spaces, it is difficult to get uniform filling of the input space. The consequence of a nonuniform filling can be a sharp increase in the variance of the Monte Carlo estimator, which severely reduces the attraction for the method.

The classical way to improve the application of Monte Carlo methods to high-dimensional problems is to partition the stochastic input space in subdomains with uniform statistical properties. Instead of dealing with the entire stochastic space, we deal with subdomains. This is like defining a much coarser grid in space to work with. The space decomposition in subdomains may accelerate substantially the statistical convergence for multidimensional problems. Stratified sampling and importance sampling schemes are different sampling weighting schemes based on stochastic space decomposition in partitions. Using space decomposition we are able to control the variation of the estimator variance over the input domain by focusing on the most important stochastic space regions (mostly contributing to the integrand estimate) and adding more random data points in those regions.

Another useful way to deal with the high dimensionality of stochastic space is to simulate correlated random samples, instead of statistically independent samples, that describe a random path or walk in stochastic space. If the random walk has a higher attraction toward the regions with larger probability masses, then the time spent in a region vs. the time (number of steps) spent in another region is proportional to the probability mass distributions in those regions. By withdrawing samples at equal time intervals (number of steps), we can build an approximation of the prescribed joint probability distribution. There are two distinct classes of Monte Carlo methods for simulating random samples from a prescribed multivariate probability distribution: (1) static Monte Carlo (SMC), which is based on the generation of independent samples from univariate marginal densities or conditional densities; and (2) dynamic Monte Carlo (DMC), which is based on spatially correlated samples that describe the random evolutions of a *fictitious* stochastic dynamic system that has a stationary probability distribution identical

with the prescribed joint probability distribution. Both classes of Monte Carlo methods are discussed herein.

This chapter focuses on the application of Monte Carlo methods to simulate random samples of a variety of stochastic models — from the simplest stochastic models described by one-dimensional random variable models, to the most complex stochastic models described by hierarchical stochastic network-like models. The chapter also includes a solid section on high-dimensional simulation problems that touches key numerical issues including sequential sampling, dynamic sampling, computation of expectations and probabilities, modeling uncertainties, and sampling using stochastic partial-differential equation solutions. The goal of this chapter is to provide readers with the conceptual understanding of different simulation methods without trying to overwhelm them with all the implementation details that can be found elsewhere as cited in the text.

20.2 One-Dimensional Random Variables

There are a large number of simulation methods available to generate random variables with various probability distributions. Only the most popular of these methods are reviewed in this chapter: (1) inverse probability transformation, (2) sampling acceptance-rejection, (3) bivariate functional transformation, and (4) density decomposition. The description of these methods can be found in many textbooks on Monte Carlo simulations [1–5]. Herein, the intention is to discuss the basic concepts of these methods and very briefly describe their numerical implementations. In addition to these classical simulation methods, there is a myriad of many other methods, most of them very specific to particular types of problems, that have been developed over the past decades but for obvious reasons are not included.

20.2.1 Uniform Random Numbers

The starting point for any stochastic simulation technique is the construction of a reliable uniform random number generator. Typically, the random generators produce random numbers that are uniformly distributed in the interval [0, 1]. The most common uniform random generators are based on the linear congruential method [1–5].

The quality of the uniform random number generator reflects on the quality of the simulation results. It is one of the key factors that needs special attention. The quality of a random generator is measured by (1) the uniformity in filling with samples the prescribed interval and (2) its period, which is defined by the number of samples after which the random generator restarts the same sequence of independent random numbers all over again. Uniformity is a crucial aspect for producing accurate simulations, while the generator periodicity is important only when a large volume of samples is needed. If the volume of samples exceeds the periodicity of the random number generator, then the sequence of numbers generated after this is perfectly correlated with initial sequence of numbers, instead of being statistically independent. There are also other serial correlation types that can be produced by different number generators [4]. If we need a volume of samples for simulating a rare random event that is several times larger than the generator periodicity, then the expected simulation results will be wrong due to the serial correlation of the generated random numbers.

An efficient way to improve the quality of the generated sequence of uniform numbers and break their serial correlation is to combine the outputs of few generators into a single uniform random number that will have a much larger sequence periodicity and expectedly better space uniformity. As general advice to the reader, it is always a good idea to use statistical testing to check the quality of the random number generator.

20.2.2 Inverse Probability Transformation

Inverse probability transformation (IPT) is based on a lemma that says that for any random variable x with the probability distribution F, if we define a new random variable y with the probability distribution F^{-1} (i.e., the inverse probability transformation of F), then $y = F^{-1}(u)$ has a probability distribution F.

The argument u of the inverse probability transformation stands for a uniformly distributed random variable.

The general algorithm of the IPT method for any type of probability distribution is as follows:

Step 1: Initialize the uniform random number generator.
Step 2: Implement the algorithm for computing F^{-1}.
Step 3: Generate a uniform random number u in the interval [0, 1].
Step 4: Compute the generated deviate x with the distribution F by computing $x = F^{-1}(u)$.

Because the probability or cumulative distribution functions (CDFS) are monotonic increasing functions, the IPT method can be applied to any type of probability distribution, continuous or discrete, analytically or numerically defined, with any probability density function (PDF) shapes, from symmetric to extremely skewed, from bell-shaped to multimodal-shaped.

Very importantly, IPT can be also used to transform a sequence of Gaussian random deviates into a sequence of nonGaussian random deviates for any arbitrarily shaped probability distribution. A generalization of the IPT method, called the transformation method, is provided by Press et al. [4].

20.2.3 Sampling Acceptance-Rejection

Sampling acceptance-rejection (SAR) uses an auxiliary, trial, or proposal density to generate a random variable with a prescribed distribution. SAR is based on the following lemma: If x is a random variable with the PDF $f(x)$ and y is another variable with the PDF $g(y)$ and the probability of $f(y) = 0$, then the variable x can generated by first generating variable y and assuming that there is a positive constant α such as $\alpha \geq f(x)/g(x)$. If u is a uniform random variable defined on [0, 1], then the PDF of y conditioned on relationship $0 \leq u \leq f(y)/\alpha g(y)$ is identical with $f(x)$.

A typical numerical implementation of the SAR method for simulating a variable x with the PDF $f(x)$ is as follows:

Step 1: Initialize the uniform random number generator and select constant α.
Step 2: Generate a uniform number u on [0, 1] and a random sample of y.
Step 3: If $u > f(y)/\alpha g(y)$, then reject the pair (u, y) and go back to the previous step.
Step 4: Otherwise, accept the pair (u, y) and compute $x = y$.

It should be noted that SAR can be easily extended to generate random vectors and matrices with independent components.

The technical literature includes a variety of numerical implementations based on the principle of SAR [6]. Some of the newer implementations are the weighted resampling and sampling importance resampling (SIR) schemes [6, 7] and other adaptive rejection sampling [8].

20.2.4 Bivariate Transformation

Bivariate transformation (BT) can be viewed as a generalization of the IPT method for bivariate probability distributions. A well-known application of BT is the Box-Muller method for simulating standard Gaussian random variables [9]. Details are provided in [4–6].

20.2.5 Density Decomposition

Density decomposition (DD) exploits the fact that an arbitrary PDF, $f(x)$, can be closely approximated in a linear combination elementary density function as follows:

$$f(x) = \sum_{i=1}^{N} p_i g_i(x) \tag{20.1}$$

in which $0 \leq p_i < 1$, $1 \leq i \leq N$, and $\sum_{i=1}^{N} p_i = 1$.

The typical implementation of DD for a discrete variable x is as follows:

Step 1: Initialize for the generation of a uniform random number and a variable z_i with the PDF $g_i(x)$. This implies that $x = z_i$ with the probability p_i.
Step 2: Compute the values $G_i(x) = \sum_{j=1}^{i} p_j$ for all i values, $i = 1, N$.
Step 3: Generate a uniform random variable u on [0, 1].
Step 4: Loop $j = 1, N$ and check if $u < G_i(x)$. If not, go to the next step.
Step 5: Generate z_j and assign $x = z_j$.

It should be noted that the speed of the above algorithm is highest when the p_i values are ordered in descending order. It should be noted that DD can be combined with SAR. An example of such a combination is the Butcher method [10] for simulating normal deviates.

20.3 Stochastic Vectors with Correlated Components

In this section the numerical simulation of stochastic vectors with independent and correlated components is discussed. Both Gaussian and nonGaussian vectors are considered.

20.3.1 Gaussian Vectors

A multivariate Gaussian stochastic vector $\mathbf{x}$ of size m with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ is completely described by the following multidimensional joint probability density function (JPDF):

$$f_x(\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{m}{2}} (\det \Sigma)^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right] \tag{20.2}$$

The probability distribution of vector $\mathbf{x}$ is usually denoted as $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. If the Gaussian vector is a standard Gaussian vector, $\mathbf{z}$, with a zero mean, $\mathbf{0}$, and a covariance matrix equal to an identity matrix, $\mathbf{I}$, then its probability distribution is $N(\mathbf{0}, \mathbf{I})$. The JPDF of vector $\mathbf{z}$ is defined by:

$$f_z(\mathbf{0}, \mathbf{I}) = \frac{1}{(2\pi)^{\frac{m}{2}} (\det \Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{z}^T\mathbf{z}\right) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z_i^2\right) \tag{20.3}$$

To simulate a random sample of a Gaussian stochastic vector $\mathbf{x}$ (with correlated components) that has the probability distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, a two computational step procedure is needed:

Step 1: Simulate a standard Gaussian vector $\mathbf{z}$ (with independent random components) having a probability distribution $N(\mathbf{0}, \mathbf{I})$. This step can be achieved using standard routines for simulating independent normal random variables.
Step 2: To compute the vector $\mathbf{x}$ use the linear matrix equation

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{S}\mathbf{z} \tag{20.4}$$

where matrix $\mathbf{S}$ is called the *square-root matrix* of the positively definite covariance matrix $\boldsymbol{\Sigma}$. The matrix $\mathbf{S}$ is a lower triangular matrix that is computed using the Choleski decomposition that is defined by the equation $\mathbf{S}\mathbf{S}^T = \boldsymbol{\Sigma}$.

Another situation of practical interest is to generate a Gaussian vector $\mathbf{y}$ with a probability distribution $N(\mu_y, \Sigma_{yy})$ that is conditioned on an input Gaussian vector $\mathbf{x}$ with a probability distribution $N(\mu_x, \Sigma_{xx})$. Assuming that the augmented vector $[\mathbf{x}, \mathbf{y}]^T$ has the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ defined by

$$\mu = \begin{pmatrix} \mu_y \\ \mu_x \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{pmatrix} \tag{20.5}$$

then the statistics of the conditional vector **y** are computed by the matrix equations:

$$\begin{aligned} \mu_y &= \mu_y + \Sigma_{yx} \cdot \Sigma_{xx}^{-1}(\mathbf{x} - \mu_x) \\ \Sigma_{yy} &= \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy} \end{aligned} \tag{20.6}$$

It is interesting to note that the above relationships define a stochastic condensation procedure of reducing the size of a stochastic vector from the size of the total vector $[\mathbf{y}, \mathbf{x}]^T$ with probability distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to a size of the reduced vector **y** with a probability distribution $N(\mu_y, \Sigma_{yy})$. The above simulation procedures employed for Gaussian vectors can be extended to nonGaussian vectors, as shown in the next subsection.

An alternate technique to the Choleski decomposition for simulating stochastic vectors with correlated components is the popular principal component analysis. Principal component analysis (PCA) is based on eigen decomposition of the covariance matrix. If PCA is used, then the matrix equation in the Step 2 shown above is replaced by the following matrix equation

$$\mathbf{x} = \mu + \Phi\lambda^{1/2}\mathbf{z} \tag{20.7}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\Phi}$ are the eigenvalue and eigenvector matrices, respectively, of the covariance matrix.

20.3.2 NonGaussian Vectors

A nonGaussian stochastic vector is completely defined by its JPDF. However, most often in practice, a nonGaussian vector is only partially defined by its second-order moments (i.e., the mean vector and the covariance matrix) and its marginal probability distribution (MCDF) vector. This definition loses information about the high-order statistical moments that are not defined.

These partially defined nonGaussian vectors form a special class of nonGaussian vectors called translation vectors. Thus, the translation vectors are nonGaussian vectors defined by their second-order moments and their marginal distributions. Although the translation vectors are not completely defined stochastic vectors, they are of great practicality. First, because in practice, most often we have statistical information limited to second-order moments and marginal probability distributions. Second, because they capture the most critical nonGaussian aspects that are most significantly reflected in the marginal distributions. Of great practical benefit is that translation vectors can be easily mapped into Gaussian vectors, and therefore can be easily handled and programmed.

To generate a nonGaussian (translation) stochastic vector **y** with a given covariance matrix $\boldsymbol{\Sigma}_{yy}$ and a marginal distribution vector **F**, first a Gaussian image vector **x** with a covariance matrix $\boldsymbol{\Sigma}_{xx}$ and marginal distribution vector $\boldsymbol{\Phi}(\mathbf{x})$ is simulated. Then, the original nonGaussian vector **y** is simulated by applying IPT to the MCDF vector **F** as follows:

$$\mathbf{y} = \mathbf{F}^{-1}\Phi(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \tag{20.8}$$

However, for simulating the Gaussian image vector we need to define its covariance matrix $\boldsymbol{\Sigma}_{xx}$ as a transform of the covariance matrix $\boldsymbol{\Sigma}_{yy}$ of the original nonGaussian vector. Between the elements of the scaled covariance matrix or correlation coefficient matrix of the original nonGaussian vector, $\rho_{yi,yj}$, and the elements of the scaled covariance or correlation coefficient of the Gaussian image vector, $\rho_{xi,xj}$, there is the following relation:

$$\rho_{yi,yj} = \frac{1}{\sigma_{yi}\sigma_{yj}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[F_i^{-1}\Phi(x_i) - \mu_{yi}\right]\left[F_i^{-1}\Phi(x_j) - \mu_{yj}\right]\phi(x_i, x_j)\, dx_i dx_j \tag{20.9}$$

where the bivariate Gaussian probability density is defined by:

$$\phi(x_i, x_j) = \frac{1}{2\pi\left(1-\rho_{xi,xj}^2\right)^{0.5}\sigma_{xi}\sigma_{xj}} \exp\left(-\frac{1}{2}\frac{(x_i-\mu_{xi})^2/\sigma_{xi}^2 - 2\rho_{xi,xj}/\sigma_{xi}\sigma_{xj} + (x_j-\mu_{xj})^2/\sigma_{xj}^2}{\left(1-\rho_{xi,xj}^2\right)}\right) \quad (20.10)$$

Two probability transformation options are attractive: (1) the components of the Gaussian image stochastic vector have means and variances equal to those of the original nonGaussian vector (i.e., $\mu_y = \mu_x$ and $\sigma_y = \sigma_x$), or (2) the Gaussian image vector is standard Gaussian vector (i.e., $\mu_x = \mathbf{0}$ and $\boldsymbol{\sigma}_x = \mathbf{1}$). Depending on the selected option, the above equations take a simpler form.

Thus, problem of generating nonGaussian vectors is a four-step procedure for which the two steps are identical with those used for generating a Gaussian vector:

Step 1: Compute the covariance matrix of Gaussian vector **x** using Equation 20.9.
Step 2: Generate a standard Gaussian vector **z.**
Step 3: Generate a Gaussian vector **x** using Equation 20.4 or Equation 20.7.
Step 4: Simulate a nonGaussian vector **y** using the following inverse probability transformation of the MCDF, $\mathbf{y} = \mathbf{F}^{-1}\Phi(\mathbf{x})$.

Sometimes in practice, the covariance matrix transformation is neglected, assuming that the correlation coefficient matrix of the original nonGaussian vector and that of the image Gaussian vector are identical. This assumption is wrong and may produce a violation of the physics of the problem. It should be noted that calculations of correlation coefficients have indicated that if $\rho_{xi,xj}$ is equal to 0 or 1, then $\rho_{yi,yj}$ is also equal to 0 or 1, respectively. However, when $\rho_{xi,xj}$ is –1, then $\rho_{yi,yj}$ is not necessarily equal to –1. For significant negative correlations between vector components, the effects of covariance matrix transformation are becoming significant, especially for nonGaussian vectors that have skewed marginal PDF shapes. Unfortunately, this is not always appreciated in the engineering literature. There are a number of journal articles on probabilistic applications for which the significance of covariance matrix transformation is underestimated; for example, the application of the popular Nataf probability model to nonGaussian vectors with correlated components neglects the covariance matrix transformation. Grigoriu [11] showed a simple example of a bivariate lognormal distribution for which the lowest value of the correlation coefficient is –0.65, and not –1.00, which is the lowest value correlation coefficient for a bivariate Gaussian distribution.

For a particular situation, of a nonGaussian component y_i and a Gaussian component x_j, the correlation coefficient $\rho_{yi,xj}$ can be computed by

$$\rho_{yi,xj} = \frac{1}{\sigma_{yi}\sigma_{xj}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} [F^{-1}\Phi(x_i) - \mu_{yi}](x_j - \mu_{xj})\phi(x_i, x_j)dx_i dx_j \quad (20.11)$$

As shown in the next section, the above probability transformation simulation procedure can be extended from nonGaussian vectors to multivariate nonGaussian stochastic fields. The nonGaussian stochastic fields that are partially defined by their mean and covariance functions and marginal distributions are called translation stochastic fields.

Another way to generate nonGaussian stochastic vectors is based on the application of the SAR method described for one-dimensional random variables to random vectors with correlated components. This application of SAR to multivariate cases is the basis of the Metropolis-Hastings algorithm that is extremely popular in the Bayesian statistics community. Importantly, the Metropolis-Hastings algorithm [12] is not limited to translation vectors. The basic idea is to sample directly from the JPDF of the nonGaussian vector using an adaptive and important sampling strategy based on the application of SAR. At each

simulation step, random samples are generated from a simple proposal or trial JPDF that is different than the target JPDF and then weighted in accordance with the important ratio. This produces dependent samples that represent a Markov chain random walk in the input space. The Metropolis-Hastings algorithm is the cornerstone of the Markov chain Monte Carlo (MCMC) simulation discussed in Section 20.5 of this chapter.

To move from vector sample or state $\mathbf{x}^t$ to state $\mathbf{x}^{t+1}$, the following steps are applied:

Step 1: Sample from the trial density, $\mathbf{y} \approx q(\mathbf{x}^t, \mathbf{y})$, that is initially assumed to be identical to the conditional density $q(\mathbf{y}|\mathbf{x}^t)$.

Step 2: Compute acceptance probability

$$\alpha(\mathbf{x}^t, \mathbf{y}) = \min\left(1, \frac{\pi(\mathbf{y})q(\mathbf{y}, \mathbf{x}^t)}{\pi(\mathbf{x}^t)q(\mathbf{x}^t, \mathbf{y})}\right) \tag{20.12}$$

where π is the target JPDF.

Step 3: Compute $\mathbf{x}^{t+1} = \mathbf{y}$ for sampling acceptance, otherwise $\mathbf{x}^{t+1} = \mathbf{x}^t$ for sampling rejection.

Based on the ergodicity of the simulated Markov chain, the random samples of the target JPDF are simulated by executing repeated draws at an equal number of steps from the Markov chain movement in the stochastic space. There is also the possibility to use multiple Markov chains simultaneously.

A competing algorithm with the Metropolis-Hastings algorithm is the so-called Gibbs sampler [13]. The Gibbs sampler assumes that the probability of sampling rejection is zero; that is, all samples are accepted. This makes it simpler but less flexible when compared with the Metropolis-Hastings algorithm. One step of the Gibbs sampler that moves the chain from state $\mathbf{x}^t$ to state $\mathbf{x}^{t+1}$ involves the following simulation steps by breaking the vector $\mathbf{x}$ in its k components:

$$\begin{aligned}
&\text{Step 1: Sample } x_1^{t+1} \approx \pi\left(x_1 | x_2^t, \ldots, x_k^t\right).\\
&\text{Step 2: Sample } x_2^{t+1} \approx \pi\left(x_2 | x_1^{t+1}, x_3^t, \ldots, x_k^t\right).\\
&\quad\vdots\\
&\text{Step j: Sample } x_j^{t+1} \approx \pi\left(x_j | x_1^{t+1}, \ldots, x_{j-1}^{t+1}, x_{j+1}^t, \ldots, x_k^t\right).\\
&\quad\vdots\\
&\text{Step k: Sample } x_k^{t+1} \approx \pi\left(x_k | x_1^{t+1}, \ldots, x_{k-1}^{t+1}\right).
\end{aligned} \tag{20.13}$$

To ensure an accurate vector simulation for both the Metropolis-Hastings and the Gibbs algorithms, it is important to check the ergodicity of the generated chain, especially if they can be stuck in a local energy minimum. Gibbs sampler is the most susceptible to getting stuck in different parameter space regions (metastable states).

Most MCMC researchers are currently using derivative of the Metropolis-Hastings algorithms. When implementing the Gibbs sampler or Metropolis-Hastings algorithms, key questions arise: (1) How do we need to block components to account for the correlation structure and dimensionality of the target distribution? (2) How do we choose the updating scanning strategy: deterministic or random? (3) How do we devise the proposal densities? (4) How do we carry out convergence and its diagnostics on one or more realizations of the Markov chain?

20.4 Stochastic Fields (or Processes)

A stochastic process or field is completely defined by its JPDF. Typically, the term "stochastic process" is used particularly in conjunction with the time evolution of a dynamic random phenomenon, while the term "stochastic field" is used in conjunction with the spatial variation of a stochastic surface. A space-time stochastic process is a stochastic function having time and space as independent arguments. The term "space-time stochastic process" is synonymous with the term "time-varying stochastic field." More generally, a stochastic function is the output of a complex physical stochastic system. Because a stochastic output can be described by a stochastic surface in terms of input parameters (for given

ranges of variability), it appears that the term "stochastic field" is a more appropriate term for stochastic function approximation. Thus, *stochastic field* fits well with stochastic boundary value problems. *Stochastic process* fits well with stochastic dynamic, phenomena, random vibration, especially for stochastic stationary (steady state) problems that assume an infinite time axis. The term "stochastic field" is used hereafter.

Usually, in advanced engineering applications, simplistic stochastic models are used for idealizing component stochastic loading, material properties, manufacturing geometry and assembly deviations. To simplify the stochastic modeling, it is often assumed that the *shape* of spatial random variations is deterministic. Thus, the spatial variability is reduced to a single random variable problem, specifically to a random scale factor applied to a deterministic spatial shape. Another simplified stochastic model that has been extensively used in practice is the traditional response surface method based on quadratic regression and experimental design rules (such as circumscribed central composite design, CCCD, or Box-Benken design, BBD). The response surface method imposes a global quadratic trend surface for approximating stochastic spatial variations that might violate the physics of the problem. However, the traditional response surface method is practical for mildly nonlinear stochastic problems with a reduced number of random parameters.

From the point of view of a design engineer, the simplification of stochastic modeling is highly desired. A design engineer would like to keep his stochastic modeling as simple as possible so he can understand it and simulate it with a good confidence level (obviously, this confidence is subjective and depends on the analyst's background and experience). Therefore, the key question of the design engineer is: Do I need to use stochastic field models for random variations, or can I use simpler models, random variable models? The answer is yes and no on a case-by-case basis. Obviously, if by simplifying the stochastic modeling the design engineer significantly violates the physics behind the stochastic variability, then he has no choice; he has to use refined stochastic field models. For example, stochastic field modeling is important for turbine vibration applications due to the fact that blade mode-localization and flutter phenomena can occur. These blade vibration-related phenomena are extremely sensitive to small spatial variations in blade properties or geometry produced by the manufacturing or assembly process. Another example of the need for using a refined stochastic field modeling is the seismic analysis of large-span bridges. For large-span bridges, the effects of the nonsynchronicity and spatial variation of incident seismic waves on structural stresses can be very significant. To capture these effects, we need to simulate the earthquake ground motion as a dynamic stochastic field or, equivalently, by a space-time stochastic process as described later in this section.

A stochastic field can be homogeneous or nonhomogeneous, isotropic or anisotropic, depending on whether its statistics are invariant or variant to the axis translation and, respectively, invariant or variant to the axis rotation in the physical parameter space. Depending on the complexity of the physics described by the stochastic field, the stochastic modeling assumptions can affect negligibly or severely the simulated solutions. Also, for complex problems, if the entire set of stochastic input and stochastic system parameters is considered, then the dimensionality of the stochastic space spanned by the stochastic field model can be extremely large.

This chapter describes two important classes of stochastic simulation techniques of continuous multivariate stochastic fields (or stochastic functionals) that can be successfully used in advanced engineering applications. Both classes of simulation techniques are based on the decomposition of the stochastic field in a set of elementary uncorrelated stochastic functions or variables. The most desirable situation from an engineering perspective is to be able to simulate the original stochastic field using a reduced number of elementary stochastic functions or variables. The dimensionality reduction of the stochastic input is extremely beneficial because it also reduces the overall dimensionality of the engineering reliability problem.

This chapter focuses on stochastic field simulation models that use a limited number of elementary stochastic functions, also called *stochastic reduced-order models*. Many other popular stochastic simulation techniques, used especially in conjunction with random signal processing — such as discrete autoregressive process models AR, moving-average models MA, or combined ARMA or ARIMA models, Gabor

transform models, wavelet transform models, and many others — are not included due to space limitation. This is not to shadow their merit.

In this chapter two important types of stochastic simulation models are described:

1. *One-level hierarchical stochastic field (or stochastic functional) model*. This simulation model is based on an *explicit* representation of a stochastic field. This representation is based on a statistical function (causal relationship) approximation by nonlinear regression. Thus, the stochastic field is approximated by a stochastic hypersurface **u** that is conditioned on the stochastic input **x**. The typical explicit representation of a stochastic field has the following form:

$$\mathbf{u}|\mathbf{x} = \mathbf{u}(\mathbf{x}) = \boldsymbol{\mu}_{u|x} + [\mathbf{u}(\mathbf{x}) - \boldsymbol{\mu}_{u|x}] \tag{20.14}$$

   In the above equation, first the conditional mean term, $\boldsymbol{\mu}_{u|x}$, is computed by minimizing the global mean-square error over the sample space. Then, the randomly fluctuating term $[\mathbf{u}(\mathbf{x}) - \boldsymbol{\mu}_{u|x}]$ is treated as a zero-mean decomposable stochastic field that can be factorized using a Wiener-Fourier series representation. This type of stochastic approximation, based on regression, is limited to a convergence in mean-square sense.

   In the traditional response surface method, the series expansion term is limited to a single term defined by a stochastic residual vector defined directly in the original stochastic space (the vector components are the differences between exact and mean values determined at selected sampling points via experimental design rules).

2. *Two-level hierarchical stochastic field (or stochastic functional) model.* This simulation model is based on an *implicit* representation of a stochastic field. This representation is based on the Joint Probability Density Function (JPDF) (non-causal relationship) estimation. Thus, the stochastic field **u** is described by the JPDF of an augmented stochastic system $[\mathbf{x}, \mathbf{u}]^T$ that includes both the stochastic input **x** and the stochastic field **u**. The augmented stochastic system is completely defined by its JPDF $f(\mathbf{x}, \mathbf{u})$. This JPDF defines implicitly the stochastic field correlation structure of the field. Then, the conditional PDF of the stochastic field $f(\mathbf{u}|\mathbf{x})$ can be computed using the JPDF of augmented system and the JPDF of the stochastic input $f(\mathbf{x})$ as follows:

$$f(\mathbf{u}|\mathbf{x}) = f(\mathbf{x}, \mathbf{u}) / f(\mathbf{x}) \tag{20.15}$$

The JPDF of the augmented stochastic system can be conveniently computed using its projections onto a stochastic space defined by a set of locally defined, overlapping JPDF models. The set of local or conditional JPDF describe completely the local structure of the stochastic field. This type of stochastic approximation is based on the joint density estimation convergences in probability sense.

There are few key aspects that differentiate the two stochastic field simulation models. The one-level hierarchical model is based on statistical function estimation that is a more restrictive approximation problem than the density estimation employed by the two-level hierarchical model. Statistical function estimation based on regression can fail when the stochastic field projection on the input space is not convex. Density estimation is a much more general estimation problem than a statistical function estimation problem. Density estimation is always well-conditioned because there is no causal relationship implied.

The two-level hierarchical model uses the local density functions to approximate a stochastic field. The optimal solution rests between the use of a large number of small-sized isotropic-structure density functions (with no correlation structure) and a reduced number of large-sized anisotropic-structure density functions (with strong correlation structure). The preference is for a reduced number of local density functions. Cross-validation or Bayesian inference techniques can be used to select the optimal stochastic models based either on error minimization or likelihood maximization.

20.4.1 One-Level Hierarchical Simulation Models

To simulate complex pattern nonGaussian stochastic fields (functionals), it is advantageous to represent them using a Wiener-Fourier type series [14–16]:

$$\mathbf{u}(\mathbf{x}, \theta) = \sum_{i=0}^{\infty} \mathbf{u}_i(\mathbf{x}) \mathrm{f}_i(\theta) = \sum_{i=0}^{\infty} \mathbf{u}_i(\mathbf{x}) \mathbf{f}_i(\mathbf{z}(\theta)) \tag{20.16}$$

where argument **z** is a set of independent standard Gaussian random variables and **f** is a set of orthogonal basis functions (that can be expressed in terms of a random variable set **z**). A simple choice is to take the set of stochastic basis functions **f** equal to the random variables set **z**. There are two main disadvantages when using the Wiener-Fourier series approximations. First, the stochastic orthogonal basis functions f_i are multidimensional functions that require intensive computations, especially for high-dimensional stochastic problems for which a large number of coupling terms need to be included to achieve adequate convergence of the series.

Generally, under certain integrability conditions, a stochastic function can be decomposed in an orthogonal structure in a probability measure space. Assuming that the uncorrelated stochastic variables z_i, $i = 1, 2, \ldots, m$ are defined on a probability space and that $u(z_1, \ldots, z_m)$ is a square-integrable function with respect to the probability measure and if $\{p_{i,k}(z_i)\}$, $i = 1, 2, \ldots, m$ are complete sets of square-integrable stochastic functions orthogonal with respect to the probability density $P_i(dz_i)/dz_i = f_z(z_i)$, so that $E[p_k(z_i)p_l(z_i)] = 0$ for all $k \neq 1 = 0, 1, \ldots$ and $i = 1, \ldots m$, then, the function $u(z_1, \ldots, z_m)$ can be expanded in a generalized Wiener-Fourier series:

$$u(z_1, \ldots z_m) = \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} u_{k_1 \ldots k_m} p_{k_1}(z_1) \ldots p_{k_m}(z_m) \tag{20.17}$$

where $\{p_{k_1}(z_1), \ldots, p_{k_m}(z_m)\}, k_1, \ldots, k_m = 0, 1, \ldots$, are complete sets of stochastic orthogonal (uncorrelated) functions. The generalized Wiener-Fourier series coefficients can be computed by solving the integral

$$u_{k_1 \ldots k_m} = \int \ldots \int u(z_1, \ldots, z_m) p_{k_1}(z_1) \ldots p_{k_m}(z_m)\ u_1(z_1) \ldots u_m(z_m) dz_1 \ldots dz_m \tag{20.18}$$

The coefficients $u_{k_1 \ldots k_m}$ have a key minimizing property, specifically the integral difference

$$D = \int \ldots \int \left[u(z_1, \ldots, z_m) - \sum_{k_1=0}^{M_1} \cdots \sum_{k_m=0}^{M_m} g_{k_1 \ldots k_m} p_{k_m}(z_m) \right]^2 f_z(z_1) \ldots f_z(z_m) dz_1 \ldots dz_m \tag{20.19}$$

reaches its minimum only for $g_{k_1 \ldots k_m} p_{k_m} = u_{k_1 \ldots k_m} p_{k_m}$.

Several factorization techniques can be used for simulation of complex pattern stochastic fields. An example is the use of the Pearson differential equation for defining different types of stochastic series representations based on orthogonal Hermite, Legendre, Laguerre, and Cebyshev polynomials. These polynomial expansions are usually called Askey chaos series [16]. A major application of stochastic field decomposition theory is the spectral representation of stochastic fields using covariance kernel factorization. These covariance-based techniques have a large potential for engineering applications because they can be applied to any complex, static, or dynamic nonGaussian stochastic field. Herein, in addition to covariance-based factorization techniques, an Askey polynomial chaos series model based on Wiener-Hermite stochastic polynomials is presented. This polynomial chaos model based on Wiener-Hermite series has been extensively used by many researchers over the past decade [15–17].

20.4.1.1 Covariance-Based Simulation Models

Basically, there are two competing simulation techniques using the covariance kernel factorization: (1) the Choleski decomposition technique (Equation 20.4), and (2) the Karhunen-Loeve (KL) expansion (Equation 20.7). They can be employed to simulate both static and dynamic stochastic fields. A notable property of these two simulation techniques is that they can handle both real-valued and complex-valued covariance kernels. For simulating space-time processes (or dynamic stochastic fields), the two covariance-based techniques can be employed either in the time-space domain by decomposing the cross-covariance kernel or in the complex frequency-wavelength domain by decomposing the complex cross-spectral density kernel. For real-valued covariance kernels, the application of the KL expansion technique is equivalent to the application of the Proper Orthogonal Decomposition (POD expansion) and Principal Component Analysis (PCA expansion) techniques [17, 18].

More generally, the Choleski decomposition and the KL expansion can be applied to any arbitrary square-integrable, complex-valued stochastic field, $u(\mathbf{x}, \theta)$. Because the covariance kernel of the complex-valued stochastic field $\mathrm{Cov}[u(\mathbf{x}, \theta), u(\mathbf{x}', \theta)]$ is a Hermitian kernel, it can be factorized using either Choleski or KL decomposition.

If the KL expansion is used, the covariance function is expanded in the following eigenseries:

$$\mathrm{Cov}[u(\mathbf{x}, \theta), u(\mathbf{x}', \theta)] = \sum_{n=0}^{\infty} \lambda_n \Phi_n(\mathbf{x}) \Phi_n(\mathbf{x}') \tag{20.20}$$

where λ_n and $\Phi_n(\mathbf{x})$ are the eigenvalue and the eigenvector, respectively, of the covariance kernel computed by solving the integral equation (based on Mercer's theorem) [19]:

$$\int \mathrm{Cov}[u(\mathbf{x}, \theta), u(\mathbf{x}', \theta)] \Phi_n(\mathbf{x}) d\mathbf{x} = \Phi_n(\mathbf{x}') \tag{20.21}$$

As a result of covariance function being Hermitian, all its eigenvalues are real and the associated complex eigenfunctions that correspond to distinct eigenvalues are mutually orthogonal. Thus, they form a complete set spanning the stochastic space that contains the field u. It can be shown that if this deterministic function set is used to represent the stochastic field, then the stochastic coefficients used in the expansion are also mutually orthogonal (uncorrelated).

The KL series expansion has the general form

$$u(\mathbf{x}, \theta) = \sum_{i=0}^{n} \sqrt{\lambda_i} \Phi_i(\mathbf{x}) z_i(\theta) \tag{20.22}$$

where set $\{z_i\}$ represents the set of uncorrelated random variables that are computed by solving the stochastic integral:

$$z_i(\theta) = \frac{1}{\sqrt{\lambda_i}} \int_D \Phi_n(\mathbf{x})\, u(\mathbf{x}, \theta)\, d\mathbf{x} \tag{20.23}$$

The KL expansion is an optimal spectral representation with respect to the second-order statistics of the stochastic field. Equation 20.23 indicates that the KL expansion can be applied also to nonGaussian stochastic fields if sample data from it are available. For many engineering applications on continuum mechanics, the KL expansion is fast mean-square convergent; that is, only a few expansion terms need to be included.

An important practicality aspect of the above covariance-based simulation techniques is that they can be easily applied in conjunction with the marginal probability transformation (Equation 20.8 and Equation 20.9)

to simulate nonGaussian (translation) stochastic fields, either static or dynamic. For nonGaussian (translation) stochastic fields, two simulation models can be employed:

1. *Original space expansion*. Perform the simulation in the original nonGaussian space using the covariance-based expansion with a set $\{z_i\}$ of uncorrelated nonGaussian variables that can be computed as generalized Fourier coefficients by the integral

$$z_i(\theta) = \int_D u_i(\mathbf{x}) u(\mathbf{x}, \theta)\, d\mathbf{x} \tag{20.24}$$

   In particular, for the KL expansion, the Equation 20.23 is used.

2. *Transformed space expansion.* Perform the simulation in the transformed Gaussian space using the covariance-based expansion with a set $\{z_i\}$ of standard Gaussian variables and then transform the Gaussian field to nonGaussian using the marginal probability transformation (Equation 20.8 and Equation 20.9).

In the engineering literature there are many examples of the application of covariance-based expansions for simulating either static or dynamic stochastic fields. In the civil engineering field, Choleski decomposition and KL expansion were used by several researchers, including Yamazaki and Shinozuka [20], Deodatis [21], Deodatis and Shinozuka [22], and Ghiocel [23] and Ghiocel and Ghanem [24], to simulate the random spatial variation of soil properties and earthquake ground motions. Shinozuka [25] and Ghiocel and Trandafir [26] used Choleski decomposition to simulate stochastic spatial variation of wind velocity during storms. Ghiocel and Ghiocel [27, 28] employed the KL expansion to simulate the stochastic wind fluctuating pressure field on large-diameter cooling towers using a nonhomogeneous, anisotropic dynamic stochastic field model. In the aerospace engineering field, Romanovski [29], and Thomas, Dowell, and Hall [30] used the KL expansion (or POD) to simulate the unsteady pressure field on aircraft jet engine blades. Ghiocel [31, 32] used the KL expansion as a stochastic classifier for jet engine vibration-based fault diagnostics.

An important aerospace industry application is the stochastic simulation of the engine blade geometry deviations due to manufacturing and assembly processes. These random manufacturing deviations have a complex stochastic variation pattern [33]. The blade thickness can significantly affect the forced response of rotating bladed disks [34–36]. Due to the cyclic symmetry geometry configuration of engine bladed disks, small manufacturing deviations in blade geometries produce a mode-localization phenomenon, specifically called mistuning, that can increase the airfoil vibratory stresses up to a few times. Blair and Annis [37] used the Choleski decomposition to simulate blade thickness variations, assuming that the thickness variation is a homogeneous and isotropic field. Other researchers, including Ghiocel [38, 39], Griffiths and Tschopp [40], Cassenti [41], and Brown and Grandhi [35], have used the KL expansion (or POD, PCA) to simulate more complex stochastic blade thickness variations due to manufacturing. Ghiocel [36] applied the KL expansion, both in the original, nonGaussian stochastic space and transformed Gaussian space. It should be noted that the blade thickness variation fields are highly nonhomogeneous and exhibit multiple, localized anisotropic directions due to the manufacturing process constraints (these stochastic variations are also technology dependent). If different blade manufacturing technology datasets are included in the same in a single database, then the resultant stochastic blade thickness variations could be highly nonGaussian, with multimodal, leptoqurtique, and platiqurtique marginal PDF shapes. More generally, for modeling the blade geometry variations in space, Ghiocel suggested a 3V-3D stochastic field model (three variables $\Delta x, \Delta y, \Delta z$, in three dimensions x, y, z). For only blade thickness variation, a 1V-2D stochastic field (one variable, thickness, in two dimensions, blade surface grid) is sufficient. It should be noted that for multivariate-multidimensional stochastic fields composed of several elementary one-dimensional component fields, the KL expansion must be applied to the entire covariance matrix of the stochastic field set that includes the coupling of all the component fields.

One key advantage of the KL expansion over the Choleski decomposition, that makes the KL expansion (or POD, PCA) more attractive for practice, is that the transformed stochastic space obtained using the

KL expansion typically has a highly reduced dimensionality when compared with the original stochastic space. In contrast, the Choleski decomposition preserves the original stochastic space dimensionality. For this reason, some researchers [29, 30] consider the KL expansion (or POD, PCA) a *stochastic reduced-order model* for simulating complex stochastic patterns. The KL expansion, in addition to space dimensionality reduction, also provides great insight into the stochastic field structure. The eigenvectors of the covariance matrix play, in stochastic modeling, a role similar to the vibration eigenvectors in structural dynamics; complex spatial variation patterns are decomposed in just a few dominant spatial variation *mode shapes*. For example, blade thickness variation *mode shapes* provide great insight into the effects of technological process variability. These insights are very valuable for improving blade manufacturing technology.

The remaining part of this subsection illustrates the application of covariance-based stochastic simulation techniques to generate dynamic stochastic fields. Specifically, the covariance-based methods are used to simulate the stochastic earthquake ground surface motion at a given site.

To simulate the earthquake ground surface motion at a site, a nonstationary, nonhomogeneous stochastic vector process model is required. For illustrative purposes, it is assumed that the stochastic process is a nonhomogeneous, nonstationary, 1V-2D space-time Gaussian process (one variable, acceleration in an arbitrary horizontal direction, and two dimensions for the horizontal ground surface). The space-time process at any time moment is completely defined by its evolutionary cross-spectral density kernel. For a seismic acceleration field $u(\mathbf{x}, t)$, the cross-spectral density for two motion locations i and k is:

$$\mathbf{S}_{ui,uk}(\omega, t) = [\mathbf{S}_{ui,ui}(\omega, t)\mathbf{S}_{uk,uk}(\omega, t)]^{1/2} Coh_{ui,uk}(\omega, t) \quad \exp[-i\omega(X_{D,i} - X_{D,k})/V_D(t)] \tag{20.25}$$

where $\mathbf{S}_{uj,uk}(\omega)$ is the cross-spectral density function for point motions u_i and u_k, and $\mathbf{S}_{uj,uj}(\omega)$, $j = i$, k is the auto-spectral density for location point j. The function $Coh_{ui,uk}(\omega, t)$ is the stationary or "lagged" coherence function for locations i and k. The "lagged" coherence is a measure of the similarity of the two point motions including only the amplitude spatial variation. Herein, it is assumed that the frequency-dependent spatial correlation structure of the stochastic process is time-invariant. The exponential factor $\exp[i\omega(D_1 - D_2)/V_D(t)]$ represents the wave passage effect in the direction D expressed in the frequency domain by a phase angle due to two motion delays at two locations X_i and X_j. The parameter $V_D(t)$ is the apparent horizontal wave velocity in the D direction. Most often in practice, the nonstationary stochastic models of ground motions are based on the assumption that the "lagged" coherence and the apparent directional velocity are independent of time. However, in real earthquakes, the coherence and directional wave velocity are varying during the earthquake duration, depending on the time arrivals of different seismic wave packages hitting the site from various directions.

Because the cross-spectral density is Hermitian, either the Choleski decomposition or the KL expansion can be applied. If the Choleski decomposition is applied, then

$$\mathbf{S}(\omega, t) = \mathbf{C}(\omega, t)\mathbf{C}^*(\omega, t) \tag{20.26}$$

where the matrix $\mathbf{C}(\omega, t)$ is a complex-valued lower triangular matrix. Then the space-time nonstationary stochastic process can be simulated using the trigonometric series as the number of frequency components $NF \to \infty$ [21,22]:

$$u_i(t) = 2\sum_{k=1}^{NL}\sum_{j=1}^{NF}\left|C_{i,k}(\omega_j, t)\right|\sqrt{\Delta\omega}\cos[\omega_j t - \theta_{i,k}(\omega_j, t) + \Phi_{k,j}], \quad \text{for } i = 1, 2, \ldots NL \tag{20.27}$$

In the above equation, NL is the number of space locations describing the spatial variation of the motion. The first phase angle term in Equation 20.27 is computed by

$$\theta_{i,k}(\omega, t) = \tan^{-1}\left\{\frac{\mathrm{Im}\,|C_{i,k}(\omega, t)|}{\mathrm{Re}\,|C_{i,k}(\omega, t)|}\right\} \tag{20.28}$$

and the second phase angle term $\mathbf{\Phi}_{k,j}$ is a random phase angle uniformly distributed in [0, 2π] (uniform random distribution is consistent with Gaussian assumption). The above procedure can be used for any space-time stochastic process. For nonGaussian processes, the procedure can be applied in conjunction with the inverse probability equation transformation (Equation 20.8 and Equation 20.9).

If the complex-valued coherence function (including wave passage effects) is used, then its eigenfunctions are complex functions. Calculations have shown that typically one to five coherence function modes are needed to get an accurate simulation of the process [23, 24]. The number of needed coherence function modes depends mainly on the soil layering stiffness and the frequency range of interest; for high-frequency components, a larger number of coherence modes are needed.

20.4.1.2 Polynomial Chaos Series-Based Simulation Models

Ghanem and Spanos [15] discussed theoretical aspects and key details of the application Polynomial Chaos to various problems in their monograph on spectral representation of stochastic functionals.

Polynomial chaos expansion models can be formally expressed as a nonlinear functional of a set of standard Gaussian variables or, in other words, expanded in a set of stochastic orthogonal polynomial functions. The most popular polynomial chaos series model is that proposed by Ghanem and Spanos [15] using a Wiener-Hermite polynomial series:

$$u(\mathbf{x}, t, \theta) = a_0(\mathbf{x}, t)\Gamma_0 + \sum_{i_1=1}^{\infty} a_{i_1}(\mathbf{x}, t)\Gamma_1(z_{i_1}(\theta)) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} a_{i_1 i_2}(\mathbf{x}, t)\Gamma_2(z_{i_1}(\theta), z_{i_2}(\theta)) + \ \ldots \tag{20.29}$$

The symbol $\Gamma_n(z_{i_1}, \ldots, z_{i_n})$ denotes the polynomial chaoses of order n in the variables $(z_{i_1}, \ldots, z_{i_n})$. Introducing a one-to-one mapping to a set with ordered indices denoted by $\{\psi_i(\theta)\}$ and truncating the polynomial chaos expansion after the p^{th} term, Equation 20.29 can be rewritten

$$u(\mathbf{x}, t, \theta) = \sum_{j=0}^{p} u_j(\mathbf{x}, t)\psi_j(\theta) \tag{20.30}$$

The polynomial expansion functions are orthogonal in L_2 sense that is, their inner product with respect to the Gaussian measure that defines their statistical correlation, $E[\psi_j\psi_k]$, is zero. A given truncated series can be refined along the random dimension either by adding more random variables to the set $\{z_i\}$ or by increasing the maximum order of polynomials included in the stochastic expansion. The first refinement takes into account higher frequency random fluctuations of the underlying stochastic process, while the second refinement captures strong nonlinear dependence of the solution process on this underlying process. Using the orthogonality property of polynomial chaoses, the coefficients of the stochastic expansion solution can be computed by

$$u_k = \frac{E[\psi_k u]}{E[\psi_k^2]} \quad \text{for } k = 1, \ldots, K \tag{20.31}$$

A method for constructing polynomial chaoses of order n is by generating the corresponding multidimensional Wiener-Hermite polynomials. These polynomials can be generated using the partial differential recurrence rule defined by

$$\frac{\partial}{\partial z_{ij}}\Gamma_n(z_{i1}(\theta), \ldots, z_{in}(\theta)) = n\Gamma_{n-1}(z_{i1}(\theta), \ldots, z_{in}(\theta)) \tag{20.32}$$

The orthogonality of the polynomial chaoses is expressed by the inner product in L_2 sense with respect to Gaussian measure:

$$\int_{-\infty}^{\infty} \Gamma_n(z_{i1}, \ldots, z_{in})\Gamma_m(z_{i1}, \ldots, z_{in}) \exp\left(-\frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{z}\right) dz = n!\sqrt{2\pi}\,\delta_{mn} \tag{20.33}$$

Although popular in the engineering community, the polynomial chaos series may not necessarily be an efficient computational tool to approximate multivariate nonGaussian stochastic fields. The major problem is the stochastic dimensionality issue. Sometimes, the dimensionality of the stochastic transformed space defined by a polynomial chaos basis can be even larger than the dimensionality of the original stochastic space. Nair [42] discussed this stochastic dimensionality aspect related to polynomial chaos series application. Grigoriu [43] showed that the indiscriminate use of polynomial chaos approximations for stochastic simulation can result in inaccurate reliability estimates. To improve its convergence, the polynomial chaos series can be applied in conjunction with the inverse marginal probability transformation (Equation 20.8 and Equation 20.9) as suggested by Ghiocel and Ghanem [24].

20.4.2 Two-Level Hierarchical Simulation Models

The two-level hierarchical simulation model is based on the decomposition of the JPDF of the implicit input-output stochastic system in local or conditional JPDF models. The resulting stochastic local JPDF expansion can be used to describe, in detail, very complex nonstationary, multivariate-multidimensional nonGaussian stochastic fields. It should be noted that the local JPDF expansion is convergent in a probability sense, in contrast with the Wiener-Fourier expansions, which are convergent only in a mean-square sense.

The stochastic local JPDF basis expansion, or briefly, the local density expansion, can be also viewed as a Wiener-Fourier series of a composite type that provides both a global approximation and a local functional description of the stochastic field. In contrast to the classical Wiener-Fourier series that provides a global representation of a stochastic field by employing global basis functions, as the covariance kernel eigenfunctions in KL expansion or the polynomial chaoses [16], the stochastic local density expansion provides both a global and local representation of a stochastic field.

Stochastic local density expansion can be implemented as a two-layer stochastic neural-network, while Wiener-Fourier series can be implemented as a one-layer stochastic neural-network. It should be noticed that stochastic neural-network models train much faster than the usual multilayer preceptor (MLP) neural-networks. Figure 20.1 describes in pictorial format the analogy between stochastic local JPDF expansion and a two-layer neural-network model.

In the local density expansion, the overall JPDF of the stochastic model is obtained by integration over the local JPDF model space:

$$g(\mathbf{u}) = \int f(\mathbf{u}|\alpha)dp(\alpha) \tag{20.34}$$

where $p(\alpha)$ is a continuous distribution that plays the role of the probability weighting function over the local model space. In a discrete form, the weighting function can be expressed for a number N of local JPDF models by

$$p(\alpha) = \sum_{i=1}^{N} P(\alpha_i)\delta(\alpha - \alpha_i) \tag{20.35}$$

in which $\delta(\alpha - \alpha_i)$ is the Kronecker delta operator. Typically, the parameters α_i are assumed or known, and the discrete weighting parameters $P(\alpha_i)$ are the unknowns. The overall JPDF of the stochastic model can be computed in the discrete form by

$$g(\mathbf{u}) = \sum_{i=1}^{N} g(\mathbf{u}|\alpha_i)P(\alpha_i) \tag{20.36}$$

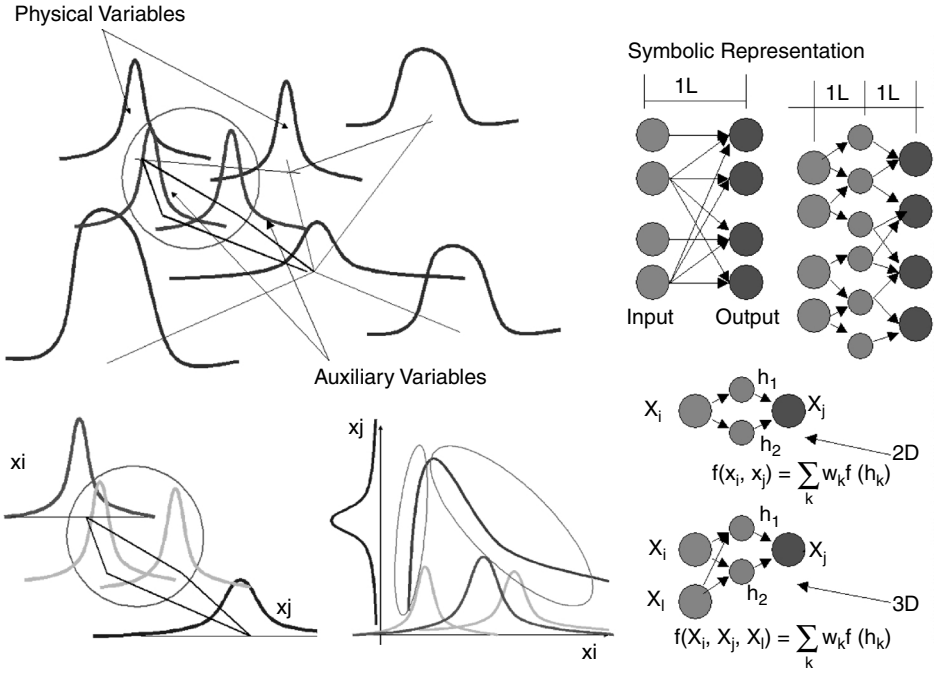

FIGURE 20.1 Analogy betweeen the local JPDF expansion and a two-layer stochastic neural network model.

The parameters α_i can be represented by the second-order statistics (computed by local averaging) of the local JPDF models. Thus, the overall JPDF expression can be rewritten as

$$g(\mathbf{u}) = \sum_{i=1}^{N} f(\mathbf{u} \mid i, \bar{\mathbf{u}}_i, \Sigma_i) P_i \tag{20.37}$$

where $\bar{\mathbf{u}}_i, \Sigma_i$ are the mean vector and covariance matrix, respectively, of the local JPDF model i. Also, $P_i = n_i/N, \sum_{i=1}^{N} P_i = 1$, and $P_i > 0$, for $i = 1, N$. Typically, the types of the local JPDF are assumed and the probability weights are computed from the sample datasets. Often, it is assumed that the local JPDFs are multivariate Gaussian models. This assumption implies that the stochastic field is described locally by a second-order stochastic field model. Thus, nonGaussianity is assumed only globally. It is possible to include nonGaussianity locally using the marginal probability transformation of the local JPDF.

The local JPDF models are typically defined on partially overlapping partitions within stochastic spaces called *soft* partitions. Each local JPDF contributes to the overall JPDF estimate in a localized convex region of the input space. The complexity of the stochastic field model is associated with the number of local JPDF models and the number of stochastic inputs that define the stochastic space dimensionality. Thus, the stochastic model complexity is defined by the number of local JPDFs times the number of input variables. For most applications, the model implementations that are based on a large number of highly localized JPDFs are not good choices because these models have high complexity. For high complexities, we need very informative data to build the locally refined stochastic models. Simpler models with a lesser number of JPDF models have a much faster statistical convergence and are more robust. Thus, the desire is to reduce model complexity at an optimal level.

The complexity reduction of the stochastic field model can be accomplished either by reducing the number of local JPDFs or by reducing their dimensionality. Complexity reduction can be achieved using cross-validation and then removing the local JPDF that produces the minimum variation of the error estimate or the model likelihood estimate. We can also use penalty functions (evidence functions) that

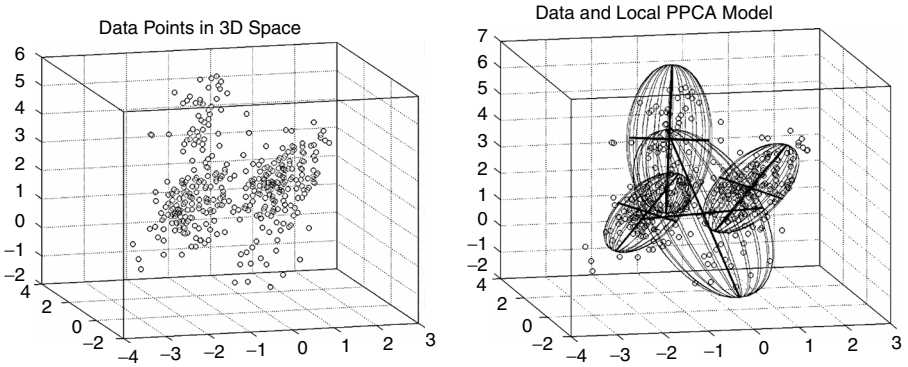

FIGURE 20.2 Sample data and local PPCA models for a trivariate stochastic field.

reduce the model complexity by selecting the most plausible stochastic models based on the evidence coming from sample datasets [44].

Alternatively, we can reduce the model complexity by reducing the dimensionality of local JPDF models using local factor analyzers. Local factor analysis reduces the number of local models by replacing the local variables in the original space by a reduced number of new local variables defined in local transformed spaces using a stochastic linear space projection. If deterministic projection instead of stochastic projection is applied, the factor analysis coincides with PCA. However, in practice, especially for high-dimensional problems, it is useful to consider a stochastic space projection by adding some random noise to the deterministic space projection. Adding noise is beneficial because instead of solving a multidimensional eigen-problem for decomposing the covariance kernel, we can implement an extremely fast convergent iterative algorithm for computing the principal covariance directions. The combination of the local JPDF expansion combined with the local probabilistic PCA decomposition is called herein the local PPCA expansion. Figure 20.2 shows the simulated sample data and the local PPCA models for a complex pattern trivariate stochastic field.

It should be noted that if the correlation structure of local JPDF models is ignored, these local JPDFs lose their functionality to a significant extent. If the correlation is neglected, a loss of information occurs, as shown in Figure 20.3. The radial basis functions, such standard Gaussians, or other kernels ignore the local correlation structure due to their shape constraint. By this they lose accuracy for

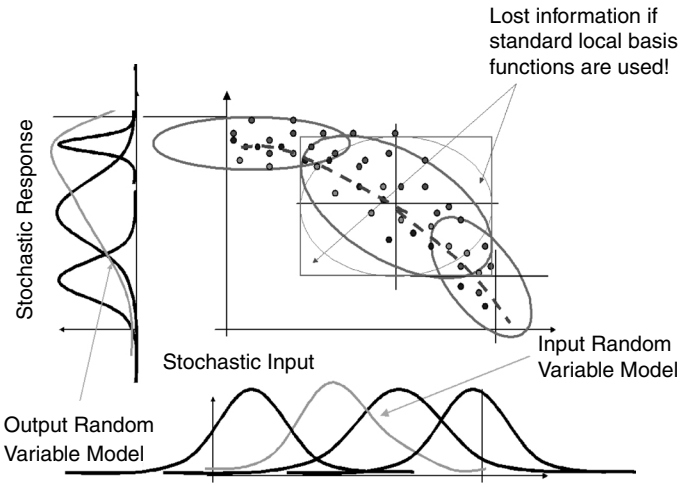

FIGURE 20.3 Loss of information if the local correlation structure is ignored.

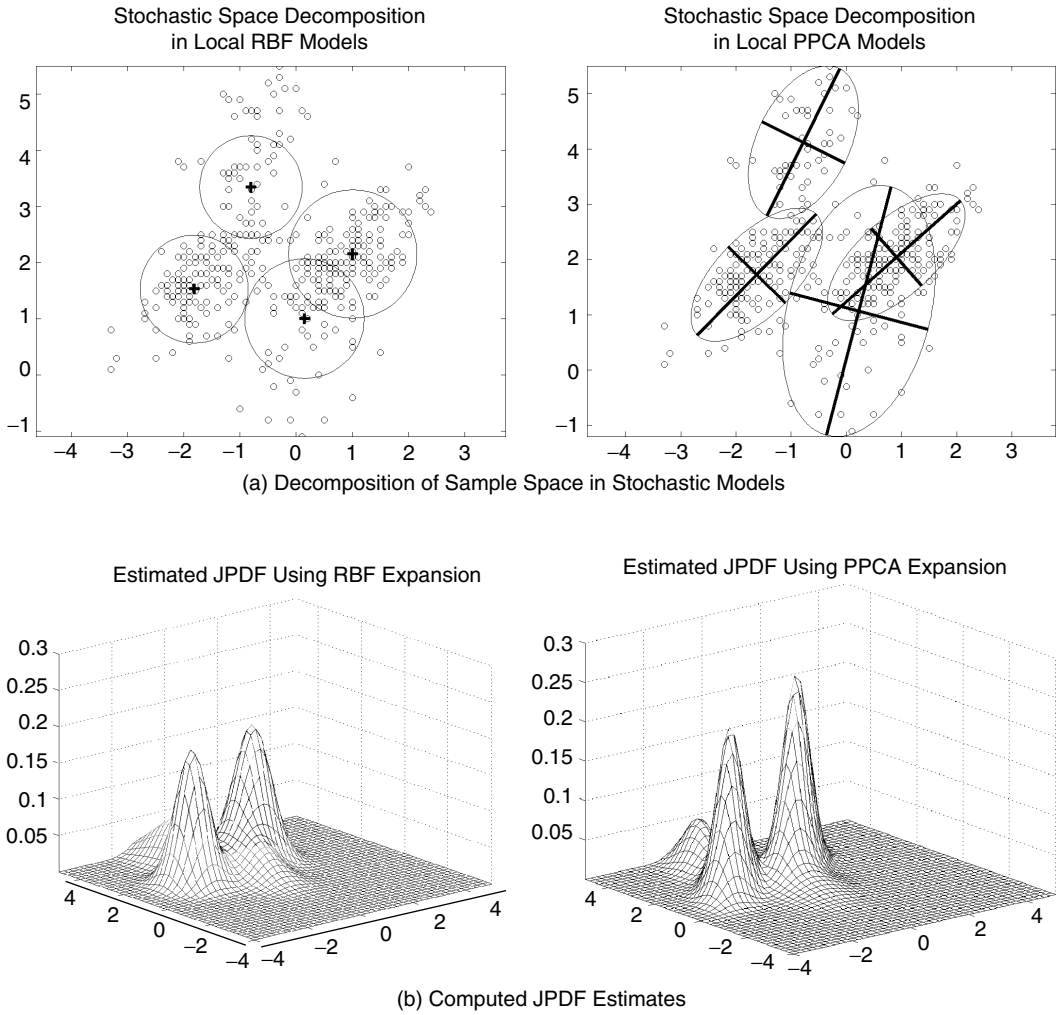

FIGURE 20.4 Local PPCA expansion model versus RBF network model. a) decomposition of sample space in local stochastic models b) computed JPDF estimates.

modeling a complex stochastic field. One way to see radial basis function expansions is to look at them as degenerated local JPDF models. These radial functions are commonly used in a variety of practical applications, often in the form of the radial basis function network (RBFN), fuzzy basis function network (FBFN), and adaptive network-based fuzzy inference systems (ANFIS). Figure 20.4 compares the RBFN model with the local PPCA expansion model. The local PPCA expansion model is clearly more accurate and better captures all the data samples, including those that are remote in the tails of the JPDF. For a given accuracy, the local PPCA expansion needs a much smaller number of local models than the radial basis function model. As shown in the figure, the local PPCA models can elongate along long data point clouds in any arbitrary direction. The radial basis functions do not have this flexibility. Thus, to model a long data point cloud, a large number of overlapping radial basis functions are needed. And even so, the numerical accuracy of radial basis function expansions may still not be as good as the accuracy of local PPCA expansions with just few arbitrarily oriented local point cloud models. It should be noted that many other popular statistical approximation models, such as Bayesian Belief Nets (BNN), Specht's Probability Neural Network (PNN), in addition to ANFIS, RBF, and FBF networks, and even the two-layer MLP networks, have some visible conceptual similarities with the more refined local PPCA expansion.

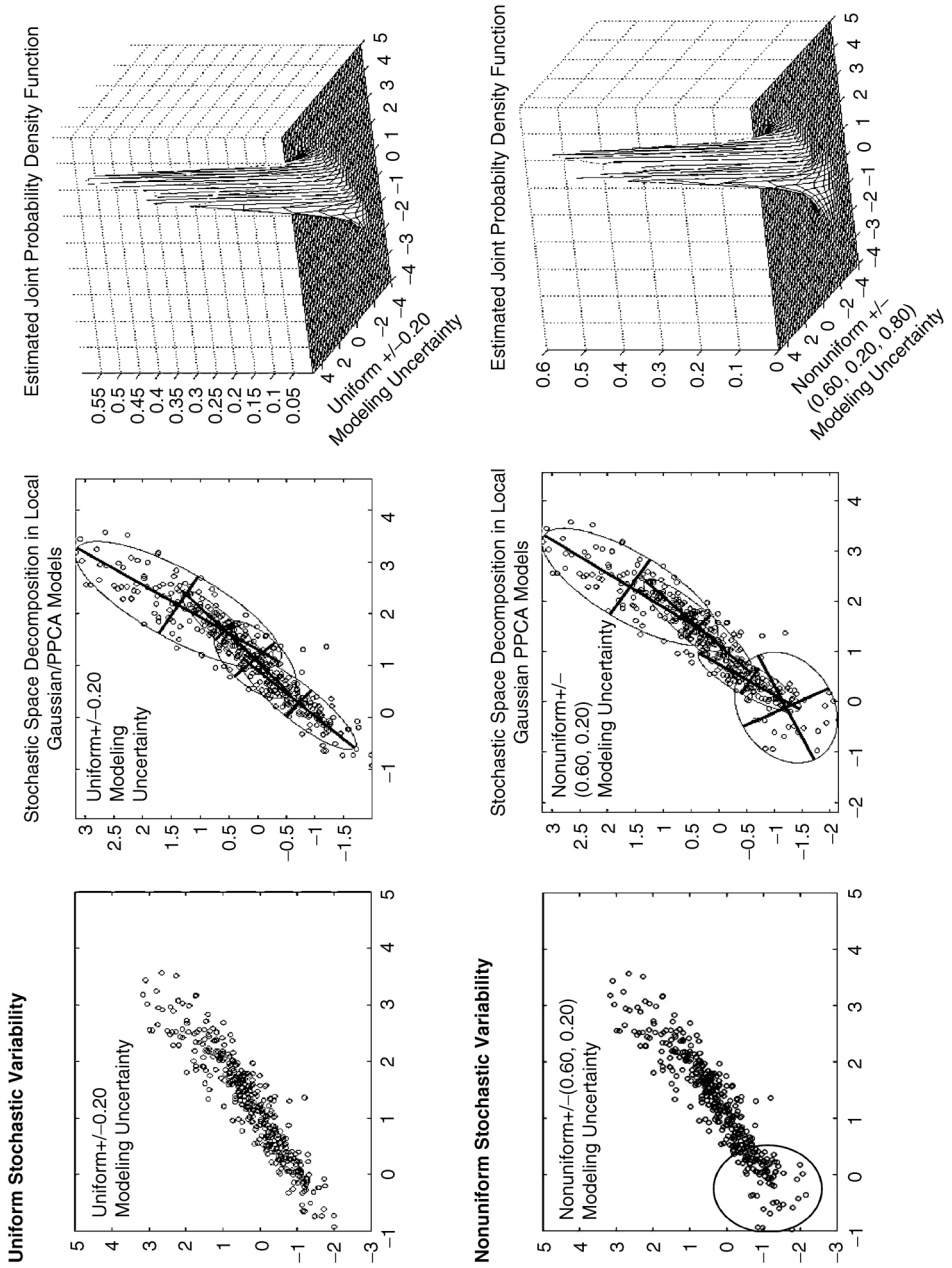

FIGURE 20.5 Local PPCA expansion model captures the nonuniform stochastic variability; a) Initial dataset, uniform data, b) Locally perturbed dataset, nonuniform data.

The stochastic local PPCA expansion is also capable of including accurately local, nonuniform stochastic variations. Figure 20.5 illustrates this fact by comparing the local PPCA expansions for two sample datasets that are identical except for a local domain shown at the bottom left of the plots. For the first dataset (left plots), the local variation was much smaller than for the second dataset (right plots). As seen by comparing the plots, the local PPCA expansion captures well this highly local variability of probabilistic response.

The decomposition of stochastic parameter space in a set of local models defined on soft partitions offers several possibilities for simulating high-dimensional complex stochastic fields. This stochastic sample space decomposition in a reduced number of stochastic local models, also called *states*, opens up a unique avenue for developing powerful stochastic simulation techniques based on stochastic process or field theory, such as dynamic Monte Carlo. Dynamic Monte Carlo techniques are discussed in the next section.

20.5 Simulation in High-Dimensional Stochastic Spaces

In simulating random samples from a joint probability density $g(\mathbf{x})$, in principle there are always ways to do it because we can relate the multidimensional joint density $g(\mathbf{x})$ to univariate probability densities defined by full conditional densities $g(x_j|x_M, \ldots, x_{j+1}, x_{j-1}, \ldots, x_1)$:

$$g(\mathbf{x}) = g(x_1, \ldots, x_K) = \prod_{j=1}^{M} g(x_j|x_{j-1}, \ldots, x_1) \tag{20.38}$$

Unfortunately, in real-life applications, these (univariate) full conditional densities are seldom available in an explicit form that is suitable to direct sampling. Thus, to simulate a random sample from $g(\mathbf{x})$, we have few choices: (1) use independent random samples drawn from univariate marginal densities (static, independent sampling, SMC) to build the full conditionals via some predictive models; or (2) use independent samples from trial univariate marginal and conditional densities (sequential importance sampling, SIS), defined by the probability chain rule decomposition of the joint density, and then adaptively weight them using recursion schemes; or (3) use independent samples drawn from the univariate full conditional densities to produce spatially dependent samples (trajectories) of the joint density using recursion schemes that reflect the stochastic system dynamics (dynamic, correlated sampling, DMC).

For multidimensional stochastic fields, the standard SMC sampling techniques can fail to provide good estimators because their variances can increase sharply with the input space dimensionality. This variance increase depends on stochastic simulation model complexity. For this reason, variation techniques are required. However, classical techniques for variance reductions such as stratified sampling, importance sampling, directional sampling, control variable, antithetic variable, and Rao-Blackwellization methods are difficult to apply in high dimensions.

For simulating complex pattern, high-dimensional stochastic fields, the most adequate techniques are the adaptive importance sampling techniques, such as the sequential importance sampling with independent samples, SIS, or with correlated samples, DMC. Currently, in the technical literature there exists a myriad of recently developed adaptive, sequential, single sample-based or population-based evolutionary algorithms for high-dimensional simulations [6, 45]. Many of these techniques are based on constructing a Markov chain structure of the stochastic field. The fundamental Markov assumption of one-step memory significantly reduces the chain structures of the full or partial conditional densities, thus making the computations affordable. In addition, Markov process theory is well-developed so that reliable convergence criteria can be implemented.

20.5.1 Sequential Importance Sampling (SIS)

SIS is a class of variance reduction techniques based on adaptive important sampling schemes [45, 46]. A useful strategy to simulate a sample of joint distribution is to build up a trial density sequentially until it converges to the target distribution. Assuming a trial density constructed as $g(\mathbf{x}) = g_1(x_1)g_2(x_2|x_1) \ldots g_N(x_N|x_1, \ldots x_{N-1})$

and using the same decomposition for the target density $\pi(\mathbf{x}) = \pi(x_1)\pi(x_2|x_1) \ldots \pi(x_N|x_1, \ldots x_{N-1})$, the importance sampling weight can be computed recursively by

$$w_t(\mathbf{x}_t) = w_{t-1}(\mathbf{x}_{t-1})\frac{\pi(\mathbf{x}_t|\mathbf{x}_{t-1})}{g_t(\mathbf{x}_t|\mathbf{x}_{t-1})} \tag{20.39}$$

An efficient SIS algorithm is based on the following recursive computation steps, using an improved adaptive sampling weight [6, 46]:

Step 1: Simulate $\mathbf{x}_t$ from $g_t(\mathbf{x}_t|\mathbf{x}_{t-1})$, let $\mathbf{x}_t = (\mathbf{x}_{t-1}, \mathbf{x}_t)$
Step 2: Compute the adaptive sampling weight by

$$w_t = w_{t-1}\frac{\pi_t(\mathbf{x}_t)}{\pi_{t-1}(\mathbf{x}_{t-1})g_t(\mathbf{x}_t|\mathbf{x}_{t-1})} \tag{20.40}$$

The entire sample of the input **x** is adaptively weighted until the convergence of the trial density to the target density is reached.

An important application of the SIS simulation technique is the *nonlinear filtering* algorithm based on the linear state-space model that is very popular in the system dynamics and control engineering community. In this algorithm, the system dynamics consist of two major coupled parts: (1) the state equation, typically represented by a Markov process; and (2) the observation equations, which are usually written as

$$\begin{aligned} x_t &\sim q_t(.|x_{t-1}, \theta) \\ y_t &\sim f_t(.|x_t, \phi) \end{aligned} \tag{20.41}$$

where y_t are observed state variables and x_t are unobserved state variables. The distribution of x_t are computed using the recursion:

$$\pi_t(x_t) = \int q_t(x_t|x_{t-1}) f_t(y_t|x_t)\pi_{t-1}(x_{t-1})\, dx_{t-1} \tag{20.42}$$

In applications, the discrete version of this model is called the Hidden Markov model (HMM). If the conditional distributions q_t and f_t are Gaussian, the resulting model is the called the *linear dynamic model* and the solution can be obtained analytically via recursion and coincides with the popular Kalman filter solution.

To improve the statistical convergence of SIS, we can combine it with resampling techniques, sometimes called SIR, or with rejection control and marginalization techniques.

20.5.2 Dynamic Monte Carlo (DMC)

DMC simulates realizations of a stationary stochastic field (or process) that has a unique stationary probability density identical to the prescribed JPDF. Instead of sampling directly in the original cartesian space from marginal distributions, as SMC does, DMC generates a stochastic trajectory of spatially dependent samples, or a random walk in the state-space (or local model space). Figure 20.6 shows the basic conceptual differences between standard SMC and DMC. The Gibbs sampler and Metropolis-Hastings algorithms presented in Section 20.3.2 are the basic ingredients of dynamic Monte Carlo techniques.

The crucial part of the DMC is how to invent the best ergodic stochastic evolution for the underlying system that converges to the desired probability distribution. In practice, the most used DMC simulation

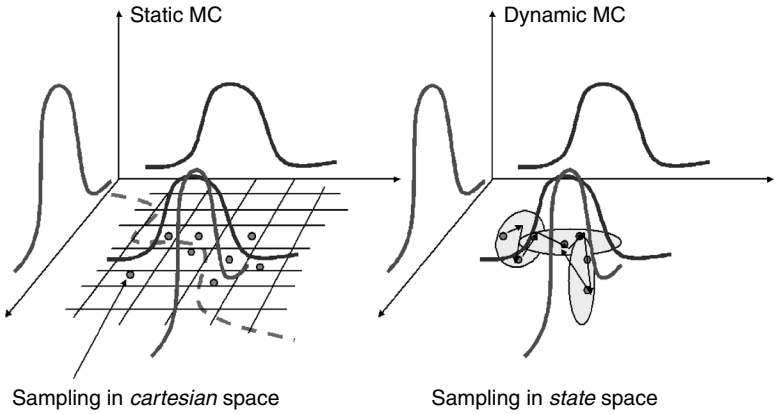

FIGURE 20.6 Conceptual differences between static Monte Carlo and dynamic Monte Carlo.

models are based on irreducible, aperiodic, reversible stationary Markov process models. The DMC simulation based on Markov process models is called Markov Chain Monte Carlo simulation, or briefly MCMC. Higher-order Markov chains or nonMarkovian are more difficult to implement. When using MCMC, a key modeling aspect is how to ensure that the samples of the chain are ergodic, especially when there is a chance that they can be stuck in a local energy minimum. The principal drawback of MCMC is that it has only local moves along its random path. This leads to a very slow convergence to equilibrium or, worse, to a stop in a local state.

A good strategy for improving MCMC movement in space is to combine the local powerful properties of MCMC with the global updating properties of molecular dynamics simulation using a Hamiltonian formulation of system dynamics. This combination is based on the Hamiltonian Markov Chain Monte Carlo (H-MCMC) algorithm [47]. The Hamiltonian MCMC is also called the Hybrid Monte Carlo (HMC). As an alternate to H-MCMC, an MCMC with random jumps has been developed [48].

Herein we describe briefly the H-MCMC algorithm. This algorithm is the preferred MCMC algorithm for high-dimensional stochastic simulation problems. The H-MCMC simulation algorithm can be efficiently implemented in conjunction with conditional PPCA expansion models. Hanson [47] claims in his research paper that "the efficiency of Hamiltonian method does not drop much with increasing dimensionality." Hanson has used H-MCMC for stochastic problems with a parameter space size up to 128 dimensions.

The H-MCMC algorithm is used to simulate a random sample of a canonical (or Gibbs, or Boltzmann) JPDF of the form:

$$\pi(\mathbf{x}) = c \text{ x } \exp(-H(\mathbf{x})) \tag{20.43}$$

This distribution is specific to Gibbs fields or equivalently to Markov fields [49]. In the above equation, $H(\mathbf{x})$ is the Hamiltonian of the dynamic system.

In the Hamiltonian formulation, the evolution of the dynamic system is completely known once the total energy or the Hamiltonian function in generalized coordinates, $H(\mathbf{q}, \mathbf{p})$, is defined:

$$H(\mathbf{q}, \mathbf{p}) = \sum_{i=1}^{n} \frac{p_i^2}{2m_i} + E(\mathbf{q}) \tag{20.44}$$

where **q** and **p** are the vectors of the generalized coordinates and generalized momenta (in the phase space), m_i are the masses associated with the individual particles of the system, and E(**q**) is the potential energy that ensures a coupling between the particles. System dynamics is represented by a trajectory in the phase space. For any arbitrary evolutionary function f(**q**, **p**), its rate of change is expressed by its time derivative:

$$\dot{f}(\mathbf{q},\mathbf{p}) = \sum_{i=1}^{n}\left(\frac{\partial f}{\partial q_i}\dot{q}_i + \frac{\partial f}{\partial p_i}\dot{p}_i\right) = \sum_{i=1}^{n}\left(\frac{\partial f}{\partial q_i}\frac{\partial H}{\partial p_i} - \frac{\partial f}{\partial p_i}\frac{\partial H}{\partial q_i}\right) \tag{20.45}$$

For stationary stochastic dynamics, the Hamiltonian conserves. Now, using the molecular dynamics (MD) approach, we can discretize the Hamiltonian function evolution using finite difference equations over the time chopped in small increments. The most used implementation is the leap-frog scheme [47]:

$$\begin{aligned} p_i\left(t+\frac{1}{2}\Delta t\right) &= p_i(t) - \frac{1}{2}\Delta t\frac{\partial E(q)}{\partial q_i(t)} q_i(t+\Delta t) = q_i(t) + \Delta t p_i\left(t+\frac{1}{2}\Delta t\right) p_i(t+\Delta t) \\ &= p_i\left(t+\frac{1}{2}\Delta t\right) - \frac{1}{2}\Delta t\frac{\partial E(\mathbf{q})}{\partial q_i(t+\Delta t)} \end{aligned} \tag{20.48}$$

The leap-frog scheme is time reversible and preserves the phase space volume. The resulting Hamiltonian-MCMC algorithm is capable of having trajectory jumps between different iso-energy surfaces and converges to a target canonical (or Gibbs, Boltzmann) distribution.

Figure 20.7 and Figure 20.8 show the application of H-MCMC to a simple bivariate stochastic field model composed of three overlapping local Gaussians. Figure 20.7 shows a simulated trajectory using the H-MCMC. The random samples were obtained by drawing sample data at each random walk step. Figure 20.8 compares the target JPDF and the estimated JPDF. The estimated JPDF is computed from the 250 random samples drawn at each step of the Markov chain evolution. It can be observed that there is a visible spatial correlation between the generated random samples. In practice, the random samples should be drawn periodically after a large number of random walk steps, so that the spatial correlation is lost. This improves considerably the convergence in probability of the H-MCMC simulation.

An extremely useful application of the DMC simulation is the stochastic interpolation for stochastic fields with missing data, especially for heavy-tailed nonGaussianity. DMC can be also used to simulate conditional-mean surfaces (response surfaces). Figure 20.9 illustrates the use of H-MCMC in conjunction with the local PPCA expansion for a univariate highly nonGaussian stochastic field with unknown mean. It should be noted that the quality of estimation depends on the data density in the sample space. Figure 20.10 shows another simple example of application of H-MCMC for simulating a univariate stochastic field with known mean using different persistence parameters (that define the magnitude of fictitious generalized momenta). The figure shows the H-MCMC samples and the exact (given) and estimated mean curves. By comparing the two plots in Figure 20.10, it should be noted that larger persistence parameters (larger generalized momenta) improve the simulation results for the mean curve in remote regions where data density is very low. Larger fictitious momenta produce larger system dynamics and, as a result, increase the variance and reduce the bias of the mean estimates. It appears that larger momenta in H-MCMC simulations are appropriate for probability estimations in the distribution tails of the joint distributions. Thus, DMC can be also employed to establish confidence bounds of stochastically interpolated surfaces for different probability levels. From the figure we can note that larger momenta provide less biased confidence interval estimates in the remote area of the distribution tails.

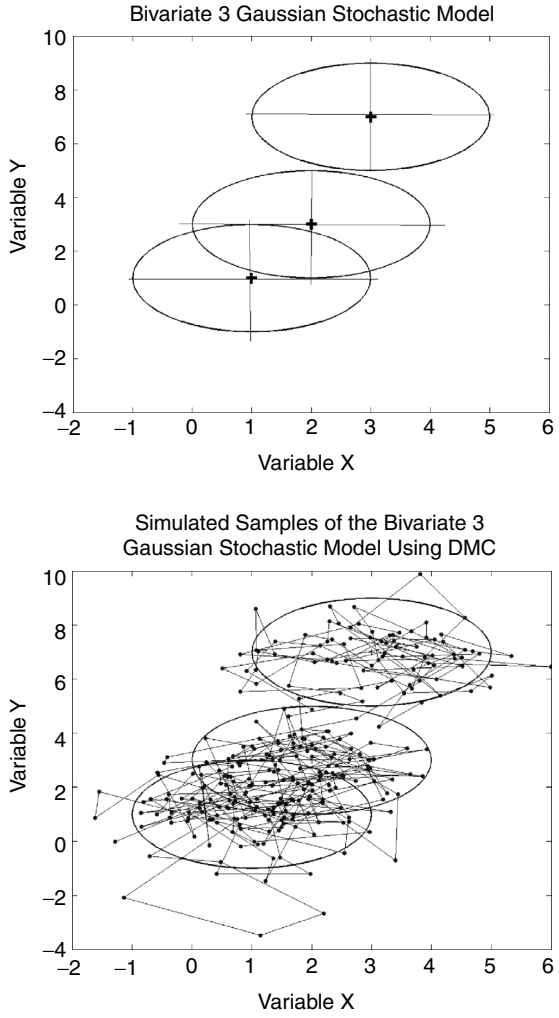

FIGURE 20.7 The prescribed stochastic model and the H-MCMC simulated random samples (drawn point data and random walk).

Other techniques to accelerate the MCMC convergence are simulated tempering and annealing. Simulated tempering and simulated annealing are popular MCMC applications of stochastic simulation to optimization under uncertainty problems. These techniques, which are described in many textbooks, are not discussed herein. This is not to shadow their merit.

20.5.3 Computing Tail Distribution Probabilities

As discussed in the introductory section of this chapter, stochastic simulation can be applied to either (1) simulate random quantities or (2) compute expectations. For probabilistic engineering analyses, both applications of stochastic simulation are of practical interest. For reliability and risk assessment analyses, the second application of simulation is used.

This section addresses the use of stochastic simulation to compute expectations. Only selective aspects are reviewed herein. The focus is on solving high-complexity stochastic problems.

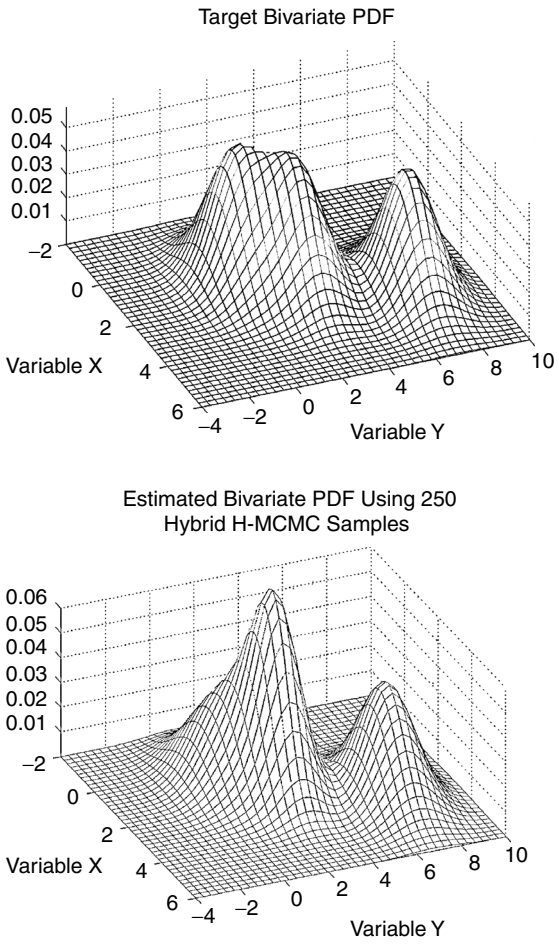

FIGURE 20.8 Target JPDF versus estimated JPDF (using 250 samples drawn at each step).

Expectations can include probability integrals for computing marginal distributions, evidences (normalization constant for the product of likelihood and prior in Bayes theorem), statistical moments, and expectation of functions. Thus, in most stochastic problems, the numerical aspect consists of computing the expected value of a function of interest f(**x**) with respect to a target probability density $g(\mathbf{x})$, where **x** is the stochastic input. Assuming that $x_1, x_2, \ldots, x_N$ are an independent identically distributed sample set from $g(\mathbf{x})$, then the Monte Carlo estimator $\hat{f}$ is computed by simple statistical averaging over the entire sample set:

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{n} f(x_i) \tag{20.49}$$

The estimator $\hat{f}$ is unbiased and has the variance

$$\text{var}(\hat{f}) = \frac{1}{N}\int [f(\mathbf{x}) - E_g(f)]^2 g(\mathbf{x})\, d\mathbf{x} \tag{20.50}$$

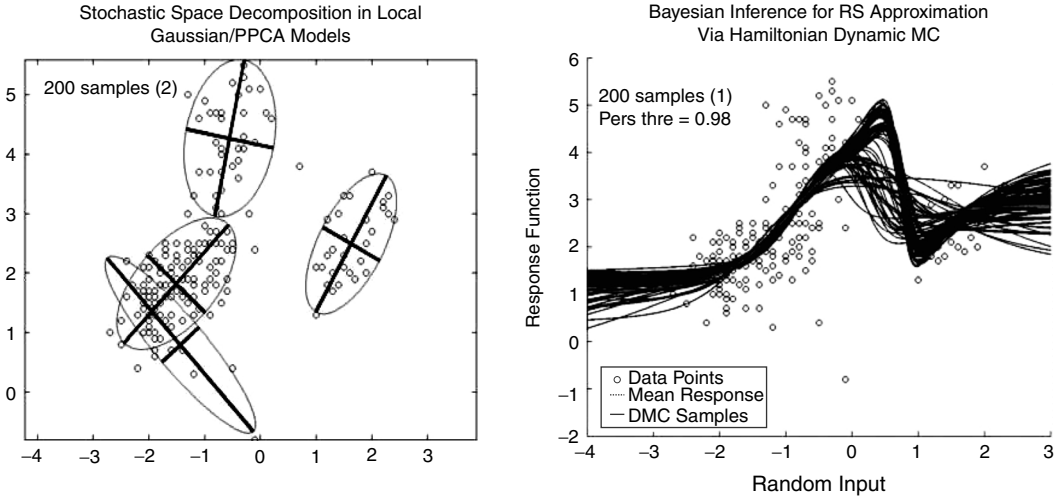

FIGURE 20.9 Stochastic Interpolation Using A Local PPCA Expansion Model via H-MCMC Simulation.

implying that the sampling error, (i.e., standard deviation) of $\hat{f}$ is O($N^{-1/2}$) independent of stochastic space dimensionality. Thus, the MC simulation estimator always converges with a constant rate that is proportional to the square-root of the number of samples. This is a key property of the standard Monte Carlo estimator that makes the SMC simulation method so popular in all engineering fields.

Many textbooks provide suggestions on how to select the required number of samples to estimate the probability of failure for a given percentage error independent of an assumed probability estimate. If the mean probability estimate $\hat{\bar{P}}_f$ has a normal distribution, the number of required samples for the confidence $1 - \alpha/2$ is given by

$$n = t^2_{1-\alpha/2,n-1}\hat{\bar{P}}_f(1-\hat{\bar{P}}_f) \tag{20.51}$$

where t is the T or Student distribution with $n - 1$ degrees of freedom.

Thus, it appears that if we respect the above equation and select a small error, then we have nothing to worry about. This is not necessarily true, especially for high-dimensional problems that involve non-Gaussian stochastic fields (or responses). The delicate problem is that for standard stochastic simulation based on SMC, although the convergence rate is independent of space dimensionality, the probability estimate variance is not independent of space dimensionality. In fact, it can increase dramatically with dimensionality increase. The probability estimate variance increase affects most severely the tail probability estimates (low and high fractiles) that typically are dependent on localized probability mass distributions in some remote areas of the high-dimensional space.

Figure 20.11 shows a simple example of the increase of probability estimate variance with space dimensionality. Consider the computation of failure probability integral, I, defined by a volume that is internal to a multivariate standard Gaussian function:

$$I = \int_0^{1/4} \ldots \int_0^{1/4} \exp\left(\frac{1}{2}x_1^2 + x_2^2 \ldots + x_n^2\right) dx_1 dx_2 \ldots dx_n \tag{20.52}$$

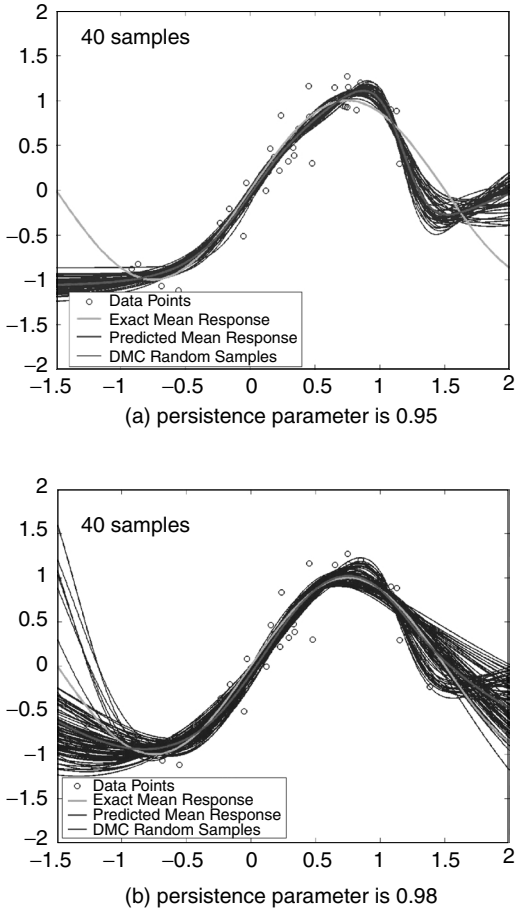

FIGURE 20.10 Effects of persistance (fictitious momenta) on the H-MCMC stochastic simulations; a) persistence parameter is 0.95, b) persistence parameter is 0.98.

Figure 20.11 shows the plot of the coefficient of variation (c.o.v.) and the 90% (normalized) confidence interval (measure of the sampling error) as a function of input space dimension N that varies up to 10. For a number of 10,000 random samples, the c.o.v. of the failure probability estimate increases from about 3% for one dimension to about 87% for ten dimensions.

Thus, for the (random) point estimates of failure probabilities that we are typically getting from our reliability analyses, the use of Equation 20.51 for determining the number of required random samples can be erroneous, and possibly unconservative, as sketched in Figure 20.12. The required number of samples increases with the complexity of the stochastic simulation model, namely with its dimensionality and its nonGaussianity (or nonlinearity of stochastic dependencies).

20.5.4 Employing Stochastic Linear PDE Solutions

A special category of stochastic fields comprises the *physics-based* stochastic fields. These physics-based fields are defined by the output of computational stochastic mechanics analyses. The physics-based stochastic fields, or stochastic functionals, can be efficiently decomposed using *stochastic reduced-order models* (ROM) based on the underlying physics of the problem. This physics is typically described by stochastic partial-differential equations. Stochastic ROMs are computed by projecting

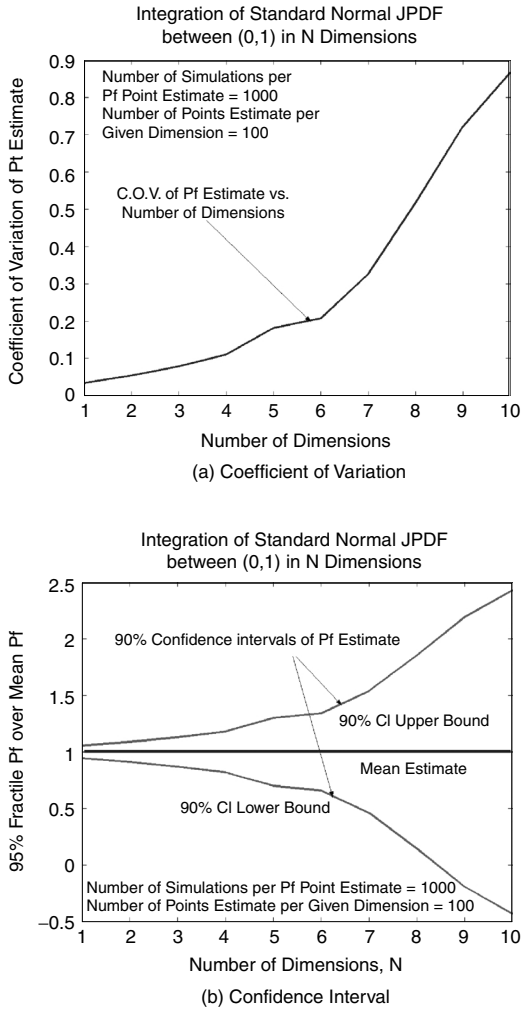

FIGURE 20.11 Effects of stochastic space dimensionality on the computed probability estimates; a) Coefficient of variation, b) Confidence interval.

the original stochastic physical field on reduced-size stochastic subspaces. Iterative algorithms using stochastic preconditioning can be employed in conjunction with physics-based stochastic ROM models.

The most used physics-based stochastic ROM models are those based on stochastic projections in the preconditioned eigen subspace, Taylor subspace, and Krylov subspace. These stochastic subspace projections are accurate for linear stochastic systems (functionals, fields) [42]. Nonlinear variations can be included by stochastically preconditioning Newton-type iterative algorithms. Both approximate and exact solvers can be used as preconditioners in conjunction with either stochastic ROM or full models. The key ingredients of the stochastic ROM implementations are (1) fast-convergent stochastic subspace expansions, (2) stochastic domain decomposition, (3) fast iterative solvers using stochastic preconditioners, and (4) automatic differentiation to compute function derivatives.

A useful industry application example of the stochastic ROM based on a eigen subspace projection is the Subset of Nominal Method (SNM) developed by Griffin and co-workers [50] for computing the mistuned (mode-localization) responses of jet engine bladed-disks due to random manufacturing deviations. The SNM approach is *exact* for proportional small variations of the linear dynamic system mass

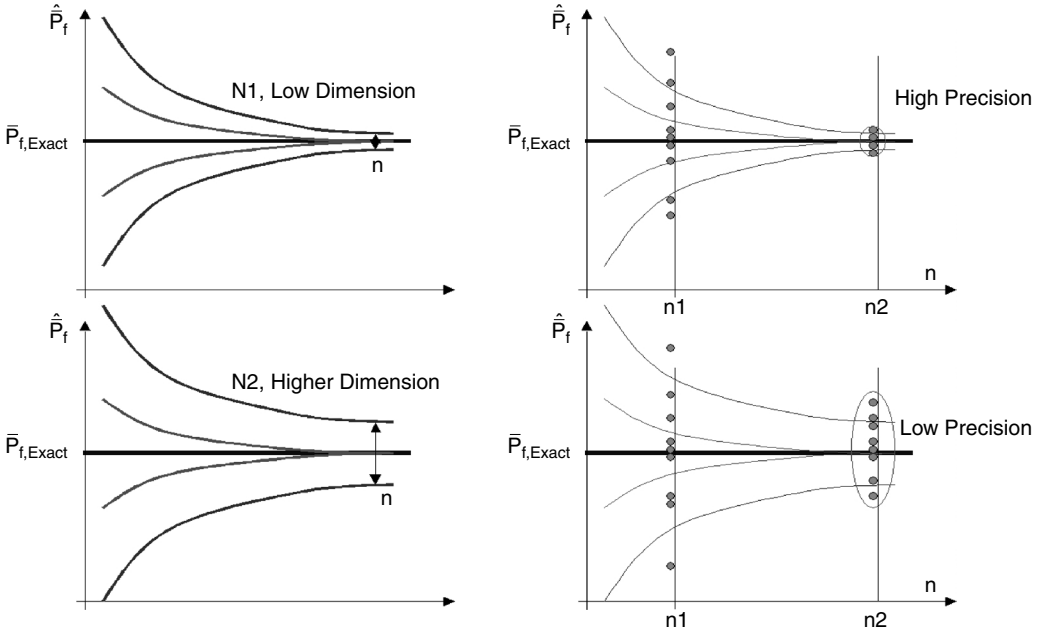

FIGURE 20.12 Space dimensionality effects on the accuracy of the computed probability estimates.

and stiffness matrices. Ghiocel [36] suggested combining the SNM approach with the KL expansion for simulating mistuned responses more realistically by including the effects of nonproportional blade variations due to manufacturing. The main idea is to use the KL expansion to reduce the stochastic input dimensionality and then use SNM iteratively to reduce the stochastic dynamic system dimensionality.

The stochastic field projections in the Taylor and Krylov subspaces can be very efficient for both static and dynamic stochastic applications. Figure 20.13 illustrates the application of the Krylov subspace expansion for computing the vibratory mistuned response of a 72-blade compressor engine blisk system. The system stochasticity is produced by the random variations in airfoil geometry due to manufacturing. Figure 20.13 shows the blade tip stochastic responses in frequency domain (transfer functions) for a given engine order excitation [34, 51]. The stochastic Krylov expansion requires only 14 basis vectors to converge. This means that only a reduced-size system of 14 differential equations must be solved at each stochastic simulation step.

The application of physics-based stochastic ROMs to stochastic simulation is limited to stochastic fields that can be described by *known* linear partial differential equations. It is obvious that the application of these stochastic ROM to computational stochastic mechanics problems is straightforward, as presented by Nair [42]. However, for stochastic fields that have an *unknown* functional structure, the physics-based stochastic ROMs are not directly applicable.

20.5.5 Incorporating Modeling Uncertainties

Engineering stochastic simulation procedures should be capable of computing confidence or variation bounds of probability estimates. The variation of probability estimates is generated by the presence of epistemic or modeling uncertainties due to (1) a lack of sufficient collection of data (small sample size issue); (2) nonrepresentative collection of statistical data with respect to the entire statistical population characteristics or stochastic system physical behavior (nonrepresentative data issue); (3) a lack of fitting the stochastic model with respect to a given statistical dataset, i.e., a bias is typically introduced due to smoothing (model statistical-fitting issue); and (4) a lack of accuracy of the prediction model with respect to real system physical behavior for given input data points, i.e., a bias is introduced at each predicted data point due to prediction inaccuracy (model lack-of-accuracy issue).

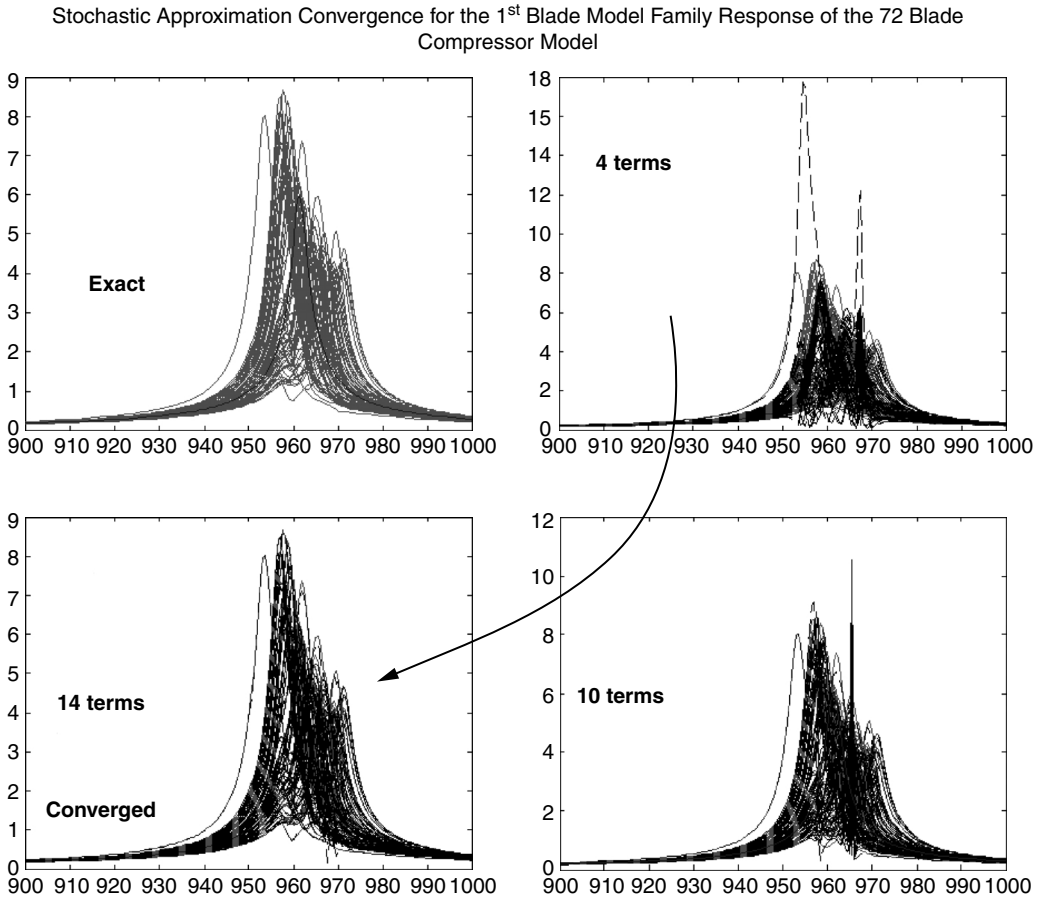

FIGURE 20.13 Convergence of stochastic Krylov subspace expansion for the random mistuned response of the blades compressor engine blisk; solid line is the full-model solution and dashed line is the reduced-model solution.

The first three modeling uncertainty categories are all associated with statistical modeling aspects. The fourth modeling uncertainty category is different. It addresses the prediction uncertainty due to the accuracy limitation of our computational models. In addition to these four uncertainty categories containing information that are objective in nature, there is also an uncertainty category related to the subjective information coming from engineering experience and judgment. Engineering judgment is an extremely useful source of information for situations where only scarce data or no data is available.

The above categories of uncertainties are described in detail in other chapters of the book. The purpose of mentioning these categories is to highlight the necessity of including them in the stochastic simulation algorithms. Unfortunately, this also increases the overall dimensionality of the stochastic simulation problem. Strategies to reduce the overall stochastic space dimensionality are needed [52].

20.6 Summary

Stochastic simulation is a powerful engineering computational tool that allows accurate predictions for complex nondeterministic problems. The most important aspect when employing simulation is to ensure that the underlying stochastic model is compatible with the physics of the problem. By oversimplifying stochastic modeling, one can significantly violate the physics behind stochastic variabilities.

Because the stochastic simulation is an almost endless subject, the focus of this chapter is on the conceptual understanding of these techniques, and not on the implementation details. This chapter reviews only the stochastic simulation techniques that in the author's opinion are the most significant for engineering design applications.

The chapter describes various stochastic simulation models — from simple, one-dimensional random variable models to complex stochastic network-based models. Significant attention is given to the decomposition of stochastic fields (or functionals) in constituent basis functions. Advances of stochastic functional analysis for developing efficient simulation techniques based on stochastic reduced-order models are included.

Another topic of a special attention is the stochastic simulation in high-dimensional spaces. The chapter reviews recent developments in stochastic simulation in high-dimensional spaces based on sequential importance sampling and dynamic simulation via a Hamiltonian formulation. Other key topics that are only briefly discussed include the computation of tail probabilities and the incorporation of epistemic or modeling uncertainties in simulation.

References

1. Hammersley, J.M. and Handscomb, D.C., *Monte Carlo Methods,* Methuen and Co. Ltd., London, 1964.
2. Knuth, D.C., *The Art of Computer Programming*, Addison-Wesley, Reading, MA, 1969.
3. Vaduva, I., *Simulation Models for Computer Applications*, The Series Mathematical Basis of Operational Research, Technical Editure, Bucharest, 1977.
4. Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., *Numerical Recipes in FORTRAN: The Art of Scientific Computing,* Cambridge University Press, New York, 1992.
5. Law, A.M. and Kelton, W.D., *Simulation Modeling and Analysis,* McGraw-Hill, New York, 1991.
6. Gamerman, D., "Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference," *Text in Statistical Science,* Chapman & Hall, New York, 1997.
7. Rubin, D.B., Using SIR Algorithm to Simulate Posterior Distributions (with discussion), in *Bayesian Statistics, Vol. 3,* Oxford University Press, Oxford, England, 395–402, 1987.
8. Gilks, W.R. and Wild, P., Adaptive Rejection Sampling for Gibbs Sampling, *Applied Statistics,* 41, 337–348, 1992.
9. Box, G.E. and Muller M.E., A Note of the Generation of Random Normal Deviates, *Annals of Mathematical Statistics,* 29, 610–611, 1958.
10. Butcher, J.C., Random Sampling from the Normal Distributions, *Computing Journal*, 3, 1961.
11. Grigoriu M., *Applied Non-Gaussian Processes,* Prentice Hall, Englewood Cliffs, NJ, 1995.
12. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., Equations of State Calculations by Fast Computing Machines, *Journal of Chemical Physics,* 21(6), 1087–1091.
13. Geman, S. and Geman, D., Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6, 721–741, 1984.
14. Ghiocel, D.M., Stochastic Field Models for Advanced Engineering Applications, *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA/ASCE/ASME Non-Deterministic Approaches Forum,* Seattle, WA, April 16–19, 2001.
15. Ghanem, R.G. and Spanos, P.D., *Stochastic Finite Elements: A Spectral Approach,* Springer-Verlag, Heidelberg, 1991.
16. Wiener, N., The Homogenous Chaos, *American Journal of Mathematics,* 60, 897–936, 1938.
17. Pearson, K., On Lines and Planes of Closest Fit to Systems of Points in Space, *Philosophical Magazine,* 2, 559–572, 1958.
18. Everitt, B.S. and Dunn, G., *Applied Multivariate Data Analysis,* Arnold, Oxford University Press, London, 2001.
19. Grigoriu, M., *Stochastic Calculus,* Birkhauser, Boston, MA, 2002.

20. Yamazaki, K. and Shinozuka, M., Safety Evaluation of Stochastic Systems by Monte Carlo Simulation *9th SMiRT Conference,* Volume M, Lausanne, August 17–21, 1987.
21. Deodatis, G., Non-Stationary Stochastic Vector Processes: Seismic Ground Motion Applications, *Probabilistic Engineering Mechanics,* 11, 149–168, 1996.
22. Shinozuka, M. and Deodatis, G., Stochastic Process Models for Earthquake Ground Motion, *Journal of Probabilistic Engineering Mechanics,* 3, 191–204, 1988.
23. Ghiocel, D.M., Uncertainties of Seismic Soil-Structure Interaction Analysis: Significance, Modeling and Examples, Invited Paper at the *US–Japan Workshop on Seismic SSI, Organized by the NSF/USGS,* Menlo Park, CA, September 22–24, 1998.
24. Ghiocel, D.M. and Ghanem, R.G., Stochastic Finite Element Analysis for Soil-Structure Interaction, *Journal of Engineering Mechanics,* ASCE, 128(1), January 2002.
25. Shinozuka, M., Stochastic Fields and Their Digital Simulation, in *Stochastic Methods of Structural Dynamics,* Schueller, G.I. and Shinozuka, M., Eds., Martinus Nijhoff, Dordrecht, 1987, 93–133.
26. Ghiocel, D.M. and Trandafir, R., Application of Space-Time Stochastic Processes in Load Modeling for Structural Design of Special Facilities, *6th National Symposium on Informatics in Civil Engineering,* Timisoara, June 1988 (in Romanian).
27. Ghiocel, D.M. and Ghiocel, D., Structural Behavior of Large Cooling Towers on Wind Load, *Scientific Bulletin of Civil Engineering,* Vol. 1, ICB, Bucharest, 1988 (in Romanian).
28. Ghiocel, D.M. and Ghiocel, D., Stochastic Modeling and Simulation of Wind Fluctuating Pressure Field on Large Diameter Cooling Tower Structures, *ASCE Probabilistic Structural Mechanics Conference,* Sandia National Laboratories, Alburquerqe, NM, July 24–26, 2004 (submitted).
29. Romanowski, M., Reduced Order Unsteady Aerodynamic and Aeroelastic Models Using Karhunen-Loeve Eigenmodes, *AIAA Journal,* Paper AIAA 96–3981, 1996.
30. Thomas, J.P., Dowell, E.H., and Hall, K.C., Three Dimensional Transonic Aeroelasticity Using Proper Orthogonal Decomposition-Based Reduced Order Models, *Journal of Aircraft,* 40(3), 544–551, 2003.
31. Ghiocel, D.M., Refined Stochastic Field Models for Jet Engine Vibration and Fault Diagnostics, *Proceedings of International Gas Turbine & Aeroengine Congress,* 2000 TURBO EXPO, Paper 2000-GT-654, Munich, Germany, May 8–11, 2000.
32. Ghiocel, D.M., A New Perspective on Health Management Using Stochastic Fault Diagnostic and Prognostic Models, *International Journal of Advanced Manufacturing Systems, Special Issue on "Non-Deterministic Methods for Design and Manufacturing Under Uncertainties,"* Vol. 4, Issue 1, October 2001.
33. Hilbert, G. and Blair, A., Accuracy of Forced Response Prediction of Gas Turbine Rotor Blades, *4th National Turbine Engine HCF Conference,* Monterey, CA, March 6–9, 1999.
34. Ghiocel, D.M., Mistuning Analysis of Bladed-Disks, in *STI BLADE-GT Engineering Research Report for AFRL/PRT at Wright-Patterson Air Force Base,* Proprietary Information, USAF Contract F33615-96-C-2678, Dayton, OH, June 2002.
35. Brown J. and Grandhi, R.V., Probabilistic Analysis of Geometric Uncertainty Effects on Blade-Alone Forced Response, *Proceedings of the 2004 ASME Turbo Expo,* IGTI 2004, Vienna, June 14–17, 2004.
36. Ghiocel, D.M., Critical Probabilistic Modeling and Implementation Issues for Geometry Mistuning in Engine Blisks, presentation at the *6th Annual FAA/AF/NASA/Navy Workshop on the Application of Probabilistic Methods to Gas Turbine Engines,* Solomon Island, MD, March 18–20, 2003.
37. Blair A. and Annis, C., Development of Probabilistic Methods for Campbell Diagram Frequency Placement, presentation at the *7th National Turbine Engine High Cycle Fatigue Conference,* Monterey, CA, April 16–19, 2003.
38. Ghiocel, D.M., Stochastic Modeling of Airfoil Thickness Spatial Variability due to Manufacturing Process, presented at the *6th National Turbine Engine High Cycle Fatigue Conference,* Jacksonville, FL, March, 5–8, 2001.
39. Ghiocel, D.M., Stochastic Field Models for Aircraft Jet Engines, *Journal of Aerospace Engineering,* ASCE, Special Issue on Reliability of Aerospace Structures, Vol. 14, No. 4, October 2001.

40. Griffiths, J.A. and Tschopp, J.A., Selected Topics in Probabilistic Gas Turbine Engine Turbomachinery Design, in *CRC Press Engineering Design Reliability Handbook,* Nikolaidis, E., Ghiocel, D.M., and Singhal, S., Eds., CRC Press, Boca Raton, FL, 2004.
41. Casenti, B., Probabilistic Blade Design, United Technologies — Pratt and Whitney Final Report, USAF Contract F33615-98-C-2928, June 2003.
42. Nair, P.B., Projection Schemes in Stochastic Finite Element Analysis, in *CRC Press Engineering Design Reliability Handbook*, Nikolaidis, E., Ghiocel, D.M., and Singhal, S., Eds., CRC Press, Boca Raton, FL, 2004.
43. Grigoriu, M., Formulation and Solution of Engineering Problems Involving Uncertainty, Invited presentation at the *Biannual SAE Probabilistic Method Committee Meeting*, Detroit, October 7–9, 2003.
44. McKay, D.J.C., *Bayesian Interpolation*, Neural Computation, 1992.
45. Robert, C.P. and Casella, G., *Monte Carlo Statistical Methods*, Springer-Verlag, Heidelberg, 1999.
46. Liu, J., Monte Carlo Strategies in Scientific Computing, *Spring Series in Statistics,* Springer-Verlag, Heidelberg, 2001.
47. Hanson, K.M., Markov Chain Monte Carlo Posterior Sampling with the Hamilton Method, *Proceedings SPIE 4322, Medical Image: Image Processing,* Sonka, M. and Hanson, M.K., Eds., San Diego, CA, 2001, 456–467.
48. Green, P.J., Monte Carlo Methods: An Overview, Invited Paper, *Proceedings of the IMA Conference on Complex Stochastic Systems and Engineering,* Leeds, U.K., September 1993.
49. Fismen, M., Exact Simulation Using Markov Chains, M.Sc. thesis, Norvegian Institute of Technology and Science, Trondheim, December 1997.
50. Yang, M.-T. and Griffin, J.H., A Reduced Order Model of Mistuning Using a Subset of Nominal Modes, *Journal of Engineering for Gas Turbines and Power,* 123(4), 893–900, 2001.
51. Keerti, A., Nikolaidis, E., Ghiocel, D.M., and Kirsch, U., Combined Approximations for Efficient Probabilistic Analysis of Structures, *AIAA Journal,* 2004.
52. Ghiocel, D.M., Combining Stochastic Networks, Bayesian Inference and Dynamic Monte Carlo for Estimating Variation Bounds of HCF Risk Prediction, presentation at the *7th National Turbine Engine High Cycle Fatigue Conference,* Monterey, CA, April 16–19, 2003.