Derivation of Gaussian Process Regression Technique

A. Ho

June 14, 2018

In order to properly discuss the foundations of *Gaussian process regression (GPR)*, it is necessary to split the topic into two separate fields and rejoin them once fully developed. These two topics are: *linear regression theory*, discussed in Section 1, and *Bayesian probability theory*, discussed in Section 2. These are then combined to derive the GPR predictive equations in Section 3, with a practical summary given in Section 3.3.

1 Basic Regression Theory

The first topic, regression theory, describes the set of statistical tools required to estimate the relationship of one quantity with respect to another, in order to predict or forecast unknown quantities of a system based on known ones. In order to perform any basic sort of regression, a *model basis* must first be defined which has the potential of describing the observations, such as the following:

$$Y = \mathbf{\Phi}(X, \beta) + \varepsilon \tag{1}$$

where (X, Y) represents the set of input data points, $\mathbf{\Phi}$ represents the set of functions used to form the model basis, called *basis functions*, β represents the vector of free parameters in the problem and ε represents the *residuals* of the model, or the difference between the value predicted by the model and the input data point. It should be noted that, for a d-dimensional problem space, X, Y, and ε are all $n \times d$ vectors, and $\mathbf{\Phi}$ is $n \times m$ matrix, where m is the number of functions in the model basis and n is the number of input data points and $m \leq n$ in order for a solution to exist. For reasons of simplicity, this outline will show the case where $d = 1$ although the equations are kept as general as possible. Once the basis functions are chosen, the weights can be adjusted until $\varepsilon = 0$ to optimize the model such that it best describes the input data.

However, depending on the number and variety of the functions chosen in the model basis, it may be impossible or extremely time-consuming to find β such that $\varepsilon = 0$. Thus, it is generally advantageous to define a *goodness-of-fit* metric, which allows a solution to be imperfect while simultaneously providing some information on how well the model describes the input data.

1.1 Least-Squares Regression

The most commonly used form of regression is the *least-squares* regression technique, in which the goodness-of-fit metric is the *sum-squared (SS) error*, calculated from the residuals as follows:

$$M_{\text{SS}} = \sum_i^n \varepsilon_i^2 \tag{2}$$

where i denotes the individual elements of the vector, ε, and n denotes the total number of elements in ε. It should be noted that the SS error is related to the more commonly-known *root-mean-squared (RMS) error*, which is calculated as follows:

$$M_{\text{RMS}} = \sqrt{\frac{M_{\text{SS}}}{n}} \tag{3}$$

Using either of these metric definitions, it can be seen that the desired solution is the combination of weights which minimize the metric. For reasons of simplicity, this outline will use the SS error, as given in Equation (2), as the metric. By noting that the derivative of a quantity is zero at its minima or maxima, the desired solution, β_*, can be mathematically described as the set of parameters which satisfies the following equation:

$$\nabla_\beta M_{\text{SS}} = 2 \sum_i^n \varepsilon_i \frac{\partial \varepsilon_i}{\partial \beta} = 0 \tag{4}$$

where the second form was obtained by substituting of Equation (2) into Equation (4). All least-squares regression methods use the gradients obtained via Equation (4), but the calculations to translate it into a solution for β can vary significantly.

1.2 Linear Least-Squares (LL) Regression

From Equation (4), if $\mathbf{\Phi}$ is composed such that it is linear in β, ie. all of the basis functions are modified as multiples of the various elements of β, then it becomes possible to separate the contributions of β from $\mathbf{\Phi}$ in Equation (1), as follows:

$$Y = \mathbf{\Phi}(X)\,\beta + \varepsilon \tag{5}$$

where β becomes a $m \times 1$ vector, with each element corresponding to a single basis function. Then, by substituting Equation (5) into Equation (4) and reorganizing the terms back into matrix form, the minimization solution can be reformulated as:

$$\mathbf{\Phi}^T(X)\,[Y - \mathbf{\Phi}(X)\,\beta] = 0 \tag{6}$$

which can be rearranged to solve for β, resulting in the *linear least-squares regression* method, mathematically expressed as follows:

$$\beta_* = \left[\mathbf{\Phi}^T(X)\,\mathbf{\Phi}(X)\right]^{-1} \mathbf{\Phi}^T(X)\,Y \tag{7}$$

An important feature of this method is that, computationally, the inversion of $\mathbf{\Phi}^T\mathbf{\Phi}$ is typically the most expensive operation, as it is a complex procedure which depends on the size of the matrix. In this case, the matrix size is $m \times m$, where m is the number of functions in the chosen model basis. It should be noted that actually calculating the inverse of this matrix is unnecessary, and thus not recommended, due to the existence of factorization algorithms. However, the computational time required for factorization is still $\sim \mathcal{O}\big(m^3\big)$. This is not normally a problem in most least-squares regression applications, as the number of basis functions is typically < 10.

1.3 Weighted Linear Least-Squares (WLL) Regression

It is also common to use a *weight* matrix, denoted by $\mathbf{\Sigma}$, in order to provide additional input regarding data quality, noise levels (as $1/\sigma_n^2$), or any other prior information concerning the system. This matrix, $\mathbf{\Sigma}$, is a $n \times n$ matrix with non-zero entries only on the main diagonal. This information can be incorporated into the regression method by using the *weighted sum-squared (WSS) error* instead of Equation (2), expressed as follows:

$$M_{\text{WSS}} = \sum_{i}^{n} \Sigma_{ii} \varepsilon_i^2 \tag{8}$$

By replacing M_{SS} with M_{WSS} in Equation (4) and reperforming the algebra using Equation (5), the *weighted linear least-squares regression* method can be found, mathematically expressed as follows:

$$\beta_* = \left[\mathbf{\Phi}^T(X)\,\mathbf{\Sigma}\,\mathbf{\Phi}(X)\right]^{-1}\mathbf{\Phi}^T(X)\,\mathbf{\Sigma}\,Y \tag{9}$$

Similarly to Equation (7), the factorization of $\mathbf{\Phi}^T\mathbf{\Sigma}\mathbf{\Phi}$ is the most computationally expensive component, also scaling as $\sim \mathcal{O}\big(m^3\big)$.

1.4 Non-Linear Least-Squares (NLL) Regression

However, if $\mathbf{\Phi}$ in Equation (4) is not linear in β, then β_* cannot be explicitly computed as each element of $\partial\varepsilon_i/\partial\beta$ will still depend on β itself. In these cases, it becomes necessary to solve the system iteratively by starting with an initial guess, β_0, and updating it in increments, $\Delta\beta_k$. First, the model basis must be linearized around β_k using a *Taylor expansion*, resulting in:

$$\mathbf{\Phi}(X,\beta) \approx \mathbf{\Phi}(X,\beta_k) + \sum_{j}^{m} \frac{\partial\mathbf{\Phi}(X,\beta_k)}{\partial\beta_{j,k}}\left(\beta_j - \beta_{j,k}\right) \tag{10}$$

Then, by substituting Equation (10) into Equation (1), the residual of each data point, i, can be expressed as:

$$\begin{aligned} \varepsilon_i &= Y_i - \boldsymbol{\Phi}(X_i, \beta_k) + \sum_j^m \frac{\partial \boldsymbol{\Phi}(X_i, \beta_k)}{\partial \beta_{j,k}} \left(\beta_j - \beta_{j,k}\right) \\ &= \Delta Y_i + \sum_j^m J_{i,j}\, \Delta\beta_j \end{aligned} \tag{11}$$

where $J_{i,j}$ represents the $(i,j)^{\text{th}}$ element of the *Jacobian* matrix, an $n \times m$ matrix of first-order partial derivatives of the model basis with respect to the parameters, β, evaluated as the data points, X.

Next, by substituting Equation (11) into Equation (4) and reorganizing the terms back into matrix form, the *non-linear least-squares regression* method is found, expressed as:

$$\Delta\beta_k = \left[\mathbf{J}^T(X, \beta_k)\ \mathbf{J}(X, \beta_k)\right]^{-1} \mathbf{J}^T(X, \beta_k)\ \Delta Y_k \tag{12}$$

for which all the quantities must be recalculated at each iteration, k, until a solution is converged upon.

It is important to note that the inclusion of non-linearity has a computational price, both in the computation of $\mathbf{J}$, $\sim \mathcal{O}\left(mn\right)$, and the factorization of $\mathbf{J}^T\mathbf{J}$, $\sim \mathcal{O}\left(m^3\right)$, per iteration.

1.5 Weighted Non-Linear Least-Squares (WNLL) Regression

Similar to the process discussed in Section 1.3, a weight matrix, $\boldsymbol{\Sigma}$, can be added to the residuals to account for any prior information concerning the system. This can be done in a very similar procedure as discussed in Section 1.3, resulting in the *weighted non-linear least-squares regression* method, expressed as follows:

$$\Delta\beta_k = \left[\mathbf{J}^T(X, \beta_k)\, \boldsymbol{\Sigma}\, \mathbf{J}(X, \beta_k)\right]^{-1} \mathbf{J}^T(X, \beta_k)\, \boldsymbol{\Sigma}\, \Delta Y_k \tag{13}$$

Again, similarly to Equation (12), the computation of $\mathbf{J}$ scales as $\sim \mathcal{O}\left(mn\right)$ and the factorization of $\mathbf{J}^T\mathbf{J}$, $\sim \mathcal{O}\left(m^3\right)$, which must both be performed per iteration.

1.6 Disadvantages of Least-Squares Methods

Although these solutions have been used extensively for many scientific and engineering applications, they suffer from a lack of generality which limits their use to describe complex systems. Firstly, as the computational time required for the algorithm scales as $\sim \mathcal{O}\big(m^3\big)$, where m is the number of functions in the model basis, it becomes necessary to pre-select a smaller number of functions to make the model, typically using some physical understanding of the system.

However, this inherently restricts the possible models that can be found by the algorithm, which can be undesired when attempting to model systems with a high degree of complexity.

As an additional restriction, in order to use the LL or WLL regression methods, the model basis must be linear in the parameters, β. While this is not difficult to do, it means that any non-linear behaviour cannot be modelled unless a large number of basis functions are chosen. Alternatively, the NLL and WNLL methods could also be used, but the computational price and the numerical instabilities of the iterative process usually makes them prohibitive for production purposes.

2 Basic Probability Theory

The second topic, probability theory, describes the set of fundamental axioms and statistical tools required to characterize the random behaviour of a system. Then, given a continuous random variable, z, the *probability* that it has a specified value, c, can be denoted as $p(z = c)$. If there is no particular interest in a specified value, the probability is usually expressed as a *probability density function (PDF)*, $p(z)$.

As probability theory is meant to describe real phenomena, it is useful to provide a translation from the mathematical construction to logical concepts. The most important concept in this discussion is that of conditionality, expressed as the probability of a variable given the quantities of other variables. Within the framework of probabilities, this can be expressed as such:

$$p(z_a|z_b) = \frac{p(z_a \cap z_b)}{p(z_b)} \equiv \frac{p(z_a, z_b)}{p(z_b)} \tag{14}$$

where $\cap$ is the *set intersection* operator, or the logical "and" operator. There also exists the *set union* operator, or the logical "or" operation, expressed as such:

$$p(z_a \cup z_b) = p(z_a) + p(z_b) - p(z_a \cap z_b) \tag{15}$$

It should be noted that if z_a and z_b are *independent* variables, or equivalently interpreted as mutually exclusive sets, then the following relations hold true:

$$\begin{gathered} p(z_a \cap z_b) = p(z_a)\,p(z_b) \\ p(z_a \cup z_b) = p(z_a) + p(z_b) \end{gathered} \tag{16}$$

which are useful properties in the derivation of the GPR equations.

Additionally, when working with a vector variable, Z, with N elements, it may be necessary to express the probability of the entire vector, called the *joint probability*, denoted as follows:

$$p(Z) = p(z_1, z_2, ..., z_N) \tag{17}$$

If all N elements are mutually independent, then it is possible to simplify Equation (17) using Equation (16), resulting in the following form:

$$p(Z) = \prod_{i}^{N} p(z_i) \tag{18}$$

2.1 Bayesian Approach to Probability

From the concept of conditional probability comes the field of *Bayesian probability theory*, which was an attempt made by Thomas Bayes to mathematically model the formation and evolution of human belief systems. The result of this attempt is known as *Bayes' theorem*, which is a rearrangement of the axiom provided in Equation (14):

$$p(A|B) = \frac{p(B|A)\, p(A)}{p(B)} \tag{19}$$

where A represents a *hypothesis*, or any form of testable belief, and B represents any form of *evidence* provided. Then, Equation (19) can be seen as a platform on which the trust in a certain hypothesis, $p(A)$, also known as the *prior*, can be updated in light of new information, $p(B|A)$, also known as the *likelihood*, resulting in a new level of trust, $p(A|B)$, also known as the *posterior*. The denominator, $p(B)$, called the *marginal likelihood*, effectively represents the chance that the provided evidence would exist regardless of which hypothesis is correct, which is mathematically modelled as follows:

$$p(B) = \sum_{A} p(B|A) \quad \text{or} \quad p(B) = \int_{-\infty}^{\infty} p(B|A)\, p(A)\, \mathrm{d}A \tag{20}$$

where the summative form is used for discrete variables and the integral form is used for continuous ones. This term is called the marginal likelihood as it removes the dependence of the likelihood on the hypothesis, A, or in other words, it *marginalizes* over A. It should be noted that the integral form of Equation (20) is usually written without the integration limits, but they were included here to enforce the fact that it is not an indefinite integral.

As most applications of GPR work on continuous random variables, the remaining discussion will focus solely on this notation.

2.2 Gaussian (Normal) Probability Distributions

As one might have expected from the name, the GPR method relies heavily on the properties of the *Gaussian*, or *normal*, PDF. The normalized form of this function, for a random variable, z, is given as follows:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right) \sim \mathcal{N}\big(\mu, \sigma^2\big) \tag{21}$$

where μ is the *first moment*, or the *mean*, of the distribution, and σ^2 is the *second moment*, or the *variance*, of the distribution. If a variable behaves according to Equation (21), it is commonly said that it is *Gaussian-distributed* and usually given the short-hand notation expressed in the second form of Equation (21).

If there exists N Gaussian-distributed variables grouped together into a vector, Z, then their joint distribution can be expressed as follows:

$$\begin{aligned} p(Z) &\sim \mathcal{N}\left(M, \Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{1,2}\sigma_1\sigma_2 & \dots & \rho_{1,N}\sigma_1\sigma_N \\ \rho_{2,1}\sigma_2\sigma_1 & \sigma_2^2 & & \\ \vdots & & \ddots & \\ \rho_{N,1}\sigma_N\sigma_1 & & & \sigma_N^2 \end{bmatrix} \right) \\ &= \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2} (Z - M)^T \Sigma^{-1} (Z - M) \right) \end{aligned} \tag{22}$$

where M is a $N \times 1$ vector with elements, μ_i, Σ is a $N \times N$ diagonal matrix with elements, σ_i^2, $|...|$ represents the *determinant* of the enclosed matrix, and $\rho_{i,j}$ represents the correlation factor between variable numbers i and j within the vector, Z. It should be noted that $\rho_{i,i} \equiv 1$ and, due to the symmetry of the Gaussian distribution, $\rho_{i,j} = \rho_{j,i}$. However, if all of the variables in Z are independent of each other, then Equation (22) can be significantly simplified by using Equation (18), as $\rho_{i,j} = 0$ for all $i \neq j$. This can be done as follows:

$$\begin{aligned} p(Z) &= \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2} (Z - M)^T \Sigma^{-1} (Z - M) \right) \\ &= \prod_i^N \frac{1}{\sqrt{2\pi\Sigma_{i,i}}} \exp\left(-\frac{(Z_i - M_i)^2}{2\Sigma_{i,i}} \right) \\ &= \frac{1}{\sqrt{\prod_i^N 2\pi\sigma_i^2}} \exp\left(-\sum_i^N \frac{(z_i - \mu_i)^2}{2\sigma_i^2} \right) \end{aligned} \tag{23}$$

It should be noted that since Σ is a diagonal matrix in the mutually independent scenario, its inverse is also a diagonal matrix but with elements, σ_i^{-2}. If a vector of variables behaves according to Equation (23), then it is usually given the short-hand notation, $p(Z) \sim \mathcal{N}(M, \Sigma)$.

3 Gaussian Process Regression

With the concepts outlined in Sections 1 and 2, it is now possible to reconstruct the GPR methodology. As in a typical regression problem, it starts with a set of input data points, (X, Y), a model basis, $\mathbf{\Phi}(X, \beta)$, and a set of free parameters in the model, β, as introduced in Section 1. Then, by treating the input and output variables as probability distributions instead of as deterministic quantities, Bayes' Theorem, as given in Equation (19), can be applied to this problem as follows:

$$p(\beta|X,Y) = \frac{p(Y|X,\beta)\,p(\beta)}{p(Y|X)} \tag{24}$$

which states that the probability a model describes the input data, $p(\beta|X,Y)$, can be estimated by knowing the likelihood that this data was generated by a process described by the model, $p(Y|X,\beta)$, and the probability of that model itself, $p(\beta)$. Another way to look at this is noting that the posterior incorporates information about the input data, meaning that the prior probability distributions of the free parameters have been constrained based on the provided evidence or input data. Within the machine learning community, it is common to say that the model has been *trained* using the input data sets.

3.1 Assumptions

From here, the first assumption made in this derivation is that the model basis is linear in β. With this assumption, the likelihood can be found by calculating the probability of Y, as described by Equation (5), which requires more assumptions in the probability distributions of the variables, X, β, and ε. Due to the decoupling of β from $\mathbf{\Phi}$, the following shorthand is used from this point forward to improve the readability of this document:

$$\mathbf{\Phi} \equiv \mathbf{\Phi}(X) \tag{25}$$

Note that no distribution is attributed to $\mathbf{\Phi}$ itself, meaning that the model basis must still be pre-selected and fixed for any given application. This turns out to not be as significant of a restriction as for the least-squares regression method, as will be shown later.

Firstly, the joint probability distribution of β is assumed to be a collection of independent Gaussian-distributed random variables with zero mean, as characterized by:

$$p(\beta) \sim \mathcal{N}(0,\Sigma_\beta) = \frac{1}{\sqrt{|2\pi\Sigma_\beta|}} \exp\left(-\frac{1}{2}\beta^T\Sigma_\beta^{-1}\beta\right) \tag{26}$$

It should be noted that this zero-mean Gaussian-distributed assumption implies no loss of generality, as β simply represents a collection of free parameters. As these are, by definition, arbitrarily chosen to satisfy some condition on a goodness-of-fit metric and no restriction is applied to Σ_β, this assumption does not restrict the solution space. This does, however, provide an inherent but weak form of *regularization*, as solutions with large β values will automatically be considered less likely than solutions with more zeros.

Secondly, the joint probability distribution of ε, or the output noise, is also assumed to be a collection of independent Gaussian-distributed random variables with zero mean, as characterized by:

$$p(\varepsilon) \sim \mathcal{N}(0,\Sigma_n) = \frac{1}{\sqrt{|2\pi\Sigma_n|}} \exp\left(-\frac{1}{2}\varepsilon^T\Sigma_n^{-1}\varepsilon\right) \tag{27}$$

By making this assumption, it is implied that the distribution of the noise in Y is Gaussian. As opposed to the assumption made on β, this assumption imposes a strong restriction concerning the noise sources of the input data, which may not always be true. However, in practice, if sufficient care is taken to transform the variables such that the noise has Gaussian-like statistics and remove the outliers without significantly altering these statistics, the results from this process still yields useful information. For improved rigour and robustness, other processes exist which build on the GPR framework and can account for non-Gaussian noise though they typically are much more intense computationally.

Lastly, the joint probability distribution of X, or the *input noise*, is assumed to be a *Dirac delta function*, given as follows:

$$p(X) = \delta(X = X) \tag{28}$$

which essentially means that X is treated as a deterministic variable, with zero probability for X to have any value other than its given value. This is also a strong restriction as it does not allow this process to inherently account for potential errors in the independent or control variable. An improved methodology has been developed for handling input noise, involving the propagation of the input noise through the model, via gradient quantities, and treating them as modifiers to the output noise. This improved GPR is called *noisy-input Gaussian process regression (NIGPR)* and will be expanded on later.

3.2 Derivation

This section will feature a number of detailed mathematical concepts, due to the combination of matrix algebra and multivariate Gaussian integrals. The results are summarized in Section 3.3.

3.2.1 The Posterior Distribution

By combining Equation (5) and the assumption outlined in Equation (27), the likelihood can be calculated as such:

$$p(Y|X,\beta) = \frac{1}{\sqrt{|2\pi\Sigma_n|}} \exp\left(-\frac{1}{2}\left(Y - \mathbf{\Phi}\beta\right)^T \Sigma_n^{-1} \left(Y - \mathbf{\Phi}\beta\right)\right) \tag{29}$$

where all variables except ε are treated as deterministic. Then, by substituting Equations (26) and (29) into the numerator of Equation (19), the posterior can be expressed as:

$$\begin{aligned} p(\beta|X,Y) &= \frac{\exp\left(-\frac{1}{2}\left(Y - \mathbf{\Phi}\beta\right)^T \Sigma_n^{-1} \left(Y - \mathbf{\Phi}\beta\right)\right) \exp\left(-\frac{1}{2}\beta^T \Sigma_\beta^{-1}\beta\right)}{p(Y|X)} \\ &= \frac{\exp\left(-\frac{1}{2}\left[\left(Y - \mathbf{\Phi}\beta\right)^T \Sigma_n^{-1} \left(Y - \mathbf{\Phi}\beta\right) + \beta^T \Sigma_\beta^{-1}\beta\right]\right)}{\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\left[\left(Y - \mathbf{\Phi}\beta\right)^T \Sigma_n^{-1} \left(Y - \mathbf{\Phi}\beta\right) + \beta^T \Sigma_\beta^{-1}\beta\right]\right) \mathrm{d}\beta} \end{aligned} \tag{30}$$

Then, the expression inside the square brackets of Equation (30) can be expanded and simplifed, as follows:

$$\begin{aligned}[...] &= (Y - \mathbf{\Phi}\beta)^T \Sigma_n^{-1} (Y - \mathbf{\Phi}\beta) + \beta^T \Sigma_\beta^{-1} \beta \\ &= Y^T \Sigma_n^{-1} Y - Y^T \Sigma_n^{-1} \mathbf{\Phi}\beta - \beta^T \mathbf{\Phi}^T \Sigma_n^{-1} Y + \beta^T \left(\mathbf{\Phi}^T \Sigma_n^{-1} \mathbf{\Phi} + \Sigma_\beta^{-1} \right) \beta \\ &= Y^T \Sigma_n^{-1} Y - 2\beta^T \mathbf{\Phi}^T \Sigma_n^{-1} Y + \beta^T A \beta\end{aligned} \tag{31}$$

where the following *matrix transpose* relation for a generic matrices, $\mathbf{U}$ and $\mathbf{V}$, is useful:

$$(\mathbf{UV})^T = \mathbf{V}^T \mathbf{U}^T \tag{32}$$

along with the fact that each term in Equation (31) yields a single-element matrix, i.e. $\left(\beta^T \mathbf{\Phi}^T \Sigma_n^{-1} Y\right)^T = \beta^T \mathbf{\Phi}^T \Sigma_n^{-1} Y$. Then, from Equation (31), a term independent of β can be added and subtracted in order to express β in quadratic form, through a process known as *completing the square*. The procedure to accomplish this from a generalized expanded form, $P(z, \mathbf{U}, V)$, where z is the variable to be written in quadratic form, $\mathbf{U}$ is any symmetric positive-definite square matrix and V is any generic vector, is as outlined below:

$$\begin{aligned}P(z, \mathbf{U}, V) &= z^T \mathbf{U} z - 2z^T V \\ &= z^T \mathbf{U} z - 2z^T \mathbf{U}\mathbf{U}^{-1} V + V^T \mathbf{U}^{-1} V - V^T \mathbf{U}^{-1} V \\ &= \left(z - \mathbf{U}^{-1} V\right)^T \mathbf{U} \left(z - \mathbf{U}^{-1} V\right) - V^T \mathbf{U}^{-1} V \\ &= Q(z, \mathbf{U}, V) - V^T \mathbf{U}^{-1} V\end{aligned} \tag{33}$$

where $Q(z, \mathbf{U}, V)$ is simply a short-hand for the quadratic form of z introduced for improved clarity. Note that the following *matrix inverse* relation, for a generic invertible square matrix, $\mathbf{U}$, is useful in the derivation of Equation (33):

$$\mathbf{U}\mathbf{U}^{-1} = \mathbf{U}^{-1}\mathbf{U} = \mathbf{I} \tag{34}$$

where $\mathbf{I}$ is the square identity matrix, which can be multiplied to any appropriate matrix or vector without altering it. Applying the procedure outlined in Equation (33) to Equation (31) yields:

$$[...] = \left(\beta - A^{-1} \mathbf{\Phi}^T \Sigma_n^{-1} Y\right)^T A \left(\beta - A^{-1} \mathbf{\Phi}^T \Sigma_n^{-1} Y\right) + Y^T B Y \tag{35}$$

where

$$\begin{aligned}A &= \mathbf{\Phi}^T \Sigma_n^{-1} \mathbf{\Phi} + \Sigma_\beta^{-1} \\ B &= \Sigma_n^{-1} - \Sigma_n^{-1} \mathbf{\Phi} A^{-1} \mathbf{\Phi}^T \Sigma_n^{-1}\end{aligned} \tag{36}$$

It should be noted that since Σ_n and Σ_β are diagonal matrices, $\Sigma_n^T = \Sigma_n$ and $\Sigma_\beta^T = \Sigma_\beta$.

The integral required to determine the normalization factor can also be calculated quickly, shown generally as an extension of the generalized quadratic completion procedure outline in Equation (33) as follows:

$$\begin{aligned}
I(z,\mathbf{U},V) &= \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2}P(z,\mathbf{U},V)\right] \mathrm{d}z \\
&= \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2}Q(z,\mathbf{U},V) + \frac{1}{2}V^T\mathbf{U}^{-1}V\right] \mathrm{d}z \\
&= \exp\left(\frac{1}{2}V^T\mathbf{U}^{-1}V\right) \frac{\sqrt{|2\pi\mathbf{U}^{-1}|}}{\sqrt{|2\pi\mathbf{U}^{-1}|}} \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2}Q(z,\mathbf{U},V)\right] \mathrm{d}z \\
&= \sqrt{|2\pi\mathbf{U}^{-1}|}\exp\left(\frac{1}{2}V^T\mathbf{U}^{-1}V\right)
\end{aligned} \tag{37}$$

where the integrand in the second last line, when combined with the square root term in the denominator, forms a normalized Gaussian distribution in z, for which the integral with respect to z from $-\infty$ to ∞ is unity.

Now, Equation (35) can be substituted back into Equation (30) and applying the procedure outlined in Equation (37), the following result is obtained:

$$\begin{aligned}
p(\beta|X,Y) &= \frac{1}{\sqrt{|2\pi A^{-1}|}} \exp\left(-\frac{1}{2}\left(\beta - \mathbf{\Gamma}Y\right)^T A \left(\beta - \mathbf{\Gamma}Y\right)\right) \\
&\sim \mathcal{N}\left(\mathbf{\Gamma}Y, A^{-1}\right)
\end{aligned} \tag{38}$$

where A is given by Equation (36) and

$$\mathbf{\Gamma} = A^{-1}\mathbf{\Phi}^T\Sigma_n^{-1} \tag{39}$$

which is introduced simply for improved readability. It should be noted that $\mathbf{\Gamma}$ is not generally a square matrix, as $\mathbf{\Phi}$ is generally not a square matrix.

It should be noted that the GPR algorithm as given in Equation (38) yields no actual advantage over the WLLS regression, described in Section 1.3, as the computation still requires an explicitly defined matrix representing the model basis, $\mathbf{\Phi}$, and by extension, the parameter vector, β, as well. This implies that there is no computational advantage gained for a large basis function size, m.

3.2.2 The Predictive Distribution

However, for most applications, knowledge about the posterior probability distributions of the free parameters, β, are largely unnecessary. It is by far more interesting to know the predictions of the model, Y_*, at specified inputs, X_*, resulting from these trained free parameter combinations. Following this premise, the probability distributions of the predictive model can be determined as follows:

$$p(Y_*|X_*,X,Y) = \int_{-\infty}^{\infty} p(Y_*|X_*,\beta)\, p(\beta|X,Y)\, \mathrm{d}\beta \tag{40}$$

where $p(\beta|X,Y)$ is given by Equation (38). A simple comparison of the components of Equation (40) reveals that it is the likelihood of a set of predictions, given a model, weighted by the posterior probability of the given model determined from the input data and integrated over all possible models. Then, by substituting Equations (29), with X, Y replaced by X_*, Y_*, and (38) into Equation (40), the integrand can be expanded and expressed as:

$$\exp\left(-\frac{1}{2}\left[\left(Y_* - \mathbf{\Phi}_*\beta\right)^T \Sigma_{n*}^{-1}\left(Y_* - \mathbf{\Phi}_*\beta\right) + \left(\beta - \mathbf{\Gamma}Y\right)^T A\left(\beta - \mathbf{\Gamma}Y\right)\right]\right) \tag{41}$$

where A is given by Equation (36), $\mathbf{\Gamma}$ is given by Equation (39) and

$$\mathbf{\Phi}_* \equiv \mathbf{\Phi}(X_*) \tag{42}$$

Due to the quadratic nature of the components of Equation (41) resulting from the Gaussian distributions, the procedure outlined in Equations (33) can be applied to the expression in the square brackets in order to yield:

$$\begin{aligned}
[...] =\ & Y_*^T\Sigma_{n*}^{-1}Y_* - \beta^T\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}Y_* - Y_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_*\beta + \beta^T\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_*\beta \\
& \qquad + \beta^T A\beta - \beta^T A\mathbf{\Gamma}Y - Y^T\mathbf{\Gamma}^T A\beta + Y^T\mathbf{\Gamma}^T A\mathbf{\Gamma}Y \\
=\ & Y_*^T\Sigma_{n*}^{-1}Y_* - 2\beta^T\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}Y_* + \beta^T\left(A + \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_*\right)\beta \\
& \qquad - 2\beta^T\mathbf{\Phi}^T\Sigma_n^{-1}Y + Y^T\Sigma_n^{-1}\mathbf{\Phi}A^{-1}\mathbf{\Phi}^T\Sigma_n^{-1}Y \\
=\ & Y_*^T\Sigma_{n*}^{-1}Y_* + Y^T\Sigma_n^{-1}\mathbf{\Phi}A^{-1}\mathbf{\Phi}^T\Sigma_n^{-1}Y \\
& \qquad + \beta^T\left(A + \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_*\right)\beta - 2\beta^T\left(\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}Y_* + \mathbf{\Phi}^T\Sigma_n^{-1}Y\right) \\
=\ & Y_*^T\Sigma_{n*}^{-1}Y_* + Y^T\Sigma_n^{-1}\mathbf{\Phi}A^{-1}\mathbf{\Phi}^T\Sigma_n^{-1}Y + P(\beta, G, Z_* + Z)
\end{aligned} \tag{43}$$

where A is given by Equation (36), $\mathbf{\Gamma}$ is given by Equation (39), both substituted in where necessary, and

$$G = A + \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_*\ , \quad Z_* = \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}Y_*\ , \quad Z = \mathbf{\Phi}^T\Sigma_n^{-1}Y \tag{44}$$

Once Equation (43) is resubstituted into Equation (41) and then into Equation (40), the first two terms of the last expression in Equation (43) can be brought out of the integral as they are independent of β. The resulting integral, labelled as J, can be solved using the procedure outlined in Equation (37), resulting in the following:

$$\begin{aligned}
J &= \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\,P(\beta, G, Z_* + Z)\right) \\
&= \sqrt{|2\pi G^{-1}|}\,\exp\left(\frac{1}{2}\left(Z_* + Z\right)^T G^{-1}\left(Z_* + Z\right)\right)
\end{aligned} \tag{45}$$

where G, Z and Z_* are given by Equation (44).

Then, by recombining the first two terms of the last expression of Equation (43) and Equation (45), the predictive distribution can be written as follows:

$$p(Y_*|X_*, X, Y) = \sqrt{\frac{|2\pi G^{-1}|}{|2\pi\Sigma_{n*}|\,|2\pi A^{-1}|}}\,\exp\Bigg(-\frac{1}{2}\Bigg[Y_*^T\Sigma_{n*}^{-1}Y_* \\ + Z^T A^{-1} Z - (Z_* + Z)^T\, G^{-1}\,(Z_* + Z)\Bigg]\Bigg) \tag{46}$$

where A is given by Equation (36) and G, Z and Z_* are given by Equation (44). However, it is much more convenient to rewrite Equation (46) as a quadratic in Y_*, such that it can be expressed explicitly as a Gaussian distribution itself. This can be done by expanding the expression in the square brackets of Equation (46) and applying the square completion procedure, outlined in Equation (33), resulting in the following:

$$\begin{aligned}[...] &= \begin{aligned} &Y_*^T\Sigma_{n*}^{-1}Y_* + Z^T A^{-1} Z - Y_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1}\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}Y_* \\ &\qquad\qquad - 2Y_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1} Z - Z^T G^{-1} Z \end{aligned} \\ &= P\big(Y_*, \Xi, \Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1} Z\big) + Z^T\left(A^{-1} - G^{-1}\right) Z \\ &= \begin{aligned} &Q\big(Y_*, \Xi, \Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1} Z\big) \\ &\quad + Z^T\big(A^{-1} - G^{-1} - G^{-1}\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\Xi\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1}\big) Z \end{aligned}\end{aligned} \tag{47}$$

where A is given by Equation (36), G and Z are given by Equation (44) and

$$\Xi = \Sigma_{n*}^{-1} - \Sigma_{n*}^{-1}\mathbf{\Phi}_*\left(A + \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_*\right)^{-1}\mathbf{\Phi}_*^T\Sigma_{n*}^{-1} \tag{48}$$

Finally, by substituting the result of Equation (47) into Equation (46), the predictive distribution can be expressed as:

$$\begin{aligned} p(Y_*|X_*, X, Y) &= D\,\exp\Big(-\frac{1}{2}\,Q\big(Y_*, \Xi, \Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1} Z\big)\Big) \\ &\sim \mathcal{N}\big(\Xi^{-1}\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1} Z, \Xi^{-1}\big) \end{aligned} \tag{49}$$

where G and Z are given by Equation (44), Ξ is given by Equation (48) and

$$D = \sqrt{\frac{|2\pi G^{-1}|}{|2\pi\Sigma_{n*}|\,|2\pi A^{-1}|}} \\ \times \exp\Big(-\frac{1}{2}Z^T\big(A^{-1} - G^{-1} - G^{-1}\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\Xi\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1}\big) Z\Big) \tag{50}$$

where A is given by Equation (36).

However, in the form given by Equation (49), there is still no computational advantage between the GPR and the WLLS regression technique, due to the explicit presence of the model basis, $\mathbf{\Phi}$. In an attempt to circumvent this restriction, the mean and variance parameters from Equation (49) can be expanded and simplified. However, this simplification involves the inversion of $\mathbf{\Phi}$, which is tricky as it is generally a non-square matrix, meaning it is conventionally non-invertible. Thus, a *pseudoinverse* should be used in place of the inverse operation and one way to mathematically perform this inversion is called the *Moore-Penrose pseudoinverse*, defined as follows:

$$\mathbf{\Phi}^{-1} \quad \longrightarrow \quad \mathbf{\Phi}^{+} \equiv \begin{cases} \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T & \Longrightarrow \quad \mathbf{\Phi}^{+}\mathbf{\Phi} = \mathbf{I} \\ & \text{or} \\ \mathbf{\Phi}^T\left(\mathbf{\Phi}\mathbf{\Phi}^T\right)^{-1} & \Longrightarrow \quad \mathbf{\Phi}\mathbf{\Phi}^{+} = \mathbf{I} \end{cases} \tag{51}$$

where $\mathbf{I}$ is the identity matrix. This operation takes advantage of the fact that any matrix multiplied by its transpose yields a square invertible matrix. It is assumed that the form of this pseudoinverse taken in the following equations is such that it reduces back to the identity matrix when reversed.

With Equation (51), it is now possible to simplify the inversion of Ξ, given by Equation (48), as follows:

$$\begin{aligned} \Xi^{-1} &= \left(\Sigma_{n*}^{-1} - \Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1}\mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\right) \\ &= \left\{\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1}\left[G\mathbf{\Phi}_*^{+} - \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\right]\right\}^{-1} \\ &= \left\{\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1}\left[\left(A + \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\mathbf{\Phi}_*\right)\mathbf{\Phi}_*^{+} - \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\right]\right\}^{-1} \\ &= \left(A\mathbf{\Phi}_*^{+} + \mathbf{\Phi}_*^T\Sigma_{n*}^{-1} - \mathbf{\Phi}_*^T\Sigma_{n*}^{-1}\right)^{-1} G\mathbf{\Phi}_*^{+}\Sigma_{n*} \\ &= \mathbf{\Phi}_* A^{-1} G\mathbf{\Phi}_*^{+}\Sigma_{n*} \end{aligned} \tag{52}$$

where A is given by Equation (36) and G is given by Equation (44). Then, by substituting Equation (52) into Equation (49), the mean value of the predictive distribution, denoted as $\mathbb{E}[Y_*]$, can be simplified as follows:

$$\begin{aligned} \mathbb{E}[Y_*] &= \Xi^{-1}\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1} Z \\ &= \mathbf{\Phi}_* A^{-1} G\mathbf{\Phi}_*^{+}\Sigma_{n*}\Sigma_{n*}^{-1}\mathbf{\Phi}_* G^{-1} Z \\ &= \mathbf{\Phi}_*\left(\mathbf{\Phi}^T\Sigma_n^{-1}\mathbf{\Phi} + \Sigma_\beta^{-1}\right)^{-1}\mathbf{\Phi}^T\Sigma_n^{-1} Y \\ &= \mathbf{\Phi}_*\left[\mathbf{\Phi}^T\Sigma_n^{-1}\left(\mathbf{\Phi}\Sigma_\beta\mathbf{\Phi}^T + \Sigma_n\right)\left(\Sigma_\beta\mathbf{\Phi}^T\right)^{-1}\right]^{-1}\mathbf{\Phi}^T\Sigma_n^{-1} Y \\ &= \mathbf{\Phi}_*\Sigma_\beta\mathbf{\Phi}^T\left(\mathbf{\Phi}\Sigma_\beta\mathbf{\Phi}^T + \Sigma_n\right)^{-1}\left(\mathbf{\Phi}^T\Sigma_n^{-1}\right)^{-1}\mathbf{\Phi}^T\Sigma_n^{-1} Y \\ &= \mathbf{\Phi}_*\Sigma_\beta\mathbf{\Phi}^T\left(\mathbf{\Phi}\Sigma_\beta\mathbf{\Phi}^T + \Sigma_n\right)^{-1} Y \end{aligned} \tag{53}$$

where A is given by Equation (36) and G and Z is given by Equation (44), all substituted in where necessary. Similarly, the variance of the predictive distribution, denoted as $\mathbb{V}[Y_*] = \Xi^{-1}$, can be expanded and simplified by using the *Woodbury matrix identity*, given as:

$$\left(V_1\mathbf{U}_1 V_2 + \mathbf{U}_2\right)^{-1} = \mathbf{U}_2^{-1} - \mathbf{U}_2^{-1} V_1\left(V_2\mathbf{U}_2^{-1} V_1 + \mathbf{U}_1^{-1}\right)^{-1} V_2\mathbf{U}_2^{-1} \tag{54}$$

which holds true for any invertible square matrices, $\mathbf{U}_1$ and $\mathbf{U}_2$, and any appropriately shaped matrices, V_1 and V_2, which satisfy the rules of matrix algebra for this equation. Then, by applying Equation (54) on A^{-1} inside Equation (52), the following result is obtained:

$$\begin{aligned}
\mathbb{V}[Y_*] &= \boldsymbol{\Phi}_* A^{-1} \left(A + \boldsymbol{\Phi}_*^T \Sigma_{n*}^{-1} \boldsymbol{\Phi}_* \right) \boldsymbol{\Phi}_*^+ \Sigma_{n*} \\
&= \boldsymbol{\Phi}_* A^{-1} A \boldsymbol{\Phi}_*^+ \Sigma_{n*} + \boldsymbol{\Phi}_* A^{-1} \boldsymbol{\Phi}_*^T \Sigma_{n*}^{-1} \boldsymbol{\Phi}_* \boldsymbol{\Phi}_*^+ \Sigma_{n*} \\
&= \Sigma_{n*} + \boldsymbol{\Phi}_* \left(\boldsymbol{\Phi}^T \Sigma_n^{-1} \boldsymbol{\Phi} + \Sigma_\beta^{-1} \right)^{-1} \boldsymbol{\Phi}_*^T \\
&= \Sigma_{n*} + \boldsymbol{\Phi}_* \left[\Sigma_\beta - \Sigma_\beta \boldsymbol{\Phi}^T \left(\boldsymbol{\Phi} \Sigma_\beta \boldsymbol{\Phi}^T + \Sigma_n \right)^{-1} \boldsymbol{\Phi} \Sigma_\beta \right] \boldsymbol{\Phi}_*^T \\
&= \boldsymbol{\Phi}_* \Sigma_\beta \boldsymbol{\Phi}_*^T + \Sigma_{n*} - \boldsymbol{\Phi}_* \Sigma_\beta \boldsymbol{\Phi}^T \left(\boldsymbol{\Phi} \Sigma_\beta \boldsymbol{\Phi}^T + \Sigma_n \right)^{-1} \boldsymbol{\Phi} \Sigma_\beta \boldsymbol{\Phi}_*^T
\end{aligned} \tag{55}$$

where A is given by Equation (36) and is substituted in where necessary.

3.2.3 The Kernel Trick

In order for this method to provide an improvement over the least-squares methods, the explicit reference to the model basis, $\boldsymbol{\Phi}$, should be removed from Equations (53) and (55). Since the predictive distribution, or the probability distributions of the trained model, is expressed in terms of its mean and variance, it is intuitive to also express the untrained model, denoted with $\widetilde{Y}$, in this format as well in order to gain some insight into the role of the model basis within this Bayesian framework.

Firstly, it is useful to express the probability distribution of the untrained model, as such:

$$p\left(\widetilde{Y}\right) = p(\beta \cap \varepsilon) = p(\beta)\, p(\varepsilon) \tag{56}$$

where the last form comes from using Equation (16), taking advantage of the assumption that β and ε are independent random variables. Then, by using Equation (5) with Y replaced with $\widetilde{Y}$, with the assumed variable distributions given by Equations (26) and (28), the mean of the untrained model can be expressed as follows:

$$\begin{aligned}
\mathbb{E}\left[\widetilde{Y}\right] &= \int_{\widetilde{Y}} \widetilde{Y} p\left(\widetilde{Y}\right) \mathrm{d}\widetilde{Y} \\
&= \int_\beta \int_\varepsilon \left(\boldsymbol{\Phi}\beta + \varepsilon \right)\, p(\beta)\, p(\varepsilon)\, \mathrm{d}\varepsilon\, \mathrm{d}\beta \\
&= \int_\beta \boldsymbol{\Phi}\beta\, p(\beta)\, \mathrm{d}\beta \int_\varepsilon p(\varepsilon)\, \mathrm{d}\varepsilon + \int_\varepsilon \varepsilon\, p(\varepsilon)\, \mathrm{d}\varepsilon \int_\beta p(\beta)\, \mathrm{d}\beta \\
&= \boldsymbol{\Phi} \int_\beta \beta\, p(\beta)\, \mathrm{d}\beta + \int_\varepsilon \varepsilon\, p(\varepsilon)\, \mathrm{d}\varepsilon \\
&= \boldsymbol{\Phi}\, \mathbb{E}[\beta] + \mathbb{E}[\varepsilon] \\
&= 0
\end{aligned} \tag{57}$$

where the integration of the probability distributions on their own is equal to unity, due to the normalization constant. The result of Equation (57) is not surprising given the assumptions taken, but it has implications that will be discussed later. Similarly, the variance of the untrained model can be written as follows:

$$
\begin{aligned}
\mathbb{V}\left[\widetilde{Y}\right] &= \int_{\widetilde{Y}} \widetilde{Y}\widetilde{Y}^T p\left(\widetilde{Y}\right) \mathrm{d}\widetilde{Y} \\
&= \int_\beta \int_\varepsilon \left(\boldsymbol{\Phi}\beta + \varepsilon\right)\left(\boldsymbol{\Phi}\beta + \varepsilon\right)^T p(\beta)\, p(\varepsilon)\, \mathrm{d}\varepsilon\, \mathrm{d}\beta \\
&= \int_\beta \int_\varepsilon \left(\boldsymbol{\Phi}\beta\beta^T\boldsymbol{\Phi}^T + \boldsymbol{\Phi}\beta\varepsilon^T + \varepsilon\beta^T\boldsymbol{\Phi}^T + \varepsilon\varepsilon^T\right) p(\beta)\, p(\varepsilon)\, \mathrm{d}\varepsilon\, \mathrm{d}\beta \\
&= \begin{matrix} \boldsymbol{\Phi}\left[\int_\beta \beta\beta^T p(\beta)\,\mathrm{d}\beta\right]\boldsymbol{\Phi}^T + \boldsymbol{\Phi}\left[\int_\beta \beta\, p(\beta)\,\mathrm{d}\beta\right]\left[\int_\varepsilon \varepsilon^T p(\varepsilon)\,\mathrm{d}\varepsilon\right] \\ + \left[\int_\varepsilon \varepsilon\, p(\varepsilon)\,\mathrm{d}\varepsilon\right]\left[\int_\beta \beta^T p(\beta)\,\mathrm{d}\beta\right]\boldsymbol{\Phi}^T + \int_\varepsilon \varepsilon\varepsilon^T p(\varepsilon)\,\mathrm{d}\varepsilon \end{matrix} \\
&= \boldsymbol{\Phi}\,\mathbb{V}[\beta]\,\boldsymbol{\Phi}^T + \boldsymbol{\Phi}\mathbb{E}[\beta]\,\mathbb{E}\left[\varepsilon^T\right] + \mathbb{E}\left[\varepsilon\right]\mathbb{E}\left[\beta^T\right]\boldsymbol{\Phi}^T + \mathbb{V}[\varepsilon] \\
&= \boldsymbol{\Phi}\Sigma_\beta\boldsymbol{\Phi}^T + 0 + 0 + \Sigma_n \\
&= \boldsymbol{\Phi}\Sigma_\beta\boldsymbol{\Phi}^T + \Sigma_n
\end{aligned} \tag{58}
$$

where the integral of only the probability density function is equal to unity, due to the normalization constant, and it is assumed that β and ε are independent random variable.

By examining Equation (58), it becomes clear that selecting the model basis, $\boldsymbol{\Phi}$, is equivalent to specifying the *covariance* of the untrained model, $\widetilde{Y}$, making it conceptually possible to replace one with the other. Coincidentally, by comparing the result of Equation (58) to Equations (53) and (55), it becomes evident that this replacement can be done mathematically as well. Specifically, a new matrix variable, K, known as the *kernel* or *Gram matrix*, is introduced, representing the covariance matrix of the untrained model, and it is defined as such:

$$
K(X_1, X_2) = \boldsymbol{\Phi}\Sigma_\beta\boldsymbol{\Phi}^T \equiv \boldsymbol{\Phi}(X_1)\,\Sigma_\beta\,\boldsymbol{\Phi}^T(X_2) \tag{59}
$$

where the subscripts 1 and 2 are used merely to emphasize the fact that they can have distinct numerical values but must represent the same conceptual variable. Substituting Equation (59) into Equations (53) and (55) yields the following simplified forms for the mean and variance of the predictive distribution:

$$
\begin{aligned}
\mathbb{E}[Y_*] &= K(X_*, X)\left[K + \Sigma_n\right]^{-1} Y \\
\mathbb{V}[Y_*] &= K_* + \Sigma_{n*} - K(X_*, X)\left[K + \Sigma_n\right]^{-1} K(X, X_*)
\end{aligned} \tag{60}
$$

where the short-hand, $K = K(X, X)$ and $K_* = K(X_*, X_*)$, was used for improved readability. From here, the computational advantage of the GPR technique becomes apparent, as the most complex calculation involves the factorization, a process which facilitates the matrix inversion operation, of the kernel, K, which is an $n \times n$ matrix, where n is the number of input data points, (X, Y).

This means that the GPR method can essentially use an infinite set of basis functions to fit the data without incurring any significant penalty in computational speed. However, the disadvantage is that the fitted function cannot be expressed in an analytical form, excluding the resulting fits from being subjected to certain types of additional analysis.

One such kernel derived from an infinite set of basis functions is called the *square exponential (SE) kernel*. Specifically, the basis functions are an infinite set of unnormalized Gaussians, with each centered on a different point in the space but all with an identical "width", denoted with σ_g, expressed as:

$$\mathbf{\Phi} = \{y_i(x)\} \qquad \text{where,} \qquad y_i(x) = \exp\left(-\frac{\left(x-\mu_i\right)^2}{2\sigma_g^2}\right) \tag{61}$$

where μ_i represents the center position of the given Gaussian basis function, each given a unique subscript i. Then, from Equation (59), the kernel element can be evaluated for any pair of input data points as follows:

$$\begin{aligned} K(X_1, X_2) &= \mathbf{\Phi}\Sigma_\beta\mathbf{\Phi}^T \\ &= \lim_{N\to\infty}\sum_i^N y_i(X_1)\,\sigma_\beta^2\,y_i(X_2) \\ &= \sigma_\beta^2\int_{-\infty}^{\infty}\exp\left(-\frac{\left(X_1-\mu\right)^2}{2\sigma_g^2}\right)\exp\left(-\frac{\left(X_2-\mu\right)^2}{2\sigma_g^2}\right)\mathrm{d}\mu \\ &= \sigma_\beta^2\int_{-\infty}^{\infty}\exp\left(-\frac{2\mu^2-2\mu\left(X_1+X_2\right)+X_1^2+X_2^2}{2\sigma_g^2}\right)\mathrm{d}\mu \\ &= \sigma_\beta^2\int_{-\infty}^{\infty}\exp\left(-\frac{\left(\mu-\frac{1}{2}\left(X_1+X_2\right)\right)^2}{2\left(\sigma_g/\sqrt{2}\right)^2}-\frac{\left(X_1-X_2\right)^2}{2\left(\sqrt{2}\sigma_g\right)^2}\right)\mathrm{d}\mu \\ &= \sigma_\beta^2\sqrt{\pi\sigma_g^2}\,\exp\left(-\frac{\left(X_1-X_2\right)^2}{2\left(\sqrt{2}\sigma_g\right)^2}\right) \\ &= \sigma\exp\left(-\frac{\left(X_1-X_2\right)^2}{2l^2}\right) \end{aligned} \tag{62}$$

where procedures similar to those outlined in Equations (33) and (37) were applied to arrive at the final expression.

In order to actually calculate the predictive distributions given in Equation (60) for performing regressions, two things are still required: a selection for the kernel, K, and a definition for the variance of the predicted noise, Σ_{n*}. The first requirement, ie. the kernel, can be arbitrarily chosen but must satisfy the conditions of being *symmetric* and at least *positive semi-definite*, due to the fact that it represents the covariance of the untrained model. The second requirement, ie. the variance of the predicted noise, is unfortunately not so flexible, as it must be done consistently with the variance of the output noise, Σ_n, in order for the predicted variance to have any meaning. Since the size of Σ_{n*} does not generally equal the size of Σ_n, it is not mathematically sufficient to simply make the two equivalent. In the most basic application of GPR, it is assumed that the variance of the output noise, σ_n^2, is the same for each individual input data point, allowing for the selection:

$$\sigma_{n*}^2 = \sigma_n^2 \tag{63}$$

This approach is called the *homoscedastic* GPR, as all the random variables are treated as having the same finite variance. It should be noted that this choice does not normally allow the regression technique to account for individual measurement errors, as it replaces this information with a constant noise value. One could pre-treat the measurement error data in order to determine a single constant that captures these errors, but methods of doing this in a self-consistent and statistically rigourous manner will not be discussed in this document.

It should be noted that the GPR technique, as given in Equation (60) has been fully derived and can be used in combination with input data, (X, Y), a noise variance estimation, σ_n^2, and a kernel, $K(X_1, X_2)$, to produce a model and make predictions from that model. From this point forward, the discussion will switch to methods of improving this methodology.

3.2.4 The Kernel Function and Application of Derivatives

From Equation (58), the elements of the kernel matrix, K, can be seen as the covariance of the noise-free untrained model between any two given points in the independent variable space, X. Thus, in order to facilitate the interpretation of the results, the kernel matrix is typically calculated from a *kernel function* or *covariance function*, denoted with $k(x_1, x_2, \theta)$, where x_1 and x_2 denote a continuous independent variable space and θ represents a set of *hyperparameters*, conceptually replacing the externally-adjustable free parameters, β. By taking the example shown in Equation (62), the SE kernel function can be expressed as:

$$k(x_1, x_2, \theta) = \sigma \exp\left(-\frac{(x_1 - x_2)^2}{2l^2}\right) \qquad \text{where} \qquad \theta = \{\sigma, l\} \tag{64}$$

where the hyperparameters allow for the fine-tuning of the model without worries of overfitting from the infinite set of basis functions. It should be noted that there are many other kernel functions available to be used, with each one having different regression model characteristics. However, these kernel function options will not be discussed in this document.

Given that the chosen kernel function is at least mixed second-order differentiable, meaning that $\partial k / \partial x_1$, $\partial k / \partial x_2$, and $\partial^2 k / \partial x_1 \partial x_2$ exist, derivative information or constraints can be added into the problem by extending the set of input data points to (X', Y'), with each defined as follows:

$$X' = \begin{bmatrix} X \\ X_d \end{bmatrix} \quad , \qquad Y' = \begin{bmatrix} Y \\ \frac{\partial Y}{\partial X_d} \end{bmatrix} \tag{65}$$

Then, in order to account for the input derivative data in the predictive equations, the appropriate kernel matrix elements must be found. As the kernel effectively represents the covariance between any two input data points, the required kernel matrix elements should also describe the covariance between the input derivative data and any other input data point, including itself.

In order to find the expressions for these covariances, a good starting point is to take the derivative of the untrained model, $\widetilde{Y}$, as follows:

$$\frac{\partial \widetilde{Y}}{\partial X} = \frac{\partial \mathbf{\Phi}}{\partial X}\beta + \frac{\partial \varepsilon}{\partial X} \tag{66}$$

since β is independent of X, as stated in Section 3.1. As it is not generally assumed that the noise is independent of X, its derivative is included in the derivative of the untrained model. Then, by employing a similar procedure to that used in Equation (58), the variance of the derivative of the untrained model can be expressed as follows:

$$\begin{aligned}
\mathbb{V}\left[\frac{\partial \widetilde{Y}}{\partial X}\right] &= \int_{\widetilde{Y}} \frac{\partial \widetilde{Y}}{\partial X_1}\frac{\partial \widetilde{Y}}{\partial X_2}^T p\Big(\widetilde{Y}\Big)\,\mathrm{d}\widetilde{Y} \\
&= \begin{aligned}&\frac{\partial \mathbf{\Phi}(X_1)}{\partial X_1}\,\mathbb{V}[\beta]\frac{\partial \mathbf{\Phi}^T(X_2)}{\partial X_2} + \mathbb{E}\left[\frac{\partial \varepsilon}{\partial X_1}\right]\mathbb{E}\left[\beta^T\right]\frac{\partial \mathbf{\Phi}^T(X_2)}{\partial X_2} \\ &\qquad + \frac{\partial \mathbf{\Phi}(X_1)}{\partial X_1}\mathbb{E}[\beta]\,\mathbb{E}\left[\frac{\partial \varepsilon^T}{\partial X_2}\right] + \mathbb{V}\left[\frac{\partial \varepsilon}{\partial X}\right]\end{aligned} \\
&= \frac{\partial^2}{\partial X_1 \partial X_2}\left(\mathbf{\Phi}\Sigma_\beta\mathbf{\Phi}^T\right) + 0 + 0 + \Sigma_{nd} \\
&= \frac{\partial^2 K(X_1, X_2)}{\partial X_1 \partial X_2} + \Sigma_{nd}
\end{aligned} \tag{67}$$

where the derivatives with respect to X_1 and X_2 can be applied to the entire expression as only $\mathbf{\Phi}$ and $\mathbf{\Phi}^T$, respectively, are dependent on those variables in that expression. The covariance between the model and its derivative can also be found in a similar fashion, as follows:

$$\begin{aligned}
\mathbb{C}\left[\widetilde{Y}, \frac{\partial \widetilde{Y}}{\partial X}\right] &= \int_{\widetilde{Y}} \widetilde{Y}\frac{\partial \widetilde{Y}}{\partial X_2}^T p\Big(\widetilde{Y}\Big)\,\mathrm{d}\widetilde{Y} \\
&= \begin{aligned}&\mathbf{\Phi}(X_1)\,\mathbb{V}[\beta]\,\frac{\partial \mathbf{\Phi}^T(X_2)}{\partial X_2} + \mathbb{E}[\varepsilon]\,\mathbb{E}\left[\beta^T\right]\frac{\partial \mathbf{\Phi}^T(X_2)}{\partial X_2} \\ &\qquad + \mathbf{\Phi}(X_1)\,\mathbb{E}(\beta)\,\mathbb{E}\left[\frac{\partial \varepsilon^T}{\partial X_2}\right] + \mathbb{C}\left[\varepsilon, \frac{\partial \varepsilon}{\partial X}\right]\end{aligned} \\
&= \frac{\partial}{\partial X_2}\left(\mathbf{\Phi}\Sigma_\beta\mathbf{\Phi}^T\right) + 0 + 0 + 0 \\
&= \frac{\partial K(X_1, X_2)}{\partial X_2}
\end{aligned} \tag{68}$$

where it is assumed that the covariance between the noise term and its derivative is zero. Through a similar derivation, the covariance with the inputs switched can be expressed as:

$$\mathbb{C}\left[\frac{\partial \widetilde{Y}}{\partial X}, \widetilde{Y}\right] = \frac{\partial K(X_1, X_2)}{\partial X_1} \tag{69}$$

Then, using Equations (67), (68) and (69), the kernel matrix and noise matrix can be updated as follows:

$$K' \equiv K(X_1', X_2') = \begin{bmatrix} K(X_1, X_2) & \frac{\partial K(X_1, X_{d2})}{\partial X_{d2}} \\ \frac{\partial K(X_{d1}, X_2)}{\partial X_{d1}} & \frac{\partial^2 K(X_{d1}, X_{d2})}{\partial X_{d1} \partial X_{d2}} \end{bmatrix} \quad , \quad \Sigma_n' = \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_{nd} \end{bmatrix} \tag{70}$$

where $\Sigma_{nd} = 0$ in the homoscedastic case, as a flat noise function has a derivative of zero and an associated error of zero. By applying Equation (70) to Equation (60), the homoscedastic predictive equations become:

$$\begin{aligned} \mathbb{E}[Y_*] &= K(X_*, X') \left[K' + \Sigma_n'\right]^{-1} Y' \\ \mathbb{V}[Y_*] &= K_* + \Sigma_{n*} - K(X_*, X') \left[K' + \Sigma_n'\right]^{-1} K(X', X_*) \end{aligned} \tag{71}$$

where Equation (63) still applies.

Additionally, provided that the chosen kernel function is at least second-order differentiable, the derivatives of the predictive distribution can also be determined. The mean value of the derivative of the GPR prediction in Equation (71) can be found as follows:

$$\begin{aligned} \mathbb{E}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \int_{Y_*} \frac{\partial Y_*}{\partial X_*} \, p(Y_*) \, \mathrm{d}Y_* \\ &= \frac{\partial}{\partial X_*} \left[\int_{Y_*} Y_* \, p(Y_*) \, \mathrm{d}Y_*\right] \\ &= \frac{\partial \, \mathbb{E}[Y_*]}{\partial X_*} \\ &= \frac{\partial K(X_*, X')}{\partial X_*} \left[K' + \Sigma_n'\right]^{-1} Y' \end{aligned} \tag{72}$$

where the derivative can be taken out of the integral since the probability density function, $p(Y_*)$, depends only on the value of the variable, Y_*, itself. In other words, this operator exchange can be done since the expectation value is a linear operation. Similarly, the variance of the derivative of the GPR prediction in Equation (71) can be found as follows:

$$\begin{aligned}
\mathbb{V}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \int_{Y_*} \frac{\partial Y_*(X_{*1})}{\partial X_{*1}} \frac{\partial Y_*(X_{*2})}{\partial X_{*2}}\, p(Y_*)\, \mathrm{d}Y_* \\
&= \frac{\partial}{\partial X_{*1}} \frac{\partial}{\partial X_{*2}} \left[\int_{Y_*} Y_*\, p(Y_*)\, \mathrm{d}Y_*\right] \\
&= \frac{\partial^2\, \mathbb{V}[Y_*]}{\partial X_{*1} \partial X_{*2}} \\
&= \frac{\partial^2 K_*}{\partial X_{*1} \partial X_{*2}} - \frac{\partial K(X_{*1}, X')}{\partial X_{*1}} \left[K' + \Sigma_n' \right]^{-1} \frac{\partial K(X', X_{*2})}{\partial X_{*2}}
\end{aligned} \tag{73}$$

where $\partial \Sigma_{n*} / \partial X_{*1} \partial X_{*2} = 0$ in the homscedastic case. Thus, as shown in Equations (72) and (73), the GPR method can also provide a prediction of the derivative of the fit and its confidence interval simply by computing the first- and mixed second-order derivatives of the kernel function. The equations for doing so are summarized below:

$$\begin{aligned}
\mathbb{E}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \frac{\partial K(X_*, X')}{\partial X_*} \left[K' + \Sigma_n' \right]^{-1} Y' \\
\mathbb{V}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \frac{\partial^2 K_*}{\partial X_* \partial X_*} - \frac{\partial K(X_*, X')}{\partial X_*} \left[K' + \Sigma_n' \right]^{-1} \frac{\partial K(X', X_*)}{\partial X_*}
\end{aligned} \tag{74}$$

where the subscripts 1 and 2 were removed to improve readability.

3.2.5 The Noise Function and Heteroscedastic GPR

In contrast to the homoscedastic GPR given in Equation (60), a *heteroscedastic* GPR would allow each of the random variables to have its own specified variance, thus allowing the accounting of individual measurement errors. As the predicted variance, Σ_{n*}, in any GPR must defined consistently with the output noise variance, Σ_n, such a modification would require a way to estimate the predicted variance from the set of measurement errors. One method to achieve this is to apply a separate GPR to the set of measurement errors, (X, σ_Y), themselves. This generates an approximation for the output variance as a function of the independent variable, $r(x)$, expressed as such:

$$r(x) = \mathbb{E}[\sigma_y] = K_n(x, X) \left[K_n + \Sigma_\sigma\right]^{-1} \sigma_Y \tag{75}$$

where the *noise kernel*, K_n, does not necessarily have to be the same as the predictive GPR kernel, K, and it is usually assumed that the variance of the measurement error is zero, ie. $\Sigma_\sigma = 0$. Since Equation (60) requires the terms in matrix notation, the *heteroscedastic noise matrix*, $R(X_1, X_2)$, is introduced and is defined as follows:

$$R(X_1, X_2) = r(X_1)\, r(X_2)\, \delta(X_1 = X_2) \tag{76}$$

where δ is the Dirac delta function, as introduced in Equation (28).

Then, by substituting Equation (76) into Equation (60), the heteroscedastic predictive distribution can now be expressed as:

$$\begin{aligned} \mathbb{E}[Y_*] &= K(X_*, X) \left[K + R\right]^{-1} Y \\ \mathbb{V}[Y_*] &= K_* + R_* + K(X_*, X) \left[K + R\right]^{-1} K(X, X_*) \end{aligned} \tag{77}$$

where the shorthand notation $K = K(X,X)$, $R = R(X,X)$, $K_* = K(X_*, X_*)$, and $R_* = R(X_*, X_*)$ was used to improve the readability of the equation. Due to the introduced dependence of the predicted noise on X_*, the predicted derivatives are also modified, as follows:

$$\begin{aligned} \mathbb{E}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \frac{\partial K(X_*, X)}{\partial X_*} \left[K + R\right]^{-1} Y \\ \mathbb{V}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \frac{\partial^2 K_*}{\partial X_* \partial X_*} + \frac{\partial^2 R_*}{\partial X_* \partial X_*} + \frac{\partial K(X_*, X)}{\partial X_*} \left[K + R\right]^{-1} \frac{\partial K(X, X_*)}{\partial X_*} \end{aligned} \tag{78}$$

It should be noted that $\partial R_* / \partial X_* \partial X_*$ can result in extremely large derivative variances if the noise function, $r(x)$, behaves erratically. As such, it is advised that the kernels used for the GPR of the measurement errors be selected carefully to avoid this artificial inflation of the error.

Although this is not a typical scenario, this derivative error inflation problem can be circumvented provided that a sufficient number of derivative data points were supplied, each with their own uncertainty information. With this data, another GPR can be performed on their uncertainties to produce an approximative function for the derivative noise, $r_d(x)$. Then, $\partial R_* / \partial X_* \partial X_*$ in Equation (78) can simply be replaced with the following expression, evaluated at the points, X_*:

$$R_d = r_d(X_{d1})\, r_d(X_{d2})\, \delta(X_{d1} = X_{d2}) \tag{79}$$

which is typically exhibits a more stable behaviour, and comes from a similar ideology as in the justification of Equation (76).

3.2.6 The Log-Marginal-Likelihood and Fit Optimization

It should be noted that the GPR technique up until this point merely produces a model with the user-specified hyperparameters, θ, without any metnion of an optimal solution. Thus, in order to draw an analogy to the minimization problem outlined in Section 1.1, it is important to define a goodness-of-fit metric for this technique. One possible metric is the *log-marginal-likelihood (LML)*, mathematically expressed as follows:

$$\log p(Y' | X') = -\frac{1}{2} Y'^T \left[K' + R'\right]^{-1} Y' - \frac{1}{2} \log \left|K' + R'\right| - \frac{1}{2} N \log 2\pi \tag{80}$$

where the vertical brackets indicate the determinant of the enclosed matrix and N represents the number of data points, including any input derivative data.

A closer examination of this expression reveals that the first term quantifies an actual goodness-of-fit, due to its dependence on Y', the second term penalizes kernel complexity, which reduces the chance of overfitting to the data, and the last term represents a normalization constant to the size of the input data set, N.

As the LML represents the probability of the marginal likelihood, or the likelihood of the evidence integrated over all possible models described by this kernel, *maximizing* it selects the hyperparameters that have the highest probability of producing fits that match the input data. It is crucial to note that this goodness-of-fit metric, similar to the SSE given in Equation (2), provides no guarantee that the physical processes behind the data is modelled correctly by the chosen kernel.

The maximization process can be accelerated by employing the derivative of the log-marginal-likelihood with respect the hyperparameters, as such:

$$\frac{\partial}{\partial \theta_j} \log p(Y'|X') = \frac{1}{2} Y'^T \left[K' + R'\right]^{-1} \frac{\partial K'}{\partial \theta_j} \left[K' + R'\right]^{-1} Y' - \frac{1}{2} \mathrm{tr}\left(\left[K' + R'\right]^{-1} \frac{\partial K'}{\partial \theta_j}\right) \tag{81}$$

which requires the derivative of the kernel function with respect to the hyperparameters, $\partial k / \partial \theta_j$. In practical implementations, it has been found that using analytical hyperparameter derivatives have little impact on computational speed of the GPR, as compared to applying numerical approximations for this derivative, when the chosen kernel contains less than 15 hyperparameters. This is likely due to the relative complexity of the analytical expressions of the derivatives as compared to the kernel function itself. However, this fact has not been verified with all kernels.

If these derivatives, or *gradients*, with respect to the hyperparameters are calculated, then the most simple and robust method for performing this maximization is called the *gradient ascent* method. There exists other optimization algorithms which provide additional benefits, but these will not be discussed further in this document.

3.2.7 Regularization

Despite the use of an infinite set of basis functions, the GPR naturally avoids overfitting the data through using the kernel trick. However, this does not mean that the chosen kernel function itself cannot reinstate the danger of overfitting by being containing too many hyperparameters. In order to provide an additional defense against this undesired regression phenomena, an additional parameter called the *regularization parameter*, λ, can be added to the optimization “loss” function, represented by the LML described in Equation (80), as follows:

$$\log p(Y'|X') = -\frac{1}{2} Y'^T \left[K' + R'\right]^{-1} Y' - \frac{\lambda}{2} \log \left|K' + R'\right| - \frac{1}{2} N \log 2\pi \tag{82}$$

Note that since the second term of this equation effectively penalizes the complexity of the kernel, and hence the complexity of the resulting model, a larger value of λ increases the favourability of simpler models. Heuristically, for kernels with 5 or less hyperparameters, λ can range anywhere between 1 and 10 but typically does not exceed 15.

3.3 Summary

The predictive distribution of the GPR, given a set of input data points, (X, Y), derivative data points, $(X_d, \partial Y / \partial X_d)$, can be expressed as:

$$\begin{aligned} \mathbb{E}[Y_*] &= K(X_*, X') \left[K' + R'\right]^{-1} Y' \\ \mathbb{V}[Y_*] &= K_* + R_* + K(X_*, X') \left[K' + R'\right]^{-1} K(X', X_*) \end{aligned} \tag{83}$$

where

$$X' = \begin{bmatrix} X \\ X_d \end{bmatrix} \quad , \qquad Y' = \begin{bmatrix} Y \\ \frac{\partial Y}{\partial X_d} \end{bmatrix} \ , \tag{84}$$

$$K' \equiv K(X_1', X_2') = \begin{bmatrix} K(X_1, X_2) & \frac{\partial K(X_1, X_{d2})}{\partial X_{d2}} \\ \frac{\partial K(X_{d1}, X_2)}{\partial X_{d1}} & \frac{\partial^2 K(X_{d1}, X_{d2})}{\partial X_{d1} \partial X_{d2}} \end{bmatrix} \ , \tag{85}$$

$$R' \equiv R(X_1', X_2') = \begin{bmatrix} R(X_1, X_2) & 0 \\ 0 & R_d(X_{d1}, X_{d2}) \end{bmatrix} \ , \tag{86}$$

K and R are matrices with size equal to the number of elements in their first argument times the number of elements in their second argument, and $K' + R'$ must be a positive semi-definite square matrix to ensure its invertibility.

Typically, K is referred to as the kernel and is defined via a kernel function, $k(x_1, x_2, \theta)$, as such:

$$K(X_1, X_2) = \begin{bmatrix} k(x_{11}, x_{11}, \theta) & k(x_{11}, x_{22}, \theta) & \dots & k(x_{11}, x_{2m}, \theta) \\ k(x_{12}, x_{21}, \theta) & k(x_{12}, x_{22}, \theta) & & \\ \vdots & & \ddots & \\ k(x_{1n}, x_{21}, \theta) & & & k(x_{1n}, x_{2m}, \theta) \end{bmatrix} \tag{87}$$

where n is the number of elements in vector argument, X_1, m is the number of elements in vector argument, X_2, and θ contains a set of externally adjustable hyperparameters in order to fine tune the resulting fit. The kernel function can be chosen by the user to encourage specific behaviours in the learned models, with each kernel function exhibiting different general characteristics.

Similarly, R can be referred to as the noise kernel and is defined via a noise function, $r(x)$, as such:

$$R(X_1, X_2) = \begin{bmatrix} r(x_{11})\, r(x_{21}) & 0 & \dots & 0 \\ 0 & r(x_{12})\, r(x_{22}) & & \\ \vdots & & \ddots & \\ 0 & & & r(x_{1n})\, r(x_{2n}) \end{bmatrix} \tag{88}$$

where the two arguments must be identical in this function, ie. $X_1 = X_2$. It should be noted that the matrix, R_d, is identical to Equation (88), except that the noise function, $r(x)$, is replaced with $r_d(x)$, a function which describes the noise in the derivative observations. Typically, the noise function itself is not known from first principles, but it can be estimated by using a separate GPR with the input set, (X, σ_Y), where σ_Y represents the measurement uncertainty of the data set, Y. In this error estimation GPR, it is recommended to set $r(x) = \epsilon$, where $\epsilon \ll 1$ is a constant value in x, to ensure that the matrix, $K' + R'$, is properly conditioned for numerical inversion operations.

Once the inputs are clearly defined, the resulting fit can be optimized by adjusting the values of the kernel function hyperparameters, θ, such that it maximizes the log-marginal-likelihood, defined as follows:

$$\log p(Y'|X') = -\frac{1}{2} Y'^T \left[K' + R'\right]^{-1} Y' - \frac{\lambda}{2} \log |K' + R'| - \frac{1}{2} N \log 2\pi \tag{89}$$

where λ is called the regularization parameter, which can be increased to push the algorithm to select smoother fits, and N is the number of input data points provided to the GPR algorithm. By using this criteria for optimization, the final fit has the highest probability of producing fits which match the input data, but does not provide any guarantee that the fit correctly models the underlying physical processes described by the data.

The predictive distribution of the derivatives of the GPR can also be computed through a similar methodology, provided that the kernel function used is differentiable either numerically or analytically, as follows:

$$\begin{aligned} \mathbb{E}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \frac{\partial K(X_*, X')}{\partial X_*} L'^{-1} Y \\ \mathbb{V}\left[\frac{\partial Y_*}{\partial X_*}\right] &= \frac{\partial^2 L_*}{\partial X_* \partial X_*} + \frac{\partial K(X_*, X')}{\partial X_*} L'^{-1} \frac{\partial K(X', X_*)}{\partial X_*} \end{aligned} \tag{90}$$

where $L \equiv K + R$ was used to improve the readability, which is further extended to provide the definitions, $L_* = K_* + R_*$ and $L' = K' + R'$.