Introduction to Uncertainty Quantification
Module 3.4: Surrogate Models for Uncertainty Propagation

1 Introduction

Let us consider again the function $Y = h(\boldsymbol{X})$ where $\boldsymbol{X}$ is a random vector. In many applications, the function $h(\boldsymbol{X})$ represents a complex numerical model of a physical system that comes at high computational cost (e.g. a finite element model). The Monte Carlo methods we discussed in the previous lesson are robust, but even with variance reduction they can be prohibitively expensive. We therefore seek alternatives that allow us to propagate uncertainty.

The most common approach in recent years has been to develop a so-called *surrogate model* (also sometimes referred to as a *metamodel* or *emulator*) that approximates the system $h(\boldsymbol{X})$ with a simpler mathematical function, $\hat{h}(\boldsymbol{X})$, that can be learned from data generated by running a small number of full model evaluations and is computationally cheap to evaluate. The fact that the surrogate model is computationally inexpensive allows Monte Carlo simulation and related methods to be performed on the approximate function, and therefore facilitates uncertainty propagation.

Many different types of surrogate models are used for uncertainty propagation. The two most widely used are Gaussian process regression (GPR) with the popular Kriging method being a special case and polynomial chaos expansions (PCE). We will cover these two methods in some detail. There are several other methods that are also used including radial basis functions (RBFs), deep neural networks (DNNs), support vector machines, and other forms of regression such as polynomial response surfaces. We will also briefly touch on some of these methods, but will not cover them in detail here.

2 Gaussian Process Regression

Gaussian Process (GP) regression is a machine learning (ML) method that seeks to identify the best fit Gaussian stochastic process to a set of data points. GP regression is a widely-used ML method to construct surrogate models for complex computational models. To introduce GP regression, consider a computational model $y(\boldsymbol{x})$ having input vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$. The model $y(\boldsymbol{x})$ is then approximated by

$$\mathcal{Y}(\boldsymbol{x}, \omega) = \mathcal{F}(\boldsymbol{x}) + Z(\boldsymbol{x}, \omega) \tag{1}$$

where $\mathcal{F}(\cdot)$ is a regression model and $Z(\cdot)$ is a zero-mean Gaussian random process having sample space indexed by $\omega \in \Omega$. We consider that the regression model $\mathcal{F}(\cdot)$ is defined through a linear combination of basis functions $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_p(\boldsymbol{x})]^T$ having coefficients $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_p]^T$ as

$$\mathcal{F}(x) = \boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) \tag{2}$$

This regression model represents the mean value or the *trend* in the model and is often modeled with polynomial functions – see Section 2.3. The Gaussian random process $Z(\cdot)$ is considered to have zero mean and covariance:

$$\mathbb{E}[Z(\boldsymbol{x}_1)Z(\boldsymbol{x}_2)] = \sigma_z^2 \mathcal{R}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta}) \tag{3}$$

where $\mathcal{R}(\cdot)$ is the autocorrelation function having hyperparameters $\boldsymbol{\theta}$ and σ_z is the variance of the process. The regression function is modeled using a positive definite kernel function $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$, several examples of which are provided in Section 2.3.

The goal of GP regression is then to learn the hyperparameters $\boldsymbol{\theta}$ and regression coefficients $\boldsymbol{\beta}$ such that the model provides the best fit to a set of training data corresponding to input realizations $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)}\}$ and corresponding model outputs $\boldsymbol{Y} = \{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}$ evaluated by $y^{(i)} = y(\boldsymbol{x}^{(i)})$. We will consider two distinct cases. In the first case, the data points are assumed to be exact and the GP model is designed to interpolate between the training points. This is referred to as Kriging [cite]. In the second case, the data points are assumed to contain some noise and the model is fit as a regressor. Additional details on both cases and more information on the method can be found in the textbook by Rasmussen and Williams [1].

2.1 Kriging for Interpolation

The first case we'll consider is the case where the GP model serves as an interpolant between the existing data points, which we refer to as Kriging. Kriging was originally developed to interpolate geospatial data by Krige [2] and the methods was later formalized in this setting by Matheron [3]. Kriging was first used for surrogate modeling in the context of design of computer experiments by Sacks et al. [4] in 1989.

The Kriging method takes the input-output data, $\boldsymbol{X}, \boldsymbol{Y}$, and builds a probability model to interpolate the data by assuming that the interpolant is a Gaussian process. That is, the Gaussian assumption states that the joint probability distribution between the interpolant (predictor) $\mathcal{Y}(\boldsymbol{x})$ and the data $\boldsymbol{Y}$ follows the joint normal distribution with

$$\left\{ \begin{array}{c} \mathcal{Y}(\boldsymbol{x}) \\ \boldsymbol{Y} \end{array} \right\} \sim N\left(\left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{\beta} \\ \boldsymbol{F}\boldsymbol{\beta} \end{array} \right\}, \sigma_z^2 \left\{ \begin{array}{cc} 1 & \boldsymbol{r}(\boldsymbol{x})^T \\ \boldsymbol{r}(\boldsymbol{x}) & \boldsymbol{R} \end{array} \right\} \right) \tag{4}$$

where

- $\boldsymbol{F}$ is the matrix of basis function evaluations at the training points given by $F_{ij} = f_j(\boldsymbol{x}^{(i)}), i = 1, \ldots, n, j = 1, \ldots, p$
- $\boldsymbol{r}(\boldsymbol{x})$ is the vector of correlations between the prediction point $\boldsymbol{x}$ and the training points $\boldsymbol{x}^{(i)}$ given by $r_i = \mathcal{R}(\boldsymbol{x}, \boldsymbol{x}^{(i)}; \boldsymbol{\theta}), i = 1, \ldots, n$
- $\boldsymbol{R}$ is the correlation matrix of points in the training set given by $R_{ij} = \mathcal{R}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}; \boldsymbol{\theta}), i, j = 1, \ldots, n$

Given the joint normal probability density in Eq. (4), the conditional distribution of the prediction at point $\boldsymbol{x}$ is also normal

$$\mathcal{Y}(\boldsymbol{x}) \mid \boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}\left(\hat{y}(\boldsymbol{x}), \sigma_{\hat{y}}^2(\boldsymbol{x})\right), \tag{5}$$

having mean

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{\beta} + \boldsymbol{r}(\boldsymbol{x})^T\boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{6}$$

and variance

$$\sigma_{\hat{y}}^2(\boldsymbol{x}) = \sigma_z^2\left(1 - \boldsymbol{r}(\boldsymbol{x})^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \boldsymbol{t}(\boldsymbol{x})^T(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{t}(\boldsymbol{x})\right) \tag{7}$$

where

$$\boldsymbol{t}(\boldsymbol{x}) = \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}) \tag{8}$$

Eq. (5) is referred to as the *Kriging predictor*. The prediction takes the form of a normal random variable having mean in Eq. (6) and variance given in Eq. (7). We specifically note that the variance of the predictor in Eq. (7) is zero at the training points and therefore the Kriging predictor serves to *interpolate* between the training data.

To fit the GP model and obtain the Kriging predictor, it is necessary to determine the best fit regression parameters $\boldsymbol{\beta}$, variance σ_z, and GP hyperparameters $\boldsymbol{\theta}$. This can be done in several ways. Here, we will describe the two most commonly applied methods – maximum likelihood estimation (MLE) and cross-validation (CV).

Maximum Likelihood Estimation

Using MLE, the predictor can be determined by identifying the hyperparameters $\boldsymbol{\theta}$, variance σ_z, and the regression coefficients $\boldsymbol{\beta}$ such that the likelihood of the observations $\boldsymbol{Y}$ is maximized. Since $\boldsymbol{Y}$ is assumed to be drawn from a multivariate normal distribution, the likelihood function can be given by

$$\mathcal{L}(\boldsymbol{\theta}, \sigma_z, \boldsymbol{\beta}|\boldsymbol{Y}) = \frac{(\det \boldsymbol{R})^{-1/2}}{(2\pi\sigma_z^2)^{n/2}} \exp\left[-\frac{1}{2\sigma_z^2}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta})^T \boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta})\right] \tag{9}$$

Maximizing this likelihood function, we can obtain the regression coefficients $\boldsymbol{\beta}$ and variance σ_z^2 analytically as

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}(\boldsymbol{\theta}) = \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{Y} \tag{10}$$

and

$$\hat{\sigma_z}^2 = \sigma_z^2(\boldsymbol{\theta}) = \frac{1}{n} (\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta})^T \boldsymbol{R}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{11}$$

The proof can be found in [5].

We can then obtain the hyperparameters $\boldsymbol{\theta}$ by solving the following optimization problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left(-\log(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{Y}))\right) \tag{12}$$

Applying the relations in Eqs. (9), (10), and (11) this optimization problem can be expressed

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \left(\log(\det(\boldsymbol{R}) + n \log(2\pi\sigma_z^2) + n\right) \tag{13}$$

Solution of this optimization problem can be performed with your choice of optimizer. Performing this optimization is outside the scope of this document.

Cross-Validation

An alternate approach to obtain the GP parameters is to perform cross-validation (CV). In the process of *K-fold cross-validation*, the training data $\mathcal{D} = [\boldsymbol{X}, \boldsymbol{Y}]$ is split into K mutually exclusive and collectively complete subsets $\mathcal{D}_k, k = 1, \ldots, K$ such that

$$\begin{aligned} &\mathcal{D}_i \cap \mathcal{D}_j = \emptyset, \quad \forall i \neq j \quad \text{and} \\ &\cup_{k=1}^{K} \mathcal{D}_k = \mathcal{D} \end{aligned} \tag{14}$$

Then, K predictions are obtained by estimating the model parameters such that, in the k^{th} prediction all of the data are used for training except the k^{th} subset. The model is then used to predict the values from the k^{th} subset. The special case of *leave-one-out cross-validation* (LOO-CV) sets $K = n$ so that each subset contains a single training point.

The CV error for the k^{th} subset is given by

$$\epsilon_{CV}^k = \sum_{(x^{(i)}, y^{(i)}) \in \mathcal{D}_k} \left(y^{(i)} - \hat{y}_{\mathcal{D}\setminus\mathcal{D}_k}\left(\boldsymbol{x}^{(i)} \mid \boldsymbol{\beta}, \sigma_z^2, \boldsymbol{\theta}\right)\right)^2 \tag{15}$$

where $\hat{y}_{\mathcal{D}\setminus\mathcal{D}_k}(\cdot)$ denotes the mean Kriging predictor using all training data except those in $\mathcal{D}_k$. The total CV error is then given by averaging the CV errors across all K subsets

$$\epsilon_{CV} = \frac{1}{K}\sum_{k=1}^{K}\epsilon_{CV}^{k} \tag{16}$$

To train the model using CV, we then aim to identify the parameters $\boldsymbol{\beta}$, σ_z^2, and $\boldsymbol{\theta}$ that minimize the total CV error where we can again take advantage of the analytical expression $\boldsymbol{\beta}$ in Eq. (10). We further note that $\hat{y}(\boldsymbol{x})$ does not depend on σ_z^2. Therefore, the minimization reduces to

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \epsilon_{CV}(\boldsymbol{\theta} \mid \boldsymbol{Y}) \tag{17}$$

The variance is then computed as

$$\hat{\sigma}_z^2 = \hat{\sigma}_z^2(\hat{\boldsymbol{\theta}}) = \frac{1}{K}\sum_{k=1}^{K}\sum_{(x^{(i)},y^{(i)})\in\mathcal{D}_k}\frac{\left(y^{(i)} - \hat{y}_{\mathcal{D}\setminus\mathcal{D}_k}\left(\boldsymbol{x}^{(i)} \mid \hat{\boldsymbol{\theta}}\right)\right)^2}{\sigma^2_{\hat{y},\mathcal{D}\setminus\mathcal{D}_k}(\boldsymbol{x}^{(i)} \mid \hat{\boldsymbol{\theta}})/\sigma_z^2} \tag{18}$$

where $\sigma^2_{\hat{y},\mathcal{D}\setminus\mathcal{D}_k}(\cdot)$ is the variance of the GP predictor given from Eq. (7) estimated from all data except those in $\mathcal{D}_k$. A more detailed discussion of these estimates can be found in [6].

2.2 Regression of Noisy Functions

When using a GP for prediction of noisy function, we assume that the noisy function can be expressed as

$$y = h(\boldsymbol{X}) + \varepsilon \tag{19}$$

where ε is an additive noise this is assumed to follow a zero-mean Gaussian distribution $\varepsilon \sim N(0, \boldsymbol{\Sigma}_{\mathrm{n}})$ with covariance matrix $\boldsymbol{\Sigma}_{\mathrm{n}}$. Depending on the nature of the noise, three cases arise:

- *Homoscedastic noise*: $\boldsymbol{\Sigma}_{\mathrm{n}} = \sigma_{\mathrm{n}}^2\boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix. Here, the variance is independent and constant across all observations.
- *Independent Heteroscedastic noise (Heterogeneous)*: $\boldsymbol{\Sigma}_{\mathrm{n}} = \mathrm{diag}(\sigma_{\mathrm{n}}^2)$. Here the noise variances differ at each point, but remain uncorrelated.
- *General Heteroscedastic noise*: Here $\boldsymbol{\Sigma}_{\mathrm{n}}$ can be any valid covariance matrix allowing for differing noise at each point and correlations between the components at each point.

With the introduction of noise, the joint Gaussian distribution from Eq. (4) becomes

$$\begin{Bmatrix} \mathcal{Y}(\boldsymbol{x}) \\ \boldsymbol{Y} \end{Bmatrix} \sim N\left(\begin{Bmatrix} \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{\beta} \\ \boldsymbol{F}\boldsymbol{\beta} \end{Bmatrix}, \begin{Bmatrix} \sigma_z^2 & \sigma_z^2\boldsymbol{r}(\boldsymbol{x})^T \\ \sigma_z^2\boldsymbol{r}(\boldsymbol{x}) & \sigma_z^2\boldsymbol{R} + \boldsymbol{\Sigma}_{\mathrm{n}} \end{Bmatrix}\right) \tag{20}$$

where the terms are defined exactly as above, except we now introduce the noise covariance $\boldsymbol{\Sigma}_{\mathrm{n}}$. This yields the conditional distribution of the predictor having the same Gaussian form

$$\mathcal{Y}(\boldsymbol{x}) \mid \boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}\left(\hat{y}(\boldsymbol{x}), \sigma_{\hat{y}}^2(\boldsymbol{x})\right), \tag{21}$$

but with mean and variance modified to account for the noise as follows. Introducing the covariance matrix $\boldsymbol{C} = \sigma_z^2\boldsymbol{R} + \boldsymbol{\Sigma}_{\mathrm{n}}$ and the cross-covariance $\boldsymbol{c}(\boldsymbol{x}) = \sigma_z^2 r(\boldsymbol{x})$, the mean of the predictor is given by

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^T\boldsymbol{\beta} + \boldsymbol{c}(\boldsymbol{x})^T\boldsymbol{C}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{22}$$

and variance is given by

$$\sigma_{\hat{y}}^2(\boldsymbol{x}) = \left(\sigma_z^2 - \boldsymbol{c}(\boldsymbol{x})^T \boldsymbol{C}^{-1} \boldsymbol{c}(\boldsymbol{x}) + \boldsymbol{u}_c(\boldsymbol{x})^T (\boldsymbol{F}^T \boldsymbol{C}^{-1} \boldsymbol{F})^{-1} \boldsymbol{u}_c(\boldsymbol{x})\right) \tag{23}$$

where

$$\boldsymbol{u}_c(\boldsymbol{x}) = \boldsymbol{F}^T \boldsymbol{C}^{-1} \boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}) \tag{24}$$

One can clearly see that these equations are identical to the noise-free Eqs. (6)-(8) but accounting for the full covariance with noise.

In the following, we will deal only with the *homoscedastic* case for which $\boldsymbol{C} = \sigma_z^2 \boldsymbol{R} + \sigma_{\mathrm{n}}^2 \boldsymbol{I}$. The total variance is therefore given by

$$\sigma_t^2 = \sigma_z^2 + \sigma_{\mathrm{n}}^2 \tag{25}$$

Let us define the ratio of noise to total variance by

$$\tau = \frac{\sigma_{\mathrm{n}}^2}{\sigma_t^2} \tag{26}$$

and perform the following factorizations

$$\tilde{\boldsymbol{r}}(\boldsymbol{x}) = (1 - \tau) \boldsymbol{r}(\boldsymbol{x}) \tag{27}$$

and

$$\tilde{\boldsymbol{R}} = (1 - \tau) \boldsymbol{R} + \tau \boldsymbol{I}. \tag{28}$$

We can then write the joint Gaussian distribution as

$$\left\{ \begin{array}{c} \mathcal{Y}(\boldsymbol{x}) \\ \boldsymbol{Y} \end{array} \right\} \sim N\left(\left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{\beta} \\ \boldsymbol{F}\boldsymbol{\beta} \end{array} \right\}, \sigma_t^2 \left\{ \begin{array}{cc} 1 & \tilde{\boldsymbol{r}}^T(\boldsymbol{x}) \\ \tilde{\boldsymbol{r}}(\boldsymbol{x}) & \tilde{\boldsymbol{R}} \end{array} \right\} \right) \tag{29}$$

conditional distribution of the predictor as

$$\mathcal{Y}(\boldsymbol{x}) \mid \boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}\left(\hat{y}(\boldsymbol{x}), \sigma_{\hat{y}}^2(\boldsymbol{x})\right), \tag{30}$$

having mean

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^T \boldsymbol{\beta} + \tilde{\boldsymbol{r}}(\boldsymbol{x})^T \tilde{\boldsymbol{R}}^{-1} (\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{31}$$

and variance

$$\sigma_{\hat{y}}^2(\boldsymbol{x}) = \sigma_t^2 \left(1 - \tilde{\boldsymbol{r}}(\boldsymbol{x})^T \tilde{\boldsymbol{R}}^{-1} \tilde{\boldsymbol{r}}(\boldsymbol{x}) + \boldsymbol{u}(\boldsymbol{x})^T (\boldsymbol{F}^T \tilde{\boldsymbol{R}}^{-1} \boldsymbol{F})^{-1} \boldsymbol{u}(\boldsymbol{x})\right) \tag{32}$$

where

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{F}^T \tilde{\boldsymbol{R}}^{-1} \tilde{\boldsymbol{r}}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}) \tag{33}$$

In the noisy case described here, the Gaussian predictor does not possess zero variance, and thus it does not provide an exact estimator at the existing data points. It therefore serves as a regressor, rather than an interpolant. Training the GP model follows the same process as in the noiseless case above where again we can determine the model variance σ_z^2, regression parameters $\boldsymbol{\beta}$, correlation function hyperparameters $\boldsymbol{\theta}$ using either MLE or CV, but we now must also learn the noise variance σ_{n}^2. We review these two approaches next – again for the homoscedastic case.

Maximum Likelihood Estimation

Once again, since $\boldsymbol{Y}$ is assumed to be drawn from a multivariate normal distribution, the likelihood function can be given by

$$\mathcal{L}(\boldsymbol{\theta}, \sigma_z, \boldsymbol{\beta}|\boldsymbol{Y}) = \frac{(\det \tilde{\boldsymbol{R}})^{-1/2}}{(2\pi\sigma_t^2)^{n/2}} \exp\left[-\frac{1}{2\sigma_t^2}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta})^T \tilde{\boldsymbol{R}}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta})\right] \tag{34}$$

which is identical to Eq. (9) except with σ_z^2 replaced with σ_t^2 and $\boldsymbol{R}$ replaced with $\tilde{\boldsymbol{R}}$. Maximizing this likelihood function, we can obtain the regression coefficients $\boldsymbol{\beta}$ and total variance σ_t^2 analytically as

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}(\boldsymbol{\theta}) = \left(\boldsymbol{F}^T \tilde{\boldsymbol{R}}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^T \tilde{\boldsymbol{R}}^{-1} \boldsymbol{Y} \tag{35}$$

and

$$\hat{\sigma_t}^2 = \sigma_t^2(\boldsymbol{\theta}) = \frac{1}{n} (\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta})^T \tilde{\boldsymbol{R}}^{-1}(\boldsymbol{Y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{36}$$

We can then obtain the hyperparameters $\boldsymbol{\theta}$ by solving the following optimization problem

$$\hat{\boldsymbol{\theta}}, \hat{\tau} = \arg\min_{\boldsymbol{\theta}, \tau} \frac{1}{2} \left(\log(\det(\tilde{\boldsymbol{R}}) + n \log(2\pi\sigma_t^2) + n\right) \tag{37}$$

where we now also learn the additional parameter $\tau \in (0, 1)$ that expresses the fraction of variance attributed to noise. Solution of this optimization problem can be performed with your choice of optimizer. Performing this optimization is outside the scope of this document.

Cross-Validation

Using CV, we recognize that the CV error now depends on the additional parameter τ because we use the revised estimator $\hat{y}_{\mathcal{D}\setminus\mathcal{D}_k}(\boldsymbol{x}|\boldsymbol{\theta}, \tau)$ from Eq. (31) again estimated from all data except those in $\mathcal{D}_k$. Therefore, the minimization can be expressed as

$$\hat{\boldsymbol{\theta}}, \tau = \arg\min_{\boldsymbol{\theta}, \tau} \epsilon_{CV}(\boldsymbol{\theta}, \tau \mid \boldsymbol{Y}) \tag{38}$$

The total variance is then computed as

$$\hat{\sigma}_t^2 = \hat{\sigma}_t^2(\hat{\boldsymbol{\theta}}, \hat{\tau}) = \frac{1}{K} \sum_{k=1}^{K} \sum_{(x^{(i)}, y^{(i)}) \in \mathcal{D}_k} \frac{\left(y^{(i)} - \hat{y}_{\mathcal{D}\setminus\mathcal{D}_k}\left(\boldsymbol{x}^{(i)} \mid \hat{\boldsymbol{\theta}}, \hat{\tau}\right)\right)^2}{\sigma^2_{\hat{y}, \mathcal{D}\setminus\mathcal{D}_k}(\boldsymbol{x}^{(i)} \mid \hat{\boldsymbol{\theta}}, \hat{\tau})/\sigma_t^2} \tag{39}$$

where $\sigma^2_{\hat{y}, \mathcal{D}\setminus\mathcal{D}_k}(\cdot)$ is the variance of the GP predictor given from Eq. (32) estimated from all data except those in $\mathcal{D}_k$.

2.3 Common Kernels and Regression Functions

In this section, we briefly review the commonly used correlation functions (kernels) and regression models in Gaussian process regression.

Correlation Functions (Kernels)

The correlation function $\mathcal{R}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta})$ is typically expressed through a symmetric positive semi-definite kernel used to define the functional form of the linear dependence between points in the GP. The form of this function can have profound impact on the resulting GP and its fit to the data. For example, the form of the kernel encodes assumptions such as continuity, differentiability, etc. and its hyperparameters $\boldsymbol{\theta}$ typically dictate a length-scale that dictates how strongly correlated points in the GP are as a function of

the distance between them. In multi-dimensional cases having M dimensions the hyperparameter vector is typically also M-dimensional (i.e. $\boldsymbol{\theta} \in \mathbb{R}^M$) such that there is one parameter (length-scale) per dimension.

Usually the multi-dimensional correlation function will be expressed as a function of one-dimensional kernels $K(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta})$ using either the *ellipsoidal form* where

$$\mathcal{R}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta}) = K(h) \tag{40}$$

where

$$h = \left[\sum_{i=1}^{M} \left(\frac{x_{1i} - x_{2i}}{\theta_i} \right)^2 \right]^{0.5} \tag{41}$$

or in *separable form* where

$$\mathcal{R}(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\theta}) = \prod_{i=1}^{M} K(x_{1i}, x_{2i}; \theta_i) \tag{42}$$

A special case of these correlation functions is the *isotropic* correlation function in which $\theta_i = \theta, \forall i$. That is, the correlation function has the same hyperparameter in each dimension. and the above expressions simplify as

$$\mathcal{R}(\boldsymbol{x}_1, \boldsymbol{x}_2; \theta) = K(h) \tag{43}$$

where

$$h = \frac{1}{\theta} \left[\sum_{i=1}^{M} (x_{1i} - x_{2i})^2 \right]^{0.5} \tag{44}$$

or in *separable form* where

$$\mathcal{R}(\boldsymbol{x}_1, \boldsymbol{x}_2; \theta) = \prod_{i=1}^{M} K(x_{1i}, x_{2i}; \theta) \tag{45}$$

and the above optimizations reduce to identifying a scalar hyperparameter θ rather than the full M-dimensional vector $\boldsymbol{\theta}$.

In all of the above cases (ellipsoidal, separable, isotropic), the kernel is typically selected from one of the following popular functional forms (all expressed in scalar form for integration in the ellipsoidal or separable forms):

- *Linear*: The linear kernel takes the following form:

$$K(x_1, x_2; \theta) = \max \left(0, 1 - \frac{|x_1 - x_2|}{\theta} \right) \tag{46}$$

  Here, the length-scale parameter dictates the slope of the function and the function implies a $\mathcal{C}^0$ continuous but non-differentiable Gaussian process. Examples of this kernel are shown in 1a.

- *Exponential*: The exponential kernel takes the following form:

$$K(x_1, x_2; \theta) = \exp \left[-\frac{|x_1 - x_2|}{\theta} \right] \tag{47}$$

  Here, the length-scale parameter dictates the rate of exponential decay of the function and the function again implies a $\mathcal{C}^0$ continuous but non-differentiable Gaussian process. Examples of this kernel are shown in 1b.

- *Gaussian*: (aka squared exponential or radial basis function kernel) The Gaussian kernel takes the following form:

$$K(x_1, x_2; \theta) = \exp\left[-\frac{1}{2}\left(\frac{|x_1 - x_2|}{\theta}\right)^2\right] \tag{48}$$

Here, the length-scale parameter correspond to the standard deviation of the Gaussian, which also dictate the rate of decay of the squared exponential. Unlike the exponential and linear, this kernel implies a continuous and infinitely differentiable Gaussian process – which is often desirable for practical applications and for encouraging smoothness of the resulting surrogate. Examples of this kernel are shown in 1c.

- *Matérn*: The Matérn kernel takes the following form:

$$K(x_1, x_2; \theta) = \frac{1}{2^{\nu-1}\Gamma(\nu)}\left(2\sqrt{\nu}\frac{|x_1 - x_2|}{\theta}\right)^{\nu} \mathcal{K}_\nu\left(2\sqrt{\nu}\frac{|x_1 - x_2|}{\theta}\right) \tag{49}$$

where $\nu \geq 1/2$ is a shape parameter, $\Gamma(\cdot)$ is the Gamma function, and $\mathcal{K}_\nu$ is the modified Bessel function of the second kind. The resulting GP is $\lceil \nu - 1 \rceil$ times differentiable where $\lceil \cdot \rceil$ denotes the ceiling function. For $\nu = 1/2$, the Matérn corresponds to the exponential kernal and as $\nu \to \infty$ it approaches the Gaussian kernel. The most commonly applied cases are the $\nu = 3/2$ and $\nu = 5/2$ cases which can be written as

$$K(x_1, x_2; \theta, \nu = 3/2) = \left(1 + \sqrt{3}\frac{|x_1 - x_2|}{\theta}\right)\exp\left[-\sqrt{3}\frac{|x_1 - x_2|}{\theta}\right] \tag{50}$$

and

$$K(x_1, x_2; \theta, \nu = 5/2) = \left(1 + \sqrt{5}\frac{|x_1 - x_2|}{\theta} + \frac{5}{3}\frac{|x_1 - x_2|}{\theta}\right)\exp\left[-\sqrt{5}\frac{|x_1 - x_2|}{\theta}\right] \tag{51}$$

Examples of these kernels are shown in Fig 1d and 1e.

Regression Functions

The regression function defines the underlying trend of the model. It is typically defined, as shown in Eq. (2) by a set of basis functions $\boldsymbol{f}(\boldsymbol{x})$ and corresponding coefficients $\boldsymbol{\beta}$. These are typically defined through simple polynomials and the following summarizes the most commonly used regression model types.

- *Simple Kriging*: In simple Kriging, the basis functions are defined arbitrarily but the coefficients are pre-determined and are all set equal to 1, i.e. $\beta_j = 1, \forall j$. That is

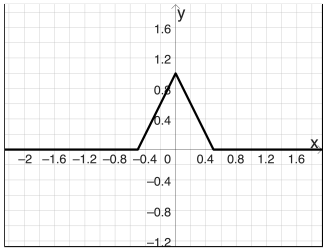
$$\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{P} f_j(\boldsymbol{x}) \tag{52}$$

- *Ordinary Kriging*: In ordinary Kriging, the regression is simply set equal to a constant. That is $f_0(\boldsymbol{x}) = 1$ and $f_j(\boldsymbol{x}) = 0, \forall j > 0$ thus

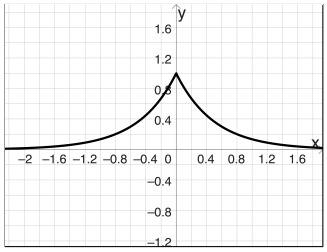
$$\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) = \beta_0 f_0(\boldsymbol{x}) = \beta_0 \tag{53}$$

- *Universal Kriging*: Universal Kriging is the most general form of the Kriging regression model and generally prescribes the trend in terms of an abritrary set of functions, usually specified as polynomials in the form

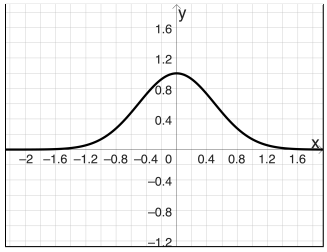
$$\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) = \sum_{j=0}^{P} \beta_j f_j(\boldsymbol{x}) \tag{54}$$



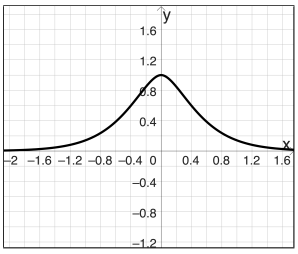
(a) A linear kernel



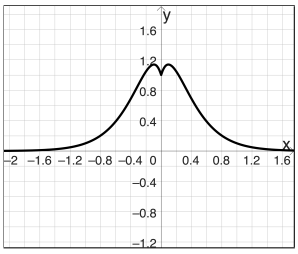
(b) An exponential kernel



(c) A Gaussian kernel



(d) Matérn kernel with $\nu = 3/2$



(e) Matérn kernel with $\nu = 5/2$

Figure 1: Several examples of common kernels.

Some of the common forms for universal Kriging include the linear and quadratic models given by

$$\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{M} \beta_i x_i \tag{55}$$

and

$$\boldsymbol{\beta}^T \boldsymbol{f}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{M} \beta_i x_i + \sum_{i=1}^{M} \sum_{j=1}^{M} \beta_{ij} x_i x_j, \tag{56}$$

respectively.

Another novel form of Kriging trend uses a set of orthogonal polynomial basis functions, corresponding to a polynomial chaos expansion (PCE) – a commonly used form of surrogate model for uncertainty quantification discussed next. This PCE Kriging method is described in detail in [7].

3 Polynomial Chaos Expansions

The polynomial chaos expansion (PCE) aims to develop a surrogate model for the transformation $y = h(\boldsymbol{x})$ by constructing an approximation through a series expansion in orthogonal polynomials. The following will build the basic concept and show how it is used for uncertainty quantification.

3.1 General Setting

Consider an m-dimensional random vector $\boldsymbol{X}$ having independent components defined through the joint probability density function $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{m} f_{X_i}(x_i)$. The PCE approximates the transformation $Y = h(\boldsymbol{X})$ as

$$Y = h(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^m} \beta_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \tag{57}$$

where $\beta_{\boldsymbol{\alpha}}$ are a set of deterministic coefficients, $\boldsymbol{\alpha}$ is a set of integers called the *multi-index* (more on this later), and $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ are multivariate orthogonal polynomials expressed as a function of a set of independent random variables $\boldsymbol{\xi}$ referred to as the *germ*. These multivariate polynomials are orthogonal with respect to the probability density of the germ $p_\xi(\xi)$ as

$$\langle \psi_j, \psi_k \rangle = \int \psi_j(\xi)\psi_k(\xi)p_\xi(\xi)d\xi = \delta_{jk} \tag{58}$$

where δ_{jk} is the Kronecker delta with $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ if $j \neq k$. The multi-variate polynomial basis $\Psi(\boldsymbol{\xi})$ are then constructed as a tensor product of the univariate orthogonal polynomials as

$$\Psi(\boldsymbol{\xi}) = \prod_{i=1}^{M} \psi_{\alpha_1}(\xi_i) \tag{59}$$

To make the construction of the PCE practical, the summation in Eq. (57) needs to be truncated to include only a finite number of terms P. This is most commonly achieved by truncating such that only terms whose total degree is less than or equal to a value p, i.e. $|\boldsymbol{\alpha}| \leq p$. This truncation defines a multi-index set

$$\mathcal{A}^{M,p} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^M : |\boldsymbol{\alpha}| = \sum_{i=1}^{M} \alpha_i \leq p \right\}. \tag{60}$$

whose cardinality is given by

$$\text{card}\, \mathcal{A}^{M,p} = \frac{(M+p)!}{M!\, p!} \equiv P\,. \tag{61}$$

This basis set is referred to as the *total degree basis*. Here, we see that the number of basis functions (and hence the number of coefficients to solve) grows extremely fast with the dimension M and the polynomial order p. To reduce the number of basis functions and improve scalability, Blatman and Sudret [8] proposed the so-called *hyperbolic truncation* scheme such that the basis set is defined by

$$\mathcal{A}^{M,p,q} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^M : ||\boldsymbol{\alpha}||_q \equiv \left(\sum_{i=1}^{M} \alpha_i^q \right)^{1/q} \leq p \right\}. \tag{62}$$

In this scheme, only the basis vectors are retained having a specified L_q norm where q is a parameter of the scheme. In the case where $q = 1$, the truncation corresponds to the total degree basis. When $q < 1$, the scheme truncates terms that have higher-order interactions. This can dramatically reduce the cardinality of the basis set for high-dimensional problems with high polynomial order (i.e. M and p large). This scheme is illustrated in Figure Create a figure like Figures 1 and 2 in [8] - Figures 2 and 3 are based on this reference.

With this truncated basis set, we can now express the PCE in terms of a finite number of terms with some truncation error ϵ as:

$$Y = h(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \beta_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) + \epsilon. \tag{63}$$

The objective then in PCE is to determine the coefficients $\boldsymbol{\beta}$ that minimize the truncation error ϵ. This can be viewed as a classical regression problem, and approaches to solve this regression problem will be discussed below. However, first we will discuss the selection of orthogonal polynomials for problems with different distribution types.

Add single index notation.

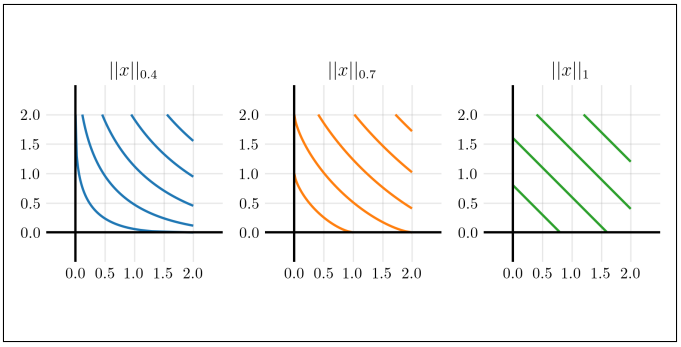


Figure 2: The level sets of the 0.4-norm, 1-norm, and 2-norm.Updated norms to 0.4, 0.7, and 1

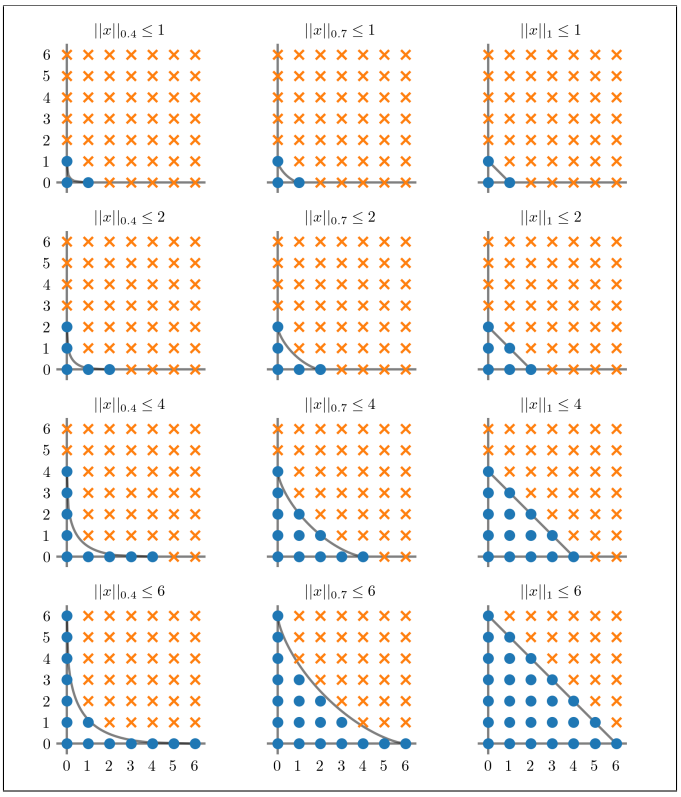


Figure 3: Truncation schemes for various norms and thresholds.Updated norms to 0.4, 0.7, and 1

3.1.1 Wiener-Askey Scheme

The specific orthogonal polynomials are selected according the Wiener-Askey scheme as detailed in [9]. The Wiener-Askey scheme specifically specifies the following polynomials corresponding to germs of different distributions:

- *Gaussian–Hermite*: When the germ ξ follows a standard normal distribution, the orthogonal polynomials are the (probabilists) Hermite polynomials $H_n(x)$. The Hermite polynomials are orthogonal

with respect to the following standard normal weight function

$$w(x) = e^{-\frac{x^2}{2}} \tag{64}$$

satisfying

$$\int_{-\infty}^{\infty} H_m(x)H_n(x)\, w(x)\, dx = 0 \quad \forall m \neq n. \tag{65}$$

We therefore have the following orthogonality condition

$$\int_{-\infty}^{\infty} H_m(x)H_n(x)\, e^{-\frac{x^2}{2}}\, dx = \sqrt{2\pi}\, n!\, \delta_{nm}, \tag{66}$$

Hence, the Hermite polynomials are orthogonal with respect to the standard normal probability density function. Additional details on the Hermite Polynomials can be found in Appendix A.1.

- *Beta–Jacobi* When the germ ξ follows the beta distribution, the orthogonal polynomials are the Jacobi polynomials, $P_n^{(\alpha,\beta)}(x)$. The Jacobi polynomials are orthogonal with respect to the following weight function

$$w(x) = (1-x)^{\alpha}(1+x)^{\beta} \tag{67}$$

on the interval $[-1, 1]$ satisfying

$$\int_{-1}^{1} P_m^{(\alpha,\beta)}(x)P_n^{(\alpha,\beta)}(x)\, w(x)\, dx = 0 \quad \forall m \neq n. \tag{68}$$

We have the following orthogonality condition

$$\int_{-1}^{1} (1-x)^{\alpha}(1+x)^{\beta} P_m^{(\alpha,\beta)}(x)P_n^{(\alpha,\beta)}(x)\, dx = \frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1}\frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(n+\alpha+\beta+1)n!}\delta_{nm}, \quad \alpha,\ \beta > -1. \tag{69}$$

The Beta distribution has probability density function given by

$$f(x;\alpha,\beta) = \frac{1}{(b-a)^{\alpha+\beta+1}\mathrm{B}(\alpha+1,\beta+1)}(x-a)^{\beta}(b-x)^{\alpha} \tag{70}$$

where $\mathrm{B}(p,q)$ is the *beta function* defined by

$$\mathrm{B}(p,q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)} \tag{71}$$

where $\Gamma(\cdot)$ is the Gamma function.

From these expressions, we can see that the weight function $w(x)$ in Eq. (67) is a special scaled case of the Beta distribution defined on the interval $[-1, 1]$. Hence, the Jacobi polynomials are orthogonal with respect to the Beta distribution on $[-1, 1]$.

- *Uniform–Legendre* The special case of the Beta distribution with $\alpha = \beta = 0$ corresponds to the uniform distribution and the Jacobi polynomials become the Legendre Polynomials, $P_n(x)$. Therefore, when the germ ξ follows the uniform distribution $\sim U(-1, 1)$, the orthogonal polynomials are the Legendre polynomials, which are orthogonal with respect to the following weight function

$$w(x) = 1 \tag{72}$$

satisfying

$$\int_{-1}^{1} P_m(x)P_n(x)\,w(x)\,dx = 0 \quad \forall m \neq n. \tag{73}$$

We therefore have the following orthogonality condition

$$\int_{-1}^{1} P_m(x)P_n(x)\,w(x)\,dx = \frac{2}{2n+1}\,\delta_{nm} \tag{74}$$

- *Gamma–Generalized Laguerre*

  When the germ ξ follows the Gamma distribution, the orthogonal polynomials are the Generalized Laguerre polynomials, $L_n^{(\alpha)}(x)$. The Generalized Laguerre polynomials are orthogonal with respect to the following weight function

$$w(x) = x^{\alpha}e^{-x} \tag{75}$$

  on the interval $[0, \infty)$ satisfying

$$\int_{0}^{\infty} L_m^{(\alpha)}(x)L_n^{(\alpha)}(x)\,w(x)\,dx = 0 \quad \forall m \neq n. \tag{76}$$

  We have the following orthogonality condition

$$\int_{0}^{\infty} L_m^{(\alpha)}(x)L_n^{(\alpha)}(x)\,x^{\alpha}e^{-x}\,dx = \frac{\Gamma(n+\alpha+1)}{n!}\delta_{mn}, \quad \alpha > -1 \tag{77}$$

  The Gamma distribution has probability density function given by

$$f(x;\alpha,\beta) = \frac{x^{\alpha}e^{-x/\beta}}{\beta^{\alpha+1}\Gamma(\alpha+1)}, \quad \alpha > -1, \beta > 0 \tag{78}$$

  where $\Gamma(\cdot)$ is the Gamma function.

  Regardless of the scale parameter β and constant $\Gamma(\alpha+1)$, this density has the same form as the weight function and therefore the Generalized Laguerre polynomials are orthogonal with respect to the Gamma distribution.

- *Exponential–Laguerre*

  The special case of the Gamma distribution with $\alpha = 0$ corresponds to the exponential distribution. When the germ ξ follows the exponential distribution, the orthogonal polynomials are the Laguerre polynomials, $L_n(x)$. The Laguerre polynomials are orthogonal with respect to the following weight function

$$w(x) = e^{-x} \tag{79}$$

  on the interval $[0, \infty)$ satisfying

$$\int_{0}^{\infty} L_m(x)L_n(x)\,w(x)\,dx = 0 \quad \forall m \neq n. \tag{80}$$

  We have the following orthogonality condition

$$\int_{0}^{\infty} L_m(x)L_n(x)e^{-x}\,dx = (n!)^2\delta_{mn} \tag{81}$$

The Gamma distribution has probability density function given by

$$f(x;\alpha,\beta) = \frac{x^{\alpha}e^{-x/\beta}}{\beta^{\alpha+1}\Gamma(\alpha+1)}, \quad \alpha > -1, \beta > 0 \tag{82}$$

where $\Gamma(\cdot)$ is the Gamma function.

Hence, the Laguerre polynomials are orthogonal with respect to the exponential probability density function. Additional details on the Laguerre Polynomials can be found in Appendix A.5.

3.2 Regression Methods

Projection-based Approaches

Starting from the PCE in Eq. (57) and allowing that the germ corresponds to the vector of input random variables, $\boldsymbol{X} = \boldsymbol{\xi}$, we multiply by $\Psi_{\boldsymbol{\alpha}'}(\boldsymbol{X})$ and take the expectation, which yields

$$\beta_{\boldsymbol{\alpha}} = E[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \cdot h(\boldsymbol{X})] \tag{83}$$

by orthonarmality of the polynomials. This corresponds to the projection of the model $h(\boldsymbol{X})$ onto the set of orthonormal polynomials $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$ and simplifies the estimation of the coefficients to the estimation of an expected value. Monte Carlo estimation would be inefficient. Instead, projection-based methods employ numerical integration schemes.

Gaussian Quadrature:
As the benchmark approach for numerical integration, Gaussian quadrature expresses the integral as a weighted sum of integrand evaluations at carefully selected quadrature points, $\boldsymbol{x}^{(i)}$. That is, the expectation is approximated by:

$$\begin{aligned} \beta_{\boldsymbol{\alpha}} &= \mathbb{E}[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \cdot h(\boldsymbol{X})] \\ &= \int_{\Omega_{\boldsymbol{X}}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) h(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} \\ &\approx \sum_{i=1}^{N} w^{(i)} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}^{(i)}) h(\boldsymbol{x}^{(i)}) \end{aligned} \tag{84}$$

where the weights $w^{(i)}$ and quadrature points $\boldsymbol{x}^{(i)}$ are determined by the chosen polynomials $\Psi_{\boldsymbol{\alpha}}(x)$ and their order, and hence by the probability distributions of X_i following the Weiner-Askey scheme discussed above. In particular, the quadrature points $x^{(i)}$ in each dimension correspond to the $(p+1)$ roots of the p^{th}-order polynomial. Practically, the quadrature points and the weights are computed numerically using schemes such as the Golub-Welsch algorithm [10] or Newton's method for solving $\Psi_p(x) = 0$.

To perform this integration in multiple dimensions requires a tensor product grid of quadrature points. Since each dimension requires $(p+1)$ quadrature points, an M-dimensional integral requires $(p+1)^M$ quadrature points, which grows exponentially with dimension. Considering that the model evaluation $h(\boldsymbol{x})$ is required at each quadrature point, this becomes computationally very expensive as the number of model evaluations grows exponentially. This is a consequence of the so-called *curse of dimensionality*.

Sparse Quadrature:
To mitigate the curse of dimensionality, quadrature can be conducted on so-called sparse quadrature grids whose weights are derived by hierarchically combining lower-order univariate quadrature terms. First derived by Smolyak [11], numerous such methods have been developed including those that use piecewise linear interpolation and those using polynomial interpolation including the popular Clenshaw-Curtis grids

that use the Chebyshev Gauss-Lobatto quadrature points and schemes using the Gauss-Patterson – both of which use nested grids of sparse quadrature points. A detailed discussion of sparse quadrature is beyond the scope of these notes.

Least Squares

Introduced by Berveiller et al. [12], the most common approach to solving the PCE coefficients is by setting up the following least-squares problem. Recall the truncated PCE from Eq. (63) where the truncation error ϵ can be expressed as

$$\epsilon = h(\boldsymbol{X}) - \boldsymbol{\beta}^T \Psi(\boldsymbol{X}) \tag{85}$$

where $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_{P-1}]^T$, $P = \text{card } \mathcal{A}^{M,p}$ and $\Psi(\boldsymbol{X}) = \{[Psi_0(\boldsymbol{X}), \ldots, \Psi_{P-1}(\boldsymbol{X})]^T$ are vectors containing the PCE coefficients and orthogonal polynomials, respectively. In least-squares minimization, we aim to find the coefficients $\hat{\boldsymbol{\beta}}$ that minimize the expected mean square error of this truncation as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \mathbb{E}\left[\left(h(\boldsymbol{X}) - \boldsymbol{\beta}^T \Psi(\boldsymbol{X})\right)^2\right] \tag{86}$$

The standard approach to solving this minimization is referred to as *Ordinary Least Squares* (OLS). In OLS, we begin by evaluating the model $Y = h(\boldsymbol{X})$ and the basis polynomials $\Psi(\boldsymbol{X})$ at a set of N samples $\boldsymbol{\mathcal{X}} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}]^T$ (often referred to as the experimental design) to obtain the model responses $\mathcal{Y} = [y^{(1)}, y^{(2)}, \ldots, y^{(N)}]^T$ and the matrix of polynomial evaluations $\boldsymbol{\Psi}$ having terms $\Psi_{ij} = \Psi_j(\boldsymbol{x}^{(i)})$. The OLS problem can then be formulated practically as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \frac{1}{N} \sum_{i=1}^{N} \left[\boldsymbol{\beta}^T \boldsymbol{\Psi}(\boldsymbol{x}^i) - h(\boldsymbol{x}^{(\boldsymbol{i})})\right]^2 \tag{87}$$

The closed-form solution to the minimization problem is then determined by

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{\Psi}^T \boldsymbol{\Psi}\right)^{-1} \boldsymbol{\Psi}^T \mathcal{Y} \tag{88}$$

The advantage of OLS is that it allows the coefficients to be determined from an arbitrary set of samples of the random vector $\boldsymbol{X}$ as long as $N \geq P$.

Sparse Regression Methods

As summarized by Luthen et al. [13], there are several ways to formulate the regression problem to promote a sparse solution; that is the solution that requires the fewest number of polynomials, P. Strictly speaking, to promote sparsity we aim to minimize the number of non-zero coefficients β, which is commonly denoted by $\|\boldsymbol{\beta}\|_0 = \sum_i \boldsymbol{I}(\beta_i \neq 0)$ where $I(\cdot)$ is the identity function that takes value 1 if the condition is satisfied and zero otherwise. We can integrate this condition as a penalty into the optimizer by reformulating the minimization as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \mathbb{E}\left[\left(h(\boldsymbol{X}) - \boldsymbol{\beta}^T \Psi(\boldsymbol{X})\right)^2\right] + \lambda \|\boldsymbol{\beta}\|_0, \tag{89}$$

which is referred to as l_0-minimization. However, this optimization is non-convex and requires a combinatorial solution that is comptutationally intractable for most practical problems.

Instead, it is common to formulate the problem as an l_1-minimization problem to achieve convexity, wherein $\|\boldsymbol{\beta}\|_0$ is replaced with $\|\boldsymbol{\beta}\|_1 = \sum_i |\beta_i|$. There are several ways to formulate this l_1-minimization problem. Perhaps the most common is the *Lagrange formulation* given by

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \mathbb{E}\left[\left(h(\boldsymbol{X}) - \boldsymbol{\beta}^T \Psi(\boldsymbol{X})\right)^2\right] + \lambda \|\boldsymbol{\beta}\|_1, \tag{90}$$

Alternatively, the l_1-minimization can be formulated as a constrained optimization problem using either the *basis persuit* method where the l_1 norm itself is minimized and constrained such that the least squares problem produces a sufficiently small error as

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \arg\min_{\beta} \|\boldsymbol{\beta}\|_1 \\ &\text{s.t. } \mathbb{E}\left[\left(h(\boldsymbol{X}) - \boldsymbol{\beta}^T\Psi(\boldsymbol{X})\right)^2\right] \leq \sigma, \end{aligned} \tag{91}$$

where σ is an error threshold, or the *least absolute shrinkage and selection operator* (LASSO) where the l_2 norm is again minimized subject to a constraint that the l_1 norm must be smaller than a threshold τ as

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \arg\min_{\beta} \mathbb{E}\left[\left(h(\boldsymbol{X}) - \boldsymbol{\beta}^T\Psi(\boldsymbol{X})\right)^2\right] \\ &\text{s.t. } \|\boldsymbol{\beta}\|_1 \leq \tau, \end{aligned} \tag{92}$$

Numerous numerical methods have been devised to solve these sparse regression problems. Among the most widely-used are *Least Angle Regression* (LARS) [8] and its extension termed *Hybrid LARS with LOO-CV* [8], the *Orthogonal Matching Persuit* (OMP) algorithm, and the *Subspace Pursuit* (SP) method. All of these are greedy algorithms that iteratively add and remove basis vectors according to some criterion. Details of these algorithms will not be provided here. For an in-depth review, see Luthen et al. [13].

Bayesian Compressive Sensing

In Bayesian compressive sensing (BCS), or sparse Bayesian learning (SBL), the regression problem is reformulated as a Bayesian inverse problem in a special way to promote sparsity of the resulting expansion. Since we haven't covered Bayesian inversion in detail yet, we will provide only a brief overview of the method and will return to this at a later time.

The method starts by treating the training data $\mathcal{Y}$ as independent noisy realizations of the PCE model following a normal distribution with mean $\boldsymbol{\Psi\beta}$ and identical variances σ^2, i.e. $p(\mathcal{Y}) = N(\boldsymbol{\Psi\beta}, \sigma^2\boldsymbol{I})$ where $\boldsymbol{I}$ is the identity matrix. The variance σ is either a specified fixed value or a random variable whose distribution must be specified. The parameters of the PCE model, $\boldsymbol{\beta}$ are also treated as normal random variables with zero mean and variances γ_i (each coefficient has its own variance), i.e. $p(\beta_i) = N(0, \gamma_i)$. These variances are the so-called *hyperparameters* of the model and are also treated as random variables having specified distribution $p(\gamma_i)$. The goal of Bayesian inference in this setting is to learn the distribution $p(\boldsymbol{\beta}|\mathcal{Y})$ where the prior parameter distribution is given by $p(\boldsymbol{\beta}) = \int p(\boldsymbol{\beta}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})d\boldsymbol{\gamma}$. In BCS, certain forms of the parameter distributions, $p(\gamma_i)$, are selected such that the effective prior distribution $p(\boldsymbol{\beta})$ promotes sparsity. That is, it has a high weight near zero and heavy tails. Moreover, rather than seeking the full posterior distribution of the parameters $p(\boldsymbol{\beta}|\mathcal{Y})$, BCS seeks to identify only the *maximum a posterior* (MAP) estimate of the parameters $\boldsymbol{\beta}^{\text{MAP}}$, that is the estimate of $\boldsymbol{\beta}$ that maximizes $p(\boldsymbol{\beta}|\mathcal{Y})$. Thanks to these simplifications and the assumption that $p(\boldsymbol{\beta}|\boldsymbol{\gamma})$ and $p(\mathcal{Y}|\boldsymbol{\beta}, \sigma^2)$ are normal, many of the requisite calculations can be performed analytically.

3.3 Moment Estimation with PCE

The PCE has the added advantage that the moments of the function can be estimated directly from the coefficients. Consider the m^{th} moment about the origin given by

$$\mathbb{E}[Y^m] = \int [h(\boldsymbol{X})]^m \, p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \tag{93}$$

Applying the truncated PCE from Eq. (63) yields

$$\begin{aligned}\mathbb{E}[Y^m] &\approx \int \left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \beta_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right]^m p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \int \sum_{\boldsymbol{\alpha}_1\in\mathcal{A}} \cdots \sum_{\boldsymbol{\alpha}_m\in\mathcal{A}} \beta_{\boldsymbol{\alpha}_1}\cdots\beta_{\boldsymbol{\alpha}_m}\Psi_{\boldsymbol{\alpha}_1}(\boldsymbol{x})\cdots\Psi_{\boldsymbol{\alpha}_m}(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \sum_{\boldsymbol{\alpha}_1\in\mathcal{A}} \cdots \sum_{\boldsymbol{\alpha}_m\in\mathcal{A}} \beta_{\boldsymbol{\alpha}_1}\cdots\beta_{\boldsymbol{\alpha}_m}\int \Psi_{\boldsymbol{\alpha}_1}(\boldsymbol{x})\cdots\Psi_{\boldsymbol{\alpha}_m}(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}\end{aligned} \tag{94}$$

As we can see, moment estimation therefore requires only integration over the polynomial basis functions, which simplifies greatly thanks to their orthogonality with respect to the probability measure $p_{\boldsymbol{X}}(\boldsymbol{x})$. This can be see in the estimation of the first two moments such that the mean value is computed by

$$\mathbb{E}[Y] \approx \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \beta_{\boldsymbol{\alpha}}\int \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \tag{95}$$

Recall that $\Psi_0(\boldsymbol{x}) = 1$, this expression can be rewritten as

$$\mathbb{E}[Y] \approx \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \beta_{\boldsymbol{\alpha}}\int \Psi_0(\boldsymbol{x})\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \tag{96}$$

Thanks to the orthonormality of the polynomials, this simplifies to

$$\mathbb{E}[Y] \approx \beta_{\mathbf{0}} \tag{97}$$

Similarly, the second moment can be estimated by

$$\begin{aligned}\mathbb{E}[Y^m] &\approx \int \left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}} \beta_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right]^2 p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \sum_{\boldsymbol{\alpha}_1\in\mathcal{A}}\sum_{\boldsymbol{\alpha}_2\in\mathcal{A}} \beta_{\boldsymbol{\alpha}_1}\beta_{\boldsymbol{\alpha}_2}\int \Psi_{\boldsymbol{\alpha}_1}(\boldsymbol{x})\Psi_{\boldsymbol{\alpha}_2}(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \beta_{\boldsymbol{\alpha}}^2\int \left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right]^2 p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \beta_{\boldsymbol{\alpha}}^2\end{aligned} \tag{98}$$

Recall that the variance is given by $\sigma_Y^2 = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2$. Therefore, we have

$$\begin{aligned}\sigma_Y^2 &\approx \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \beta_{\boldsymbol{\alpha}}^2 - \beta_{\mathbf{0}}^2 \\ &\approx \sum_{\substack{\boldsymbol{\alpha}\in\mathcal{A}\\ \boldsymbol{\alpha}\neq\mathbf{0}}} \beta_{\boldsymbol{\alpha}}^2\end{aligned} \tag{99}$$

That is, the variance can be obtained simply by the sum of the squares of all of the coefficients in the model excluding the intercept $\beta_{\mathbf{0}}$.

Using Eq. (94), the higher-order moments of the model output can also be determined, but this requires the computation of higher-order products of the orthogonal polynomials, which are not trivial to evaluate and are therefore rarely used in practice. Nonetheless, expressions have been derived for third- and fourth-order relations for certain classes of polynomials (e.g. Hermite, Jacobi and generalized Laguerre) [14].

A closely related topic is the estimation of variance based sensitivity indices. This will be discussed in further detail in the following module on Global Sensitivity Analysis.

4 Other Surrogate Models

Numerous other surrogate models have been developed in practice, but are not as widely used as the GP and PCE surrogates. These include, but are not limited to, surrogate models based on radial basis functions, support vector regression, artificial neural networks, and polynomial regressions. These methods, though sometimes used in practice, generally do not have the same advantages as GPs and PCEs and are therefore less commonly used for UQ applications. We therefore will not explore them in further detail.

A Definitions and Properties of Orthogonal Polynomials

Orthogonal polynomials in the Askey Scheme are defined through the *generalized hypergeometric series* denoted ${}_pF_q(a_1,\dots,a_p;b_1,\dots,b_q;x)$. First, denoting the rising factorial through the Pochhammer symbol

$$\begin{aligned}(a)_0 &= 1,\\ (a)_n &= a(a+1)(a+2)\cdots(a+n-1), \quad n\geq 1\end{aligned} \tag{100}$$

the generalized hypergeometric series is expressed:

$${}_pF_q(a_1,\dots,a_p;b_1,\dots,b_q;x) = \sum_{k=0}^{\infty}\frac{(a_1)_k\cdots(a_p)_k}{(b_1)_k\cdots(b_q)_k}\,\frac{x^k}{k!}. \tag{101}$$

If any a_j is a non-positive integer, i.e. $a_j=-n$, then the series only has a finite number of terms and is, in fact, a polynomial of degree n given by

$${}_pF_q(a_1,\dots,a_p;b_1,\dots,b_q;x) = \sum_{k=0}^{n}\frac{(a_1)_k\cdots(a_p)_k}{(b_1)_k\cdots(b_q)_k}\,\frac{x^k}{k!}. \tag{102}$$

All orthogonal polynomials $\{Q_n(x)\}$ also satisfy a three-term recurrence relation given by;

$$-xQ_n(x) = A_nQ_{n+1}(x) - (A_n+C_n)Q_n(x) + C_nQ_{n-1}(x), \quad n\geq 1 \tag{103}$$

where $A_n, C_n \neq 0$ and $C_n/A_{n-1}>0$. Given that $Q_{-1}(x)=0$ and $Q_0(x)=1$, all $Q_n(x)$ can be determined from the recurrence relation.

All orthogonal polynomials can also be determined by repeatedly applying the differential operator as:

$$Q_n(x) = \frac{1}{w(x)}\frac{d^n}{dx^n}[w(x)s^n(x)] \tag{104}$$

where $w(x)$ is the orthogonal weighting function and $s(x)$ is a polynomial of at most second order. This formula is referred to as the *generalized Rodrigues' formula*.

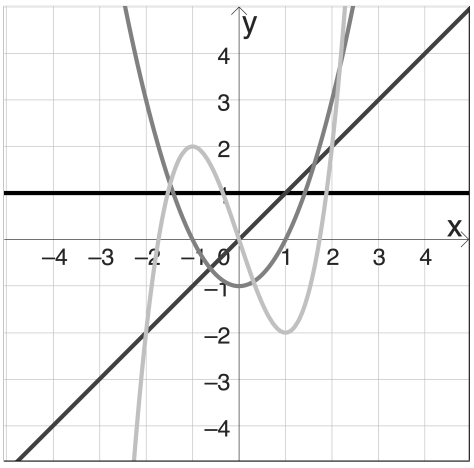
For each of the classes of orthogonal polynomials discussed above the in Askey scheme, we will provide the definition in terms of its generalized hypergeometric series, its recurrence relations, and its generalized Rodrigues' formula. We will also provide the first five orthogonal polynomials.

A.1 Hermite Polynomials

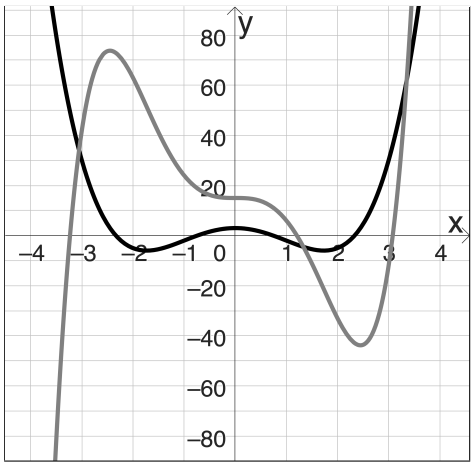
Hypergeometric Series Definition

The Hermite polynomials are defined in terms of the following hypergeometric series:

$$H_n(x) = (2x)^n\,{}_2F_0\left(-\frac{n}{2},-\frac{n-1}{2};-\frac{1}{x^2}\right) \tag{105}$$



(a) The Hermite polynomials H_0, H_1, H_2, H_3



(b) The Hermite polynomials H_4 and H_5

Figure 4: The first six Hermite polynomialsUploaded figure

Rodrigues' Formula

$$H_n(x) = (-1)^n e^{\frac{x^2}{2}} \frac{d^n}{dx^n} e^{-\frac{x^2}{2}}, \tag{106}$$

where $n = 0, 1, 2, \ldots$ is the order of the polynomial.

Recurrence Relations

$$H_{n+1}(x) = xH_n(x) - nH_{n-1}(x). \tag{107}$$

from which we can obtain the first six Hermite polynomials as

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= x \\ H_2(x) &= x^2 - 1 \\ H_3(x) &= x^3 - 3x \\ H_4(x) &= x^4 - 6x^2 + 3 \\ H_5(x) &= x^5 - 10x^3 + 15x \end{aligned} \tag{108}$$

which are plotted in Figure 4.

A.2 Jacobi Polynomials

A.2.1 Hypergeometric Series Definition

The Jacobi polynomials are defined in terms of the following hypergeometric series:

$$P_n^{(\alpha,\beta)}(x) = \frac{(\alpha+1)_n}{n!} \, {}_2F_1\left(-n, 1+\alpha+\beta+n; \alpha+1; \tfrac{1}{2}(1-x)\right), \tag{109}$$

A.2.2 Rodrigues' Formula

$$P_n^{(\alpha,\beta)}(x) = \frac{(-1)^n}{2^n n!}(1-x)^{-\alpha}(1+x)^{-\beta} \frac{d^n}{dx^n}\left\{(1-x)^{\alpha}(1+x)^{\beta}\left(1-x^2\right)^n\right\}. \tag{110}$$

where α and β are parameters of the model.

A.2.3 Recurrence Relations

For a given α and β the Jacobi polynomials can be determined from

$$\begin{aligned} 2n(n+\alpha+\beta)(2n+\alpha+\beta-2)P_n^{(\alpha,\beta)}(x) = \\ (2n+\alpha+\beta-1)\Big\{(2n+\alpha+\beta)(2n+\alpha+\beta-2)x+\alpha^2-\beta^2\Big\}P_{n-1}^{(\alpha,\beta)}(x)- \\ 2(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)P_{n-2}^{(\alpha,\beta)}(x), \end{aligned} \tag{111}$$

from which we can obtain the first few Jacobi polynomials as

$$\begin{aligned} P_0^{(\alpha,\beta)}(x) &= 1 \\ P_1^{(\alpha,\beta)}(x) &= \frac{1}{2}[2(\alpha+1)+(\alpha+\beta+2)(x-1)] \\ P_2^{(\alpha,\beta)}(x) &= \frac{1}{8}[4(\alpha+1)(\alpha+2)+4(\alpha+\beta+3)(\alpha+2)(x-1)+(\alpha+\beta+3)(\alpha+\beta+4)(x-1)^2] \end{aligned} \tag{112}$$

which are plotted in Figure 5.

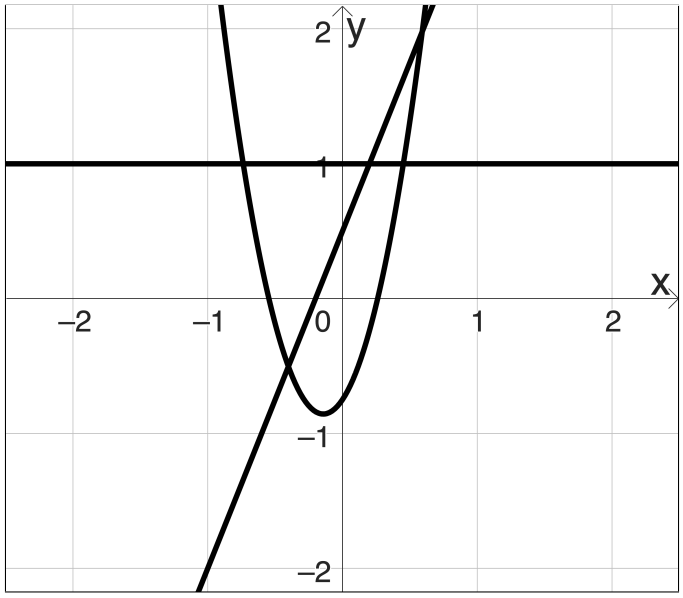


Figure 5: The first three Jacobi polynomials with $\alpha = 2, \beta = 1$.Uploaded figure.

A.3 Legendre Polynomials

A.3.1 Hypergeometric Series Definition

$$P_n(x) = P_n^{(0,0)}(x) = {}_2F_1\left(-n, n+1; 1; \tfrac{1}{2}(1-x)\right) \tag{113}$$

A.3.2 Rodrigues' Formula

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n\,. \tag{114}$$

A.3.3 Recurrence Relations

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \tag{115}$$

from which we can obtain the first five Legendre polynomials as

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= \tfrac{1}{2}\left(3x^2 - 1\right) \\ P_3(x) &= \tfrac{1}{2}\left(5x^3 - 3x\right) \\ P_4(x) &= \tfrac{1}{8}\left(35x^4 - 30x^2 + 3\right) \\ P_5(x) &= \tfrac{1}{8}\left(63x^5 - 70x^3 + 15x\right) \end{aligned} \tag{116}$$

which are plotted in Figure 6.

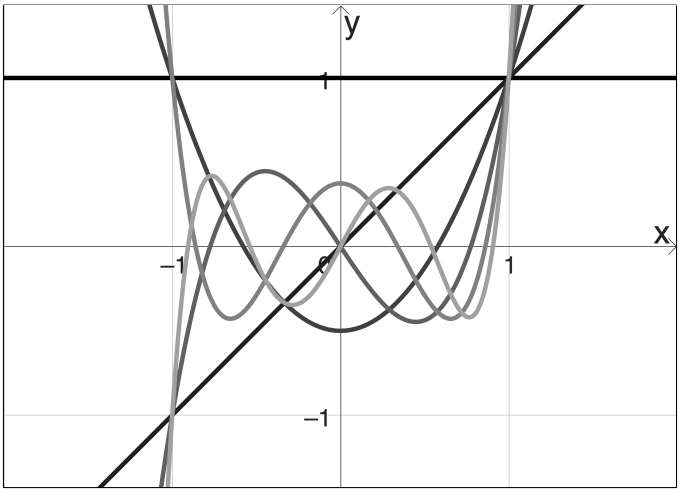


Figure 6: The first six Legendre Jacobi polynomials.Uploaded figure.

A.4 Generalized Laguerre Polynomials

A.4.1 Hypergeometric Series Definition

$$L_n^{(\alpha)}(x) = \binom{n+\alpha}{n} M(-n, \alpha+1, x) = \frac{(\alpha+1)_n}{n!}\, {}_1F_1(-n, \alpha+1, x) \tag{117}$$

A.4.2 Rodrigues' Formula

$$L_n^{(\alpha)}(x) = \frac{x^{-\alpha}e^x}{n!}\frac{d^n}{dx^n}\left(e^{-x}x^{n+\alpha}\right). \tag{118}$$

A.4.3 Recurrence Relations

$$(n+1)L_{n+1}^{(\alpha)}(x) = (2n+\alpha+1-x)L_n^{(\alpha)}(x) - (n+\alpha)L_{n-1}^{(\alpha)}(x) \tag{119}$$

from which we can obtain the first few Generalized Laguerre polynomials as

$$\begin{aligned} L_0^{(\alpha)}(x) &= 1 \\ L_1^{(\alpha)}(x) &= -x + (\alpha+1) \\ L_2^{(\alpha)}(x) &= \frac{x^2}{2} - (\alpha+2)x + \frac{(\alpha+1)(\alpha+2)}{2} \\ L_3^{(\alpha)}(x) &= \frac{-x^3}{6} + \frac{(\alpha+3)x^2}{2} - \frac{(\alpha+2)(\alpha+3)x}{2} + \frac{(\alpha+1)(\alpha+2)(\alpha+3)}{6} \end{aligned} \tag{120}$$

which are plotted in Figure 7.

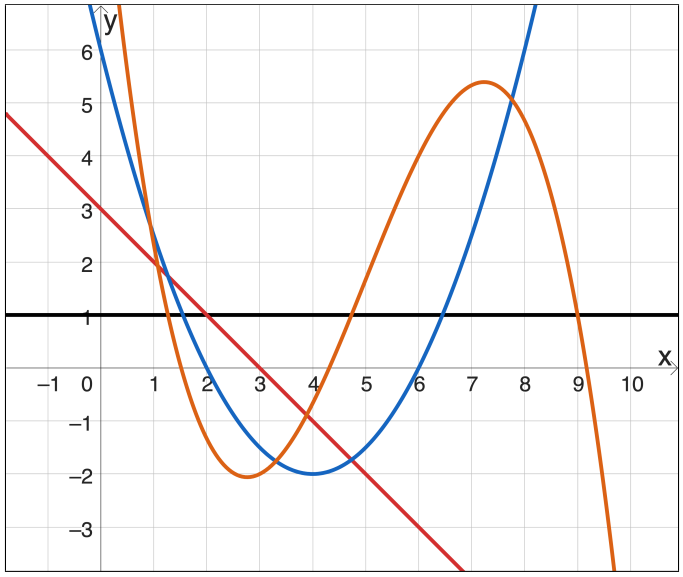


Figure 7: The first three Generalized Laguerre polynomials $L_0^{(2)}(x), L_1^{(2)}(x), L_2^{(2)}(x)$ with $\alpha = 2$.Uploaded figure.

A.5 Laguerre Polynomials

A.5.1 Hypergeometric Series Definition

The Laguerre polynomials can be determined from the hypergeometric series in Eq. (117) by setting $\alpha = 0$.

A.5.2 Rodrigues' Formula

$$L_n(x) = \frac{e^x}{n!}\frac{d^n}{dx^n}\left(e^{-x}x^n\right), \tag{121}$$

A.5.3 Recurrence Relations

$$L_{k+1}(x) = \frac{(2k+1-x)L_k(x) - kL_{k-1}(x)}{k+1}, \tag{122}$$

from which we can obtain the first five Laguerre polynomials as

$$\begin{aligned} L_0(x) &= 1 \\ L_1(x) &= 1 - x \\ L_2(x) &= \tfrac{1}{2}(x^2 - 4x + 2) \\ L_3(x) &= \tfrac{1}{6}(-x^3 + 9x^2 - 18x + 6) \\ L_4(x) &= \tfrac{1}{24}(x^4 - 16x^3 + 72x^2 - 96x + 24) \\ L_5(x) &= \tfrac{1}{120}(-x^5 + 25x^4 - 200x^3 + 600x^2 - 600x + 120) \end{aligned} \tag{123}$$

which are plotted in Figure 8.Should all six of these be on one figure?

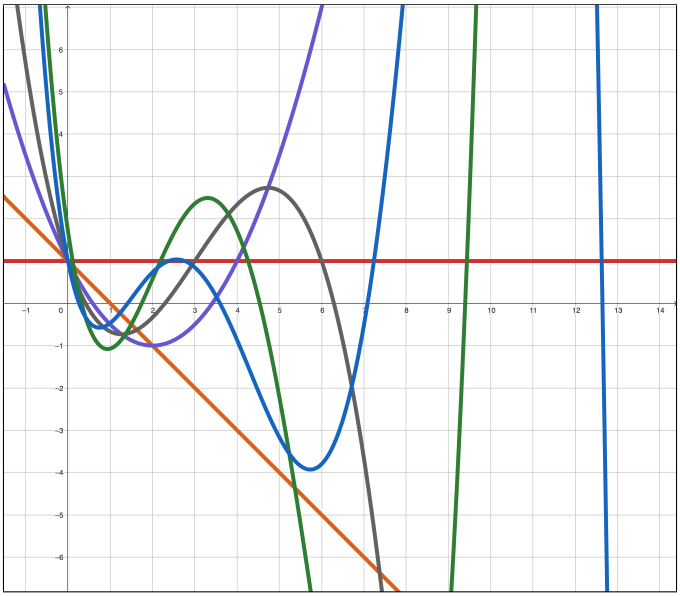


Figure 8: The first six Laguerre polynomials.Uploaded figure. I think this figure is very busy so there is also an option to separate this into two figures.

References

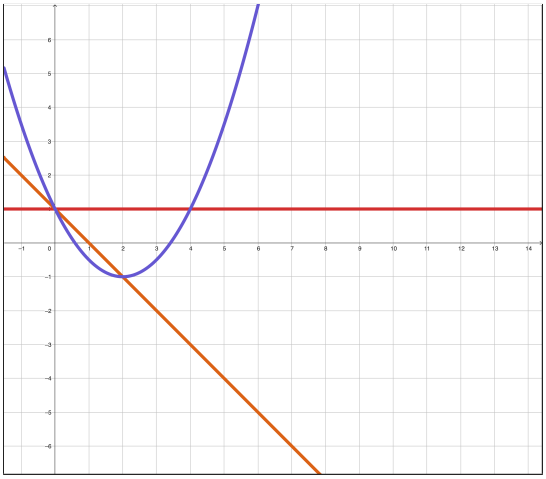
[1] C. E. Rasmussen, C. K. Williams, et al., Gaussian processes for machine learning, Vol. 1, Springer, 2006.

[2] D. G. Krige, A statistical approach to some mine valuation and allied problems on the witwatersrand: By dg krige, Ph.D. thesis, University of the Witwatersrand (1951).

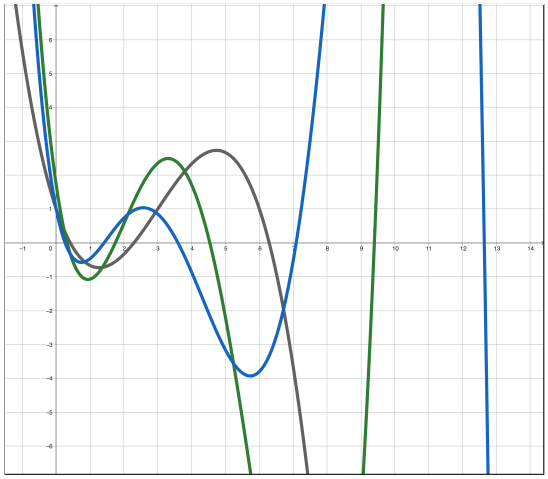
[3] G. Matheron, Principles of geostatistics, Economic geology 58 (8) (1963) 1246–1266.

[4] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, The design and analysis of computer experiments, Statistical Science 4 (1989).

[5] T. J. Santner, B. J. Williams, W. I. Notz, B. J. Williams, The design and analysis of computer experiments, Vol. 1, Springer, 2003.



(a) The Laguerre polynomials L_0, L_1 and L_2.



(b) The Laguerre polynomials L_3, L_4 and L_5.

Figure 9: The first six Hermite polynomialsUploaded figure.

[6] F. Bachoc, Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification, Computational Statistics & Data Analysis 66 (2013) 55–69.

[7] R. Schobi, B. Sudret, J. Wiart, Polynomial-chaos-based kriging, International Journal for Uncertainty Quantification 5 (2) (2015).

[8] G. Blatman, B. Sudret, Adaptive sparse polynomial chaos expansion based on least angle regression, Journal of computational Physics 230 (6) (2011) 2345–2367.

[9] D. Xiu, G. E. Karniadakis, The wiener–askey polynomial chaos for stochastic differential equations, SIAM journal on scientific computing 24 (2) (2002) 619–644.

[10] G. H. Golub, J. H. Welsch, Calculation of gauss quadrature rules, Mathematics of computation 23 (106) (1969) 221–230.

[11] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, in: Doklady Akademii Nauk, Vol. 148, Russian Academy of Sciences, 1963, pp. 1042–1045.

[12] M. Berveiller, B. Sudret, M. Lemaire, Stochastic finite element: a non intrusive approach by regression, European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique 15 (1-3) (2006) 81–92.

[13] N. Luthen, S. Marelli, B. Sudret, Sparse polynomial chaos expansions: Literature survey and benchmark, SIAM/ASA Journal on Uncertainty Quantification 9 (2) (2021) 593–649.

[14] L. Novák, On distribution-based global sensitivity analysis by polynomial chaos expansion, Computers & Structures 267 (2022) 106808.

Nomenclature

Functions

$P(\cdot)$ Probability measure

$f_X(\cdot)$ Probability density function (PDF) of a random variable X

$f_{\boldsymbol{X}}(\cdot)$ Joint probability density function of the random vector $\boldsymbol{X}$

$F_X(\cdot)$ Cumulative distribution function (CDF) of a random variable X

$F_{\boldsymbol{X}}(\cdot)$ Joint Cumulative Distribution Function of the random vector $\boldsymbol{X}$

$R_{XX}(\cdot)$ Autocorrelation function of the random process $X(\cdot)$ This is the previously used autocorrelation function notation

$\mathcal{R}(\cdot)$ Autocorrelation function of the random process $X(\cdot)$ This notation was introduced in this module. Which is preferred?

$\mathbb{E}[\cdot]$ Expected value of a random variable. Also denoted $\mu_X \triangleq E[X]$

$\boldsymbol{F}$ Matrix of basis functions

$\boldsymbol{f}(\cdot)$ Vector of basis functions

$\boldsymbol{R}$ Correlation matrix of points n the training set

$\boldsymbol{r}(\cdot)$ Vector of correlations

δ_{jk} Kronecker delta function

$\Gamma(\cdot)$ The Gamma function

$\lceil \cdot \rceil$ Ceiling function

$\mathcal{F}(\cdot)$ Regression model

$\mathcal{K}_\nu$ Modified Bessel function of the second kind

$\mathcal{L}(\cdot)$ Likelihood function

$\mathcal{Y}(\cdot)$ Approximation of a stochastic process

$\mathrm{B}(\cdot)$ Beta function

$\Psi_{\boldsymbol{\alpha}}(\cdot)$ Multivariate orthogonal polynomials

$\mathrm{diag}(\cdot)$ The diagonalizing function

$H_n(\cdot)$ Hermite polynomial of order n

$I(\cdot)$ Identity function

$K(\cdot)$ Kernel function

$L_n(\cdot)$ Laguerre polynomial

$L_n^{(\alpha)}(\cdot)$ Generalized Laguerre polynomial

$P_n^{(\alpha,\beta)}(\cdot)$ Jacobi polynomial

$_pF_q$ Hypergeometric series

Operators

$\langle \cdot, \cdot \rangle$ Inner product

Variables

$\boldsymbol{\beta}$ Vector of regression coefficients

$\boldsymbol{X}$ A random vector in $\mathbb{R}^n$

$\boldsymbol{\alpha}$ Multi-index

$\mathbf{\Sigma}_\mathrm{n}$ Covariance matrix Check if this matches other notation

$\boldsymbol{\theta}$ Parameters of the model (or function) f

ϵ Truncation error Check if epilson is used for error elsewhere

$\mathcal{A}^{M,p}$ Multi-index set

$\mathcal{C}^0$ The set of continuous functions

$\mathcal{D}$ Set of training data

σ_X Standard deviation of the random variable X

ξ Germ

X A random variable

Ω Sample space

ω A random event from the sample space Ω