Introduction to Polynomial Chaos

Daniel A. Cline
dcline1@binghamton.edu

Department of Systems Science and Industrial Engineering
Binghamton University

May 16, 2022

The polynomial chaos expansion method[1] is an efficient and powerful tool used for quantifying the uncertainty in models of complex systems, in what is broadly referred to as uncertainty quantification (UQ). Given a mathematical model with a set of parameters as inputs, one often wishes to understand how uncertainty in those parameters affects the outputs of the model. For models with modest computational demands, this uncertainty propagation (UP) can be analyzed using monte carlo simulation, where a distribution is assumed for each parameter, a large number of random samples of the parameter set are drawn, and the model is independently simulated using each random parameter set as input. Given the distribution of inputs, a distribution of model outputs can then be used to quantify model uncertainty. However, for computationally demanding models, it might not be feasible to run a large number of simulations, so alternative approaches with lower computational demands are needed. The polynomial chaos expansion method is one such approach.

This paper provides a brief introduction to polynomial chaos (PC) with a focus on detailed derivations and numerical calculations, with the goal of giving readers some intuition about the mechanics of the method and how to apply it. More detailed treatment of the material can be found in Xiu [4] (or Xiu [3]), which is more technical in nature but still quite accessible. Before covering polynomial chaos, we start with an example of the solution to a simple ODE, where we review concepts such as orthogonality and series representations of functions. This gives us the foundation to cover polynomial chaos. The latest version of this paper along with an accompanying python notebook with solutions to all exercises can be found here:

https://github.com/dcline1/PolynomialChaos

1 Review of Fourier Series Methods for Solving ODEs

It's helpful to provide an example of function expansions in a familiar deterministic context before presenting the main material of these notes. This section is based on (Haberman [1], sec. 2.3), but the material is standard and can be found in any introductory text on PDEs or ODEs containing Fourier series methods. We consider the following second order ordinary differential equation (ODE)

$$\frac{d^2u(x)}{dx^2} = -\lambda u(x), \tag{1.1}$$

[1] Also known as the spectral method or stochastic spectral method.

subject to boundary conditions $u(0) = 0$ and $u(L) = 0$ and some initial condition $f(x)$ for $0 \leq x \leq L$. If we guess $u(x)$ has the form $u(x) = e^{rx}$, we get

$$\frac{du(x)}{dx} = re^{rx} = ru(x), \qquad \frac{d^2u(x)}{dx^2} = r^2e^{rx} = r^2u(x),$$

Setting $r^2 = -\lambda$ gives us (1.1), and we have $r = \pm\sqrt{-\lambda} = \pm i\sqrt{\lambda}$, where $i = \sqrt{-1}$. Therefore, solutions of (1.1) have the form $u(x) = ce^{\pm i\sqrt{\lambda}x}$ for some constant c. For simplicity, we only consider the case where $\lambda > 0$. Using the relationships $e^{i\alpha x} = \cos\alpha x + i\sin\alpha x$ and $e^{-i\alpha x} = \cos\alpha x - i\sin\alpha x$, we may write the solutions in the following form

$$\begin{aligned} u(x) &= c_1e^{i\sqrt{\lambda}x} = c_1(\cos\sqrt{\lambda}x + i\sin\sqrt{\lambda}x), \\ u(x) &= c_2e^{-i\sqrt{\lambda}x} = c_2(\cos\sqrt{\lambda}x - i\sin\sqrt{\lambda}x), \end{aligned}$$

for some arbitrary (possibly complex) constants c_1 and c_2. Since linear combinations of solutions of ODEs are also solutions, we may alternatively write

$$\begin{aligned} u(x) &= c_1(\cos\sqrt{\lambda}x + i\sin\sqrt{\lambda}x) + c_2(\cos\sqrt{\lambda}x - i\sin\sqrt{\lambda}x) \\ &= a\cos\sqrt{\lambda}x + b\sin\sqrt{\lambda}x, \end{aligned} \tag{1.2}$$

where $a = (c_1 + c_2)$ and $b = (c_1 - c_2)i$. Note that since c_1 and c_2 can be complex, we can take b to be real.[2] We say that (1.2) is the real solution of (1.1). Applying the first boundary condition, $u(0) = 0$, we immediately get $a = 0$, since $\cos 0 = 1$. Applying the second boundary condition, $u(L) = 0$, gives

$$u(L) = b\sin\sqrt{\lambda}L = 0,$$

which is only true if $b = 0$ (the trivial solution) or $\sin\sqrt{\lambda}L = 0$, which can only occur when $\sqrt{\lambda}L = n\pi$, where n is an integer (the graph of $\sin$ only crosses zero at $n\pi$). We therefore have

$$\lambda = \left(\frac{n\pi}{L}\right)^2.$$

Plugging this back into (1.2), we see the real solution to (1.1) has the form

$$u(x) = b\sin\frac{n\pi x}{L}.$$

Again, since linear combinations of solutions of ODEs are themselves solutions, we have the general solution

$$u(x) = \sum_{n=1}^{p} b_n\Phi_n(x), \tag{1.3}$$

for some $p > 0$ where we've introduced

$$\Phi_n(x) = \sin\frac{n\pi x}{L}$$

[2]For example, take $c_1 = (a - bi)/2$ and $c_2 = (a + bi)/2$, where $a, b \in \mathbb{R}$.

for notational convenience. It just so happens that, with moderate restrictions, any piecewise-smooth function $f(x)$ on the interval $x \in [0, L]$ can be represented as

$$f(x) = \sum_{n=1}^{\infty} b_n \Phi_n(x), \tag{1.4}$$

which is known as the Fourier series of $f(x)$. In order to satisfy the initial condition, we therefore just need to determine the coefficients b_n. This is done using the orthogonality of $\sin$ functions[3]

$$\langle \Phi_m(x)\Phi_n(x) \rangle = \int_0^L \Phi_m(x)\Phi_n(x)dx = \begin{cases} L/2, & m = n \\ 0, & m \neq n \end{cases}, \tag{1.5}$$

where $\langle \cdot \cdot \rangle$ denotes our chosen inner product (in this case just the integral over our domain with no weighting function). It's common to write (1.5) more concisely as

$$\langle \Phi_m(x)\Phi_n(x) \rangle = \frac{L}{2}\delta_{mn}, \tag{1.6}$$

where

$$\delta_{mn} = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases}$$

is the Kronecker delta function. Now to determine the m^{th} coefficient, b_m, we simply multiply both sides of (1.4) by $\Phi_m(x)$ and integrate over our domain to get[4]

$$\begin{aligned} \int_0^L f(x)\Phi_m(x)dx &= \int_0^L \left(\sum_{n=1}^{\infty} b_n \Phi_n(x) \right) \Phi_m(x)dx \\ &= \sum_{n=1}^{\infty} b_n \left(\int_0^L \Phi_n(x)\Phi_m(x)dx \right) \\ &= \sum_{n=1}^{\infty} b_n \langle \Phi_n(x)\Phi_m(x) \rangle \\ &= b_m \langle \Phi_m(x)\Phi_m(x) \rangle \\ &= b_m \frac{L}{2} \end{aligned}$$

since $\langle \Phi_m(x)\Phi_m(x) \rangle = \langle \Phi_m^2(x) \rangle = L/2$ by (1.6) and we've used the orthogonality of sines to drop all other terms. Solving for b_m, we get

$$b_m = \frac{2}{L} \int_0^L f(x)\Phi_m(x)dx, \tag{1.7}$$

or more compactly,

$$b_m = \frac{\langle f(x)\Phi_m(x) \rangle}{\langle \Phi_m^2(x) \rangle}. \tag{1.8}$$

In this section, we introduced the Fourier series representation of a function and used the orthogonality of functions to obtain the coefficients. These concepts are fundamental to polynomial chaos.

[3] See, e.g., this video for a proof.

[4] We assume we can interchange the integral and summation here.

2 Polynomial Chaos Expansions

Similar to Fourier series expansions of functions as linear combinations of orthogonal trigonometric functions, $\Phi_j(x)$, we can also represent functions as linear combinations of orthogonal polynomials

$$f(x) = \sum_{j=0}^{\infty} a_j \Psi_j(x), \tag{2.1}$$

where $\{\Psi_j(x) \mid j = 0, 1, ...\}$ is a family of orthogonal polynomials, often referred to as the orthogonal polynomial basis, satisfying

$$\langle \Psi_m(x)\Psi_n(x) \rangle = \int \Psi_m(x)\Psi_n(x)w(x)dx = \gamma_m \delta_{mn}, \tag{2.2}$$

where the inner product is defined with respect to some weighting function $w(x)$ and γ_m is a normalization constant. Note the similarity to (1.5), but whereas Fourier series use orthogonal periodic functions for the basis, polynomial chaos expansions use orthogonal polynomials, which are not periodic.

Following the same procedure we used for the Fourier series, we can solve for the i^{th} coefficient, a_i, by multiplying both sides of (2.1) by $\Psi_i(x)$ and taking the inner product to get[5]

$$\begin{aligned}
\int f(x)\Psi_i(x)w(x)dx &= \int \left(\sum_{j=0}^{\infty} a_j \Psi_j(x) \right) \Psi_i(x)w(x)dx \\
&= \sum_{j=0}^{\infty} a_j \left(\int \Psi_j(x)\Psi_i(x)w(x)dx \right) \\
&= \sum_{j=0}^{\infty} a_j \langle \Psi_j(x)\Psi_i(x) \rangle \\
&= a_i \langle \Psi_i^2(x) \rangle
\end{aligned}$$

where we used polynomial orthogonality to drop all terms other than i. Solving for a_i, we get

$$a_i = \frac{\langle f(x)\Psi_i(x) \rangle}{\langle \Psi_i^2(x) \rangle}, \tag{2.3}$$

which is analogous to (1.8). In practice, we truncate (2.1) at some positive integer p, which gives us the following approximation

$$f(x) \approx f_p(x) = \sum_{j=0}^{p} a_j \Psi_j(x) \tag{2.4}$$

The weighting function, $w(x)$, is often chosen to be a probability density function, and certain classes of polynomials are orthogonal with respect to well-known probability density functions. Examples of orthogonal polynomials (and the distribution, or *germ*, corresponding to their weighting

[5] Again, we assume we can interchange the integral and summation here.

functions) include Legendre (uniform), Hermite (Gaussian), and Leguerre (exponential or Gamma) polynomials. We will focus on Hermite polynomials in this paper, but the results are easily extended to the others.

Linear algebra provides some intuition for (2.1). In linear algebra, an n-dimensional vector can be represented as a linear combination of n basis vectors. If we have a function instead of a vector, we can construct a vector as a discretized version of a function by taking function values at a discrete set of points. As we increase the resolution of the sampling points, we get an infinite dimensional vector in the limit. Therefore, functions can be thought of as infinite dimensional vectors, and we see that the linear combination of an infinite number of orthogonal polynomials for functions is analogous to a linear combination of n orthogonal basis vectors for n-dimensional vectors.

3 Hermite Polynomials

As mentioned above, orthogonal Hermite polynomials use a Gaussian weighting function. We will use z to denote the independent variable when discussing Hermite polynomials.

Polynomial chaos expansions using Hermite polynomials have the following form

$$\begin{aligned} f(z) &= a_0 H_0 \\ &+ \sum_{i_1=1}^{d} a_{i_1} H_1(z_{i_1}) \\ &+ \sum_{i_1=1}^{d} \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(z_{i_1}, z_{i_2}) \\ &+ \sum_{i_1=1}^{d} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(z_{i_1}, z_{i_2}, z_{i_3}) \\ &+ \ ... \end{aligned} \tag{3.1}$$

where

$$H_n(z_{i_1}, ..., z_{i_n}) = e^{\frac{1}{2} z^T z} (-1)^n \frac{\partial^n}{\partial z_{i_1} ... \partial z_{i_n}} e^{-\frac{1}{2} z^T z} \tag{3.2}$$

and $z = (z_1, ...z_d) \in \mathbb{R}^d$ is a d-dimensional vector. We can write (3.1) in the form of (2.1) by defining a one-to-one mapping between the $a_{i_1 ... i_n}$ and a_j coefficients and a one-to-one mapping between the $H_n(z_{i_1}, ..., z_{i_n})$ basis functions and the $\Psi_n(z)$ polynomials.

The notation in (3.1) and (3.2) can be confusing, so we'll work through the details for 1-dimensional Hermite polynomials.[6] We start with a scalar $z \in \mathbb{R}$ and consider the following Gaussian-type equation

$$e^{-\frac{1}{2} z^2}. \tag{3.3}$$

Taking the derivative, we get

$$\frac{d}{dz} e^{-\frac{1}{2} z^2} = -z e^{-\frac{1}{2} z^2}.$$

[6] See Appendix A for 2-dimensional Hermite polynomials.

Taking the derivative again by applying the product rule, we get

$$\frac{d^2}{dz^2}e^{-\frac{1}{2}z^2} = \frac{d}{dz}\left(-ze^{-\frac{1}{2}z^2}\right) = \left(z^2-1\right)e^{-\frac{1}{2}z^2}.$$

And taking the derivative again, we get

$$\begin{aligned}\frac{d^3}{dz^3}e^{-\frac{1}{2}z^2} &= \frac{d}{dz}\left(z^2-1\right)e^{-\frac{1}{2}z^2} \\ &= 2ze^{-\frac{1}{2}z^2} + (z^2-1)(-z)e^{-\frac{1}{2}z^2} \\ &= -(z^3-3z)e^{-\frac{1}{2}z^2}.\end{aligned}$$

Using the following assignments for the polynomials on the right-hand sides

$$\begin{aligned}H_1(z) &= z \\ H_2(z) &= z^2-1 \\ H_3(z) &= z^3-3z\end{aligned}$$

leads to the following general definition

$$H_n(z) = e^{\frac{1}{2}z^2}(-1)^n\frac{d^n}{dz^n}e^{-\frac{1}{2}z^2}, \tag{3.4}$$

which also satisfies $H_0(z) = 1$. By assigning $H_1(z) = H_1(z)$, $H_2(z) = H_2(z,z)$, $H_3(z) = H_3(z,z,z)$, and so on, we see that (3.4) fits the form given in (3.2), since $H_n(z)$ is just $H_n(z_{i_1},...,z_{i_n})$ for $d=1$. These are the Hermite polynomials for scalar z.[7]

It turns out that Hermite polynomials are orthogonal.[8] More concretely, it can be shown that the inner product has the following closed-form solution

$$\langle H_m(z)H_n(z)\rangle = \int_{-\infty}^{\infty} H_m(z)H_n(z)\phi(z)dx = n!\,\delta_{mn} \tag{3.5}$$

where

$$\phi(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}$$

is the standard normal probability density function. Note that (3.5) is zero whenever $m \neq n$, as required.

Starting with (3.1), we may plug in (3.2) for a 1-dimensional z, which is just (3.4), to get

$$\begin{aligned}f(z) &= a_0H_0 + a_1H_1(z) + a_{11}H_2(z,z) + a_{111}H_3(z,z,z) + ... \\ &= a_0H_0 + a_1H_1(z) + a_{11}H_2(z) + a_{111}H_3(z) + ...\end{aligned} \tag{3.6}$$

or, equivalently,

$$f(z) = a_0 + a_1z + a_{11}(z^2-1) + a_{111}(z^3+3z) + ... \tag{3.7}$$

[7] Specifically, the *probabilist's* Hermite polynomials. If we instead used e^{-z^2} for the derivation, we would get the *physicist's* Hermite polynomials.

[8] See, e.g., these videos for a proof.

Assigning $\{a_1 = a_1, a_2 = a_{11}, a_3 = a_{111}, ...\}$ and $\Psi_n(z) = H_n(z)$, we can recast this in the form of (2.1) as

$$f(z) = \sum_{j=0}^{\infty} a_j H_j(z). \tag{3.8}$$

Exercise 3.a Plot the first 3, 5, and 8 Hermite polynomials. What can be said about them?

Exercise 3.b Construct the Hermite polynomial expansion of $f(z) = 10 + 2z$.

Exercise 3.c Construct the Hermite polynomial expansion of $f(z) = \sin(4z)/z$.

Exercise 3.d Construct the Hermite polynomial expansion of $f(z) = |z|$.

4 Polynomial Chaos Expansions of Random Variables

We've seen how to represent an arbitrary deterministic function as a linear combination of orthogonal basis functions, for both the Fourier basis and for orthogonal polynomial bases. It turns out, we can also represent functions of random variables using orthogonal polynomial bases in (2.1). As might be expected, constructing orthogonal polynomials with respect to probability densities allows for some analytical results.

For a random variable, X, with known probability distribution, the natural choice of orthogonal polynomial basis is the class for which $w(x)$ in (2.2) is related to the probability density function of the random variable X. For instance, the natural choice of orthogonal polynomial basis for a Gaussian random variable, Z, is the Hermite polynomial basis $\Psi_j = H_j$, since the associated weighting function is $\phi(z)$.

Exercise 4.a Construct the Hermite polynomial expansion of $f(Z) = Z$ where $Z \sim N(0, 1)$. What are the values of the coefficients?

Exercise 4.b Construct the Hermite polynomial expansion of $f(Z) = 10+2Z$ where $Z \sim N(0, 1)$. What are the values of the coefficients?

Exercise 4.c Construct the Hermite polynomial expansion of $f(Z) = e^{\mu+\sigma Z}$, where $\mu = 1$, $\sigma = 0.1$, and $Z \sim N(0, 1)$. What are the values of the coefficients?

This works well for functions of standard classes of known probability distributions. But what if we have an arbitrary random variable Y with cumulative distribution function $F_Y(y) = P(Y \leq y)$? In this case, not only might Y not be a function of a known random variable X, but the CDF might not even have an analytical form (e.g. it could be an empirical distribution). We show below that by transforming the germ X into a uniform random variable, we can generalize the inner product $\langle Y\ \Psi_i(x)\rangle$ to handle any arbitrary Y.

We start by transforming X into a $U \sim U(0,1)$ uniform random variable using its CDF $F_X(x) = P(X \leq x)$, since $F_X(X) \sim U$.[9] Then given the CDF of Y, we can transform U into a Y random variable using the inverse transform $Y \sim F_Y^{-1}(U)$. We therefore have the mapping $Y(X) = F_Y^{-1}(F_X(X))$. This gives us the following inner product

$$\int_{-\infty}^{\infty} Y(x)\Psi_k(x)w(x)dx = \int_{-\infty}^{\infty} F_Y^{-1}(F_X(x))\Psi_k(x)w(x)dx \tag{4.1}$$

and we can rewrite (2.3) as

$$a_i = \frac{\langle F_Y^{-1}(F_X(x))\ \Psi_i(x)\rangle}{\langle \Psi_i^2(x)\rangle}. \tag{4.2}$$

While the above transformations are theoretically sound, they have a tendency to be numerically unstable due to numerical errors in the tails of (4.1). However, since $F_X(X) \sim U$ and $X \sim F_X^{-1}(U)$, we can transform the inner product in (4.1) with respect to X into the following inner product with respect to a $U(0,1)$ random variable

$$\int_{-\infty}^{\infty} F_Y^{-1}(F_X(x))\Psi_k(x)w(x)dx = \int_0^1 F_Y^{-1}(u)\Psi_k(F_X^{-1}(u))du, \tag{4.3}$$

since the weighting function (i.e. probability density function) in the inner product for the $U(0,1)$ uniform distribution is $w(u) = 1$. Note the change in the integration limits. The coefficients are therefore

$$a_i = \frac{\langle F_Y^{-1}(u)\ \Psi_k(F_X^{-1}(u))\rangle_u}{\langle \Psi_k^2(x)\rangle} \tag{4.4}$$

where $\langle \cdot\ \cdot \rangle_u$ is the inner product for the uniform distribution. Note that no adjustments are needed for the denominator, since we may continue to use the orthogonality of $\Psi_i(x)$ just as we did before. While the right-hand sides of (4.1) and (4.3) are mathematically equivalent, (4.3) avoids numerical errors in the tails, since the integral is taken from $[0,1]$ instead of $(-\infty, \infty)$.

Finally, since $F_X^{-1}(U) \sim X$, we can rewrite the polynomial chaos expansion in (2.1) as

$$f(u) = \sum_{j=0}^{\infty} a_j \Psi_j(F_X^{-1}(u)) \tag{4.5}$$

Exercise 4.d Using (4.4)–(4.5), construct the Hermite polynomial expansion of $f(Z) = e^{\mu+\sigma Z}$, where $\mu = 1$, $\sigma = 0.1$, and $Z \sim N(0,1)$. What are the values of the coefficients?

[9] The cumulative distribution function $P(X \leq x) = F_X(x) = u$ maps any value of x into $u \in [0,1]$, and the inverse transform $F_X^{-1}(u) = x$ maps $u \in [0,1]$ into x. Furthermore, $F_X^{-1}(F_X(x)) = x$ and $F_X(F_X^{-1}(u)) = u$. Therefore,

$$P(F_X(X) \leq c) = P(X \leq F_X^{-1}(c)) = F_X(F_X^{-1}(c)) = c, \qquad c \in [0,1]$$

so $F_X(X)$ is uniformly distributed, since only the uniform distribution satisfies $P(U \leq c) = c$. Similarly,

$$P(F_X^{-1}(U) \leq x) = P(U \leq F_X(x)) = F_X(x),$$

again since $P(U \leq c) = c$ where we've taken $c = F_X(x)$ in this case, so $F_X^{-1}(U) \sim X$. This latter relationship is known as the inverse transform method, since it allows us to transform a uniform random variable U into X using the inverse transform of X.

Exercise 4.e Using (4.4)–(4.5), construct the Hermite polynomial expansion of $Y \sim Beta(\alpha, \beta)$, where $\alpha = \beta = 0.5$. What are the values of the coefficients?

Exercise 4.f Using (4.4)–(4.5), construct the Hermite polynomial expansion of $Y \sim Exp(\lambda)$, where $\lambda = 0.5$. What are the values of the coefficients?

Note that instead of using (4.4)–(4.5) for the Hermite polynomial chaos expansion of the exponential random variable in Exercise 4.f, we could have directly used (2.2)–(2.4) with Leguerre polynomials, since their associated germ is the exponential (Gamma) pdf. So which approach is better? While both will work, the polynomial chaos expansion using the orthogonal polynomial basis associated with the given random variable (Leguerre in this case) will usually converge to the random variable much quicker (i.e. the expansion will require a smaller p, sometimes substantially smaller).

4.1 Statistical Moments

Note that the right side of (2.4) is an *analytical* surrogate to function f on the left. This opens up the possibility of analytically calculating the approximate moments of $f(X)$. When the weighting function in the inner product is a probability density function and the orthogonal polynomials are normalized such that $\Psi_0(X) = 1$, we can directly use the inner products to calculate expected values, since $\langle \cdot \rangle = E[\cdot]$. This gives us

$$\begin{aligned} E[f(X)] &\approx E[f_p(X)] \\ &= E[f_p(X)\Psi_0(X)] \\ &= E\left[\left(\sum_{j=0}^{p} a_j \Psi_j(X)\right)\Psi_0(X)\right] \\ &= \sum_{j=0}^{p} E\left[a_j \Psi_j(X)\Psi_0(X)\right] \\ &= \sum_{j=0}^{p} a_j \langle \Psi_j(X)\Psi_0(X)\rangle \\ &= a_0, \end{aligned} \tag{4.6}$$

since $\langle \Psi_0^2(X)\rangle = 1$, and we see that $E[f(X)] \approx a_0$. Note that this is a general result and holds

regardless of the class of orthogonal polynomials used in (2.4). Similarly,

$$\begin{aligned} E[f^2(X)] &\approx E[f_p^2(X)] \\ &= E\left[\left(\sum_{j=0}^{p} a_j\Psi_j(X)\right)\left(\sum_{j=0}^{p} a_j\Psi_j(X)\right)\right] \\ &= \left\langle\left(\sum_{j=0}^{p} a_j\Psi_j(X)\right)\left(\sum_{j=0}^{p} a_j\Psi_j(X)\right)\right\rangle \\ &= \left\langle\sum_{j=0}^{p} a_j^2\Psi_j^2(X)\right\rangle \\ &= \sum_{j=0}^{p} a_j^2\langle\Psi_j^2(X)\rangle. \end{aligned}$$

The variance is therefore

$$\begin{aligned} Var(f(X)) &= E\left[f^2(X)\right] - E^2\left[f(X)\right] \\ &\approx E\left[f_p^2(X)\right] - E^2\left[f_p(X)\right] \\ &= \sum_{j=0}^{p} a_j^2\langle\Psi_j^2(X)\rangle - a_0^2 \\ &= \sum_{j=1}^{p} a_j^2\langle\Psi_j^2(X)\rangle, \end{aligned} \tag{4.7}$$

again since $\langle\Psi_0^2(X)\rangle = 1$. Therefore, once we solve for the coefficients of our polynomial chaos representation of f, the mean and variance are essentially automatic using (4.6) and (4.7).

Exercises 4.1.a-f Calculate the mean and variance of all random variables in exercises 4.a-f using (4.6) and (4.7). Do they match the closed-form statistics for these distributions?

5 Polynomial Chaos Expansions of Functions with Random Parameters

We will use the following simple equation as our example model

$$\frac{dv(t)}{dt} = -r\,v(t), \qquad v(0) = v_0 = 3 \tag{5.1}$$

where the model parameter r is random. We wish to calculate the average model response, as well as the 95% confidence intervals. This problem can be solved using monte carlo simulation, which requires many simulations of the model. In practice, if the model is computationally expensive to evaluate, running a monte carlo simulation might be prohibitively expensive.

Alternatively, we can use polynomial chaos, which can be applied in two main ways: intrusive and non-intrusive. We will cover both approaches below.

Exercise 5.a Assuming $r \sim N(\mu, \sigma^2)$, where $\mu = 0.2$ and $\sigma = 0.05$, and $v_0 = 3$, draw $N = 10,000$ random realizations of r, and for each one, evaluate (5.1) numerically over $t = [0, 10]$ using $10,000$ timesteps. Calculate the sample mean and standard deviation at each timestep. Plot all model runs, along with the mean and 95% confidence intervals through time.

5.1 Intrusive Approach

The intrusive approach applies polynomial chaos expansions directly to both the model parameters *and* model equations. Intrusive methods therefore require full knowledge of the model equations. The most common intrusive approach is the stochastic Galerkin projection, which we cover below.

Given our example model, we start by applying the polynomial chaos expansion method to both r and $v(t)$ using the same orthogonal polynomial basis (with germ x) for both as follows

$$r(x) \approx \sum_{i=0}^{p} \hat{r}_i \Psi_i(x), \tag{5.2}$$

$$v(x,t) \approx \sum_{j=0}^{p} \hat{v}_j(t) \Psi_j(x). \tag{5.3}$$

Taking the derivative of (5.3) with respect to time gives

$$\frac{dv(x,t)}{dt} \approx \frac{d}{dt}\left(\sum_{j=0}^{p} \hat{v}_j(t)\Psi_j(x)\right) = \sum_{j=0}^{p} \frac{d\hat{v}_j(t)}{dt}\Psi_j(x). \tag{5.4}$$

Note that the polynomials, $\Psi_j(x)$, are not functions of time. Similar to the derivation in Section 2, we multiply (5.4) by $\Psi_k(x)$ for some $0 \leq k \leq p$ and take the inner product to get

$$\begin{aligned}
\int \frac{dv(x,t)}{dt}\Psi_k(x)w(x)dx &\approx \int \left(\sum_{j=0}^{p} \frac{d\hat{v}_j(t)}{dt}\Psi_j(x)\right)\Psi_k(x)w(x)dx \\
&= \sum_{j=0}^{p} \int \frac{d\hat{v}_j(t)}{dt}\Psi_j(x)\Psi_k(x)w(x)dx \\
&= \sum_{j=0}^{p} \frac{d\hat{v}_j(t)}{dt}\int \Psi_j(x)\Psi_k(x)w(x)dx \\
&= \sum_{j=0}^{p} \frac{d\hat{v}_j(t)}{dt}\langle \Psi_j(x)\Psi_k(x)\rangle \\
&= \frac{d\hat{v}_k(t)}{dt}\langle \Psi_k^2(x)\rangle.
\end{aligned} \tag{5.5}$$

Similarly, the inner product of the right-hand side of (5.1) and $\Psi_k(x)$ can be approximated using the polynomial chaos representations in (5.2) and (5.3) as follows

$$\begin{aligned}
-\int r(x)v(x,t)\Psi_k(x)w(x)dx &\approx -\int \left(\sum_{i=0}^{p}\hat{r}_i\Psi_i(x)\right)\left(\sum_{j=0}^{p}\hat{v}_j(t)\Psi_j(x)\right)\Psi_k(x)w(x)dx \\
&= -\int \sum_{i=0}^{p}\sum_{j=0}^{p}\hat{r}_i\hat{v}_j(t)\Psi_i(x)\Psi_j(x)\Psi_k(x)w(x)dx \\
&= -\sum_{i=0}^{p}\sum_{j=0}^{p}\int \hat{r}_i\hat{v}_j(t)\Psi_i(x)\Psi_j(x)\Psi_k(x)w(x)dx \\
&= -\sum_{i=0}^{p}\sum_{j=0}^{p}\hat{r}_i\hat{v}_j(t)\langle\Psi_i(x)\Psi_j(x)\Psi_k(x)\rangle, \qquad (5.6)
\end{aligned}$$

where the triple product is given by[10]

$$\langle\Psi_i(x)\Psi_j(x)\Psi_k(x)\rangle = \int \Psi_i(x)\Psi_j(x)\Psi_k(x)w(x)dx. \qquad (5.7)$$

Equating (5.5) and (5.6), we get

$$\frac{d\hat{v}_k(t)}{dt}\langle\Psi_k^2(x)\rangle = -\sum_{i=0}^{p}\sum_{j=0}^{p}\hat{r}_i\hat{v}_j(t)\langle\Psi_i(x)\Psi_j(x)\Psi_k(x)\rangle,$$

or in terms of the derivative,

$$\frac{d\hat{v}_k(t)}{dt} = -\sum_{j=0}^{p}\hat{v}_j(t)\left(\sum_{i=0}^{p}\hat{r}_i\frac{\langle\Psi_i(x)\Psi_j(x)\Psi_k(x)\rangle}{\langle\Psi_k^2(x)\rangle}\right). \qquad (5.8)$$

This looks quite complicated, but note that quantity in parentheses is constant, since $\hat{r}_i$, $\langle\Psi_k^2(x)\rangle$, and $\langle\Psi_i(x)\Psi_j(x)\Psi_k(x)\rangle$ are all constants. Therefore, for the vector $\hat{v}(t) \in \mathbb{R}^p$, we have the following p-dimensional linear system of equations

$$\frac{d\hat{v}}{dt} = A\hat{v}, \qquad (5.9)$$

where entries in the $p \times p$ matrix A are given by (5.8). In other words, the stochastic Galerkin projection gives us a single p^{th} order ODE to solve instead of the N ODE's we'd need to run for monte carlo analysis. Since p is typically small (e.g. $p < 50$) and N is typically large (e.g. $N = 10,000$), the savings in computational time can be substantial.

[10]The triple product for Hermite polynomials has a closed-form solution, given by

$$\langle H_i(z)H_j(z)H_k(z)\rangle = \int H_i(z)H_j(z)H_k(z)\phi(z)dz = \frac{i!j!k!}{(s-i)!(s-j)!(s-k)!},$$

where $s \geq \min\{i,j,k\}$ and $s = (i+j+k)/2$ is even.

Exercise 5.b Construct the Hermite polynomial expansion of $r \sim N(\mu, \sigma^2)$, where $\mu = 0.2$ and $\sigma = 0.05$ for $p = 10$. What are the values of the coefficients?

Exercise 5.c Using the coefficients from 5.b and (5.8), construct the A in (5.9) for the example model (5.1) and solve the system of equations. Plot the solutions (coefficients) through time. Calculate the mean and standard deviation through time using the solutions of the system of ODEs. Plot the mean and 95% confidence intervals through time. Compare with the monte carlo results in 5.a.

5.2 Non-Intrusive Approach

When we do not know the details of the model or the model is not amenable to the intrusive approach (e.g. when we aren't able to reformulate and recode the model, as is necessary with the intrusive approach), we can use a non-intrusive approach, where we essentially treat the model as a black box.

Just like the intrusive approach, we start by assuming r and $v(t)$ are represented using (5.2) and (5.3). But rather than explicitly deriving the equations for $v(t)$, we instead take a set of N pairs of model parameter inputs, $r_i = r(x_i)$, and model outputs, $v_i = v(x_i, t)$, where x_i is the germ and t is the same fixed value for for all i. In other words, we generate the x_i (e.g. using some experimental design methodology), derive $r(x_i)$ using its polynomial chaos expansion (5.2), and then evaluate $v_i = v(t)$ using each value of r_i. This gives us our training set $\mathcal{D} = \{(x_i, v_i) \mid 1 \leq i \leq N\}$.

We wish to solve for the coefficients in (5.3), such that the following equations are satisfied for all samples

$$\sum_{j=0}^{p} a_j \Psi_j(x_i) \approx v_i, \qquad 1 \leq i \leq N. \tag{5.10}$$

Note that this function is linear in a_j and that the x_i are transformed using the orthogonal polynomial basis functions for germ x. In other words, this problem has the following form

$$\begin{bmatrix} \Psi_0(x_1) & \cdots & \Psi_p(x_1) \\ \vdots & \ddots & \vdots \\ \Psi_0(x_N) & \cdots & \Psi_p(x_N) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_p \end{bmatrix} \approx \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}, \tag{5.11}$$

or more succinctly

$$\mathbf{\Psi a} \approx \mathbf{v}, \tag{5.12}$$

where $\mathbf{\Psi}$ is the $N \times p$ design matrix, $\mathbf{a}$ is a $p \times 1$ column vector, and $\mathbf{v}$ is a $N \times 1$ column vector. Since our goal is to solve for the coefficient vector, $\mathbf{a}$, we can formulate the problem using least-squares and solve it with linear regression.[11]

The number of function evaluations required for fitting the coefficients is typically much smaller than for monte carlo simulation. This means that it's often much faster to fit the surrogate

[11] To avoid very large coefficients due to numerical noise, it might be desirable in practice to use some form of regularization (e.g. Ridge regression) along with cross validation.

function (i.e. coefficients) and calculate moments than it is to run a full monte carlo simulation. Therefore, just like intrusive methods, the savings in computational time can be substantial.

Another popular non-intrusive approach is the stochastic collocation method, which we do not cover here (see Xiu [4]).

Exercise 5.d Taking $N = 100$, construct the vector $z_i = F_Z^{-1}(u_i)$, where $u_i = (i + 0.5)/N$ are equally spaced points over the interval $(0, 1)$. For each z_i, calculate $r_i = 0.2 + 0.05z_i$ and evaluate (5.1) numerically over $t = [0, 10]$ using $10,000$ timesteps. Construct the design matrix $\mathbf{\Psi}$ using $\{z_1, ..., z_N\}$ and Hermite polynomials as the orthogonal polynomial basis. For each timestep, solve for the coefficients $(a_1, ..., a_p)$ using linear regression such that the coefficients are functions of time $(a_1(t), ..., a_p(t))$. Plot the solutions (coefficients) through time. Calculate the mean and standard deviation through time using the coefficients. Plot the mean and 95% confidence intervals through time. Compare with the monte carlo results in 5.a.

References

[1] Haberman, R. (2003) *Applied Partial Differential Equations, 4th Ed.* Prentice Hall.

[2] O'Hagan, A. (2013) Polynomial Chaos: A Tutorial and Critique from a Statistician's Perspective.

[3] Xiu, D. (2004) *Generalized (Wiener-Askey) Polynomial Chaos* [Doctoral dissertation, Brown University].

[4] Xiu, D. (2010) *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press.

Appendix

A Hermite Polynomials in 2-Dimensions

To solidify our understanding, we work through the Hermite polynomial expansion for two dimensions. Take $z = (z_1, z_2)$. In this case, analogous to (3.3), we have

$$e^{-\frac{1}{2}z^T z} = e^{-\frac{1}{2}(z_1^2+z_2^2)}. \tag{A.1}$$

Taking the derivative with respect to z_1, we get

$$\frac{\partial}{\partial z_1} e^{-\frac{1}{2}z^T z} = -z_1 e^{-\frac{1}{2}z^T z}.$$

Applying (3.2), we get

$$H_1(z_1) = e^{-\frac{1}{2}z^T z}(-1)\left(-z_1 e^{-\frac{1}{2}z^T z}\right) = z_1. \tag{A.2}$$

Similarly, $H_1(z_2) = z_2$. Taking the derivative again with respect to z_1 by applying the product rule, we get

$$\frac{\partial^2}{\partial z_1 \partial z_1} e^{-\frac{1}{2}z^T z} = \frac{\partial}{\partial z_1}\left(-z_1 e^{-\frac{1}{2}z^T z}\right) = \left(z_1^2 - 1\right) e^{-\frac{1}{2},z^T z},$$

which gives us

$$H_2(z_1, z_1) = e^{-\frac{1}{2}z^T z}(-1)^2(z_1^2 - 1)e^{-\frac{1}{2}z^T z} = z_1^2 - 1. \tag{A.3}$$

Similarly, $H_2(z_2, z_2) = z_2^2 - 1$. Taking the cross derivative, we get

$$\frac{\partial^2}{\partial z_2 \partial z_1} e^{-\frac{1}{2}z^T z} = \frac{\partial}{\partial z_2}\left(-z_1 e^{-\frac{1}{2}z^T z}\right) = z_1 z_2 e^{-\frac{1}{2}z^T z},$$

which leads to

$$H_2(z_2, z_1) = e^{-\frac{1}{2}z^T z}(-1)^2(z_2 z_1)e^{-\frac{1}{2}z^T z} = z_2 z_1. \tag{A.4}$$

The third-order derivative with respect to z_1 is given by

$$\begin{aligned}
\frac{\partial^3}{\partial z_1 \partial z_1 \partial z_1} e^{-\frac{1}{2}z^T z} &= \frac{\partial^2}{\partial z_1 \partial z_1}\left(-z_1 e^{-\frac{1}{2}z^T z}\right) \\
&= \frac{\partial}{\partial z_1}\left(z_1^2 - 1\right) e^{-\frac{1}{2}z^T z} \\
&= 2z_1 e^{-\frac{1}{2}z^T z} + (z_1^2 - 1)(-z_1)e^{-\frac{1}{2}z^T z} \\
&= -(z_1^3 - 3z_1)e^{-\frac{1}{2}z^T z},
\end{aligned}$$

which gives us

$$H_3(z_1, z_1, z_1) = e^{-\frac{1}{2}z^T z}(-1)^3\left(-(z_1^3 - 3z_1)e^{-\frac{1}{2}z^T z}\right) = z_1^3 - 3z_1. \tag{A.5}$$

Similarly, $H_3(z_2, z_2, z_2) = z_2^3 - 3z_2$. Finally, we have

$$\begin{aligned}
\frac{\partial^3}{\partial z_2 \partial z_1 \partial z_1} e^{-\frac{1}{2}z^T z} &= \frac{\partial^2}{\partial z_2 \partial z_1} \left(-z_1 e^{-\frac{1}{2}z^T z} \right) \\
&= \frac{\partial}{\partial z_2} \left(z_1^2 - 1 \right) e^{-\frac{1}{2}z^T z} \\
&= (z_1^2 - 1)(-z_2) e^{-\frac{1}{2}z^T z} \\
&= -(z_1^2 z_2 - z_2) e^{-\frac{1}{2}z^T z},
\end{aligned}$$

which leads to

$$H_3(z_2, z_1, z_1) = e^{-\frac{1}{2}z^T z}(-1)^3 \left(-(z_1^2 z_2 - z_2) e^{-\frac{1}{2}z^T z} \right) = z_1^2 z_2 - z_2. \tag{A.6}$$

By symmetry, we also have $H_3(z_2, z_2, z_1) = z_1 z_2^2 - z_1$.

Applying (3.1) with $z = (z_1, z_2)$, we get

$$\begin{aligned}
f(z) &= a_0 H_0 \\
&+ a_1 H_1(z_1) + a_2 H_1(z_2) \\
&+ a_{11} H_2(z_1, z_1) + a_{21} H_2(z_2, z_1) + a_{22} H_2(z_2, z_2) \\
&+ a_{111} H_3(z_1, z_1, z_1) + a_{211} H_3(z_2, z_1, z_1) + a_{221} H_3(z_2, z_2, z_1) + a_{222} H_3(z_2, z_2, z_2) \\
&+ \ldots
\end{aligned} \tag{A.7}$$

or, equivalently,

$$\begin{aligned}
f(z) &= a_0 \\
&+ a_1 z_1 + a_2 z_2 \\
&+ a_{11}(z_1^2 - 1) + a_{21} z_1 z_2 + a_{22}(z_2^2 - 1) \\
&+ a_{111}(z_1^3 - 3z_1) + a_{211}(z_1^2 z_2 - z_2) + a_{221}(z_1 z_2^2 - z_1) + a_{222}(z_2^3 - 3z_2) \\
&+ \ldots
\end{aligned} \tag{A.8}$$

Enumerating the summands from left to right, we can represent (A.7) and (A.8) using the simplified notation in (2.1).