UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: Functional Estimation of Probabilistic Solutions in Ordinary Differential Equations Models

Trabajo de integración curricular presentado como requisito para la obtención del título de Matemática

Autor:

Cedeño Manrique Helen Naomi

Tutor:

Infante Quirpa Saba Rafael, Ph.D.

Urcuquí, enero de 2022

Urcuquí, 27 de enero de 2022

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES
CARRERA DE MATEMÁTICA
ACTA DE DEFENSA No. UITEY-ITE-2022-00005-AD

A los 27 días del mes de enero de 2022, a las 11:30 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dr. ACOSTA ORELLANA, ANTONIO RAMON , Ph.D.
Miembro No Tutor	Dr. MANZANILLA MORILLO, RAUL , Ph.D.
Tutor	Dr. INFANTE QUIRPA, SABA RAFAEL , Ph.D.

El(la) señor(ita) estudiante **CEDEÑO MANRIQUE, HELEN NAOMI**, con cédula de identidad No. **1316368495**, de la **ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES**, de la Carrera de **MATEMÁTICA**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-15-No.174-2015**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **FUNCTIONAL ESTIMATION OF PROBABILISTIC SOLUTIONS IN ORDINARY DIFFERENTIAL EQUATIONS MODELS.**, previa a la obtención del título de **MATEMÁTICO/A**.

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. INFANTE QUIRPA, SABA RAFAEL , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	Docente	Calificación
Presidente Tribunal De Defensa	Dr. ACOSTA ORELLANA, ANTONIO RAMON , Ph.D.	10,0
Miembro Tribunal De Defensa	Dr. MANZANILLA MORILLO, RAUL , Ph.D.	10,0
Tutor	Dr. INFANTE QUIRPA, SABA RAFAEL , Ph.D.	10,0

Lo que da un promedio de: **10 (Diez punto Cero)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

Certifico que *en cumplimiento del Decreto Ejecutivo 1017 de 16 de marzo de 2020, la defensa de trabajo de titulación (o examen de grado modalidad teórico práctica) se realizó vía virtual, por lo que las firmas de los miembros del Tribunal de Defensa de Grado, constan en forma digital.*

Firmado electrónicamente por:
HELEN NAOMI CEDENO MANRIQUE

CEDEÑO MANRIQUE, HELEN NAOMI
Estudiante

ANTONIO RAMON ACOSTA ORELLANA
Firmado digitalmente por ANTONIO RAMON ACOSTA ORELLANA
Fecha: 2022.01.27 13:10:57 -05'00'

Dr. ACOSTA ORELLANA, ANTONIO RAMON , Ph.D.
Presidente Tribunal de Defensa

Dr. INFANTE QUIRPA, SABA RAFAEL , Ph.D.
Tutor

SABA RAFAEL INFANTE QUIRPA
Firmado digitalmente por SABA RAFAEL INFANTE QUIRPA
Fecha: 2022.01.27 13:57:18 -05'00'

Dr. MANZANILLA MORILLO, RAUL , Ph.D.
Miembro No Tutor

Firmado electrónicamente por:
RAUL MANZANILLA

TORRES MONTALVÁN, TATIANA BEATRIZ
Secretario Ad-hoc

TATIANA BEATRIZ TORRES MONTALVAN
Firmado digitalmente por TATIANA BEATRIZ TORRES MONTALVAN
Fecha: 2022.01.27 13:11:18 -05'00'

Autoría

Yo, **Helen Naomi Cedeño Manrique**, con cédula de identidad **1316368495**, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así como los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el autor del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

San Miguel de Urcuquí, septiembre del 2021.

Firmado electrónicamente por:
HELEN NAOMI CEDENO MANRIQUE

Helen Naomi Cedeño Manrique

CI: 1316368495

Autorización de publicación

Yo, **Helen Naomi Cedeño Manrique**, con cédula de identidad **1316368495**, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

San Miguel de Urcuquí, septiembre del 2021.

Firmado electrónicamente por:
HELEN NAOMI
CEDENO
MANRIQUE

Helen Naomi Cedeño Manrique

CI: 1316368495

Dedicatoria

A mi amada familia, que cambió los abrazos de cada día por un sueño que estaba a ocho horas de casa. Todo lo escrito aquí es tan suyo como mío.

A todas las chicas a las que se les hizo creer que no podían aspirar a más. Crean en ustedes mismas incluso cuando sientan que no dan más, no existe techo de cristal que no puedan romper.

Agradecimientos

Después de cinco maravillosos años y de disfrutar una de las mejores experiencias de mi vida, son muchas las personas a las que me gustaría dar las gracias por haberme ayudado, cada una a su manera, a llegar al momento en el que escribo esto.

En primer lugar, quisera expresar mi gratitud infinita hacia las personas más importantes en mi vida: mi familia. A mis padres, Jorge y Helen, por todo el amor, dedicación y paciencia. No me alcanzarían las palabras para expresarles lo afortunada que soy al tenerlos. A mi hermano Miguel, por brindarme su compañía y alegrarme con sus ocurrencias. A mi hermana Melanie, no sé que habría sido de mí sin su cariño, sus consejos y sus locuras, fue una bendición poder recorrer esta etapa juntas. Gracias por ser la mejor compañía y por creer en mí incluso cuando yo no lo hacía.

Agradezco profundamente a la comunidad Yachay Tech, tomar la decisión de formarme en esta universidad cambió mi vida. Gracias a cada uno de los maestros que demostró el deseo por construir una sociedad mejor a través de nosotros, y esa fue la mejor motivación para seguir aprendiendo. Entre ellos, quisiera agradecer al Dr. Saba Infante, mi tutor, por su invaluable ayuda en cada paso de la carrera, desde el descubrimiento de mi vocación por la estadística, hasta la dedicación y comprensión en el desarrollo de este trabajo. Además, un reconocimiento especial al Dr. Juan Mayorga y al Dr. Antonio Acosta, quienes me inspiraron a dejar atrás los estigmas de las matemáticas y conocer su verdadera belleza, y que me han ofrecido su apoyo incondicional.

Gran parte de lo emocionante en este viaje se debe a las increíbles personas con las que coincidí. Mi eterno aprecio y agradecimiento a quienes se convirtieron en mi hogar: a Andrés, Argenis, Belén, Carlos, Daniel, Dayana, Gabriela, Harvey, Marcelo, Ricardo y con especial

afecto a Jean Carlos y Lady. Cada memoria que creamos juntos se quedó en la patrimonial 38, pero también se hizo un lugar permanente en mi corazón. De la misma manera, gracias a Katherine y a Salomé por esa amistad inesperada que se convirtió en la mejor compañía durante este proceso. Gracias incondicionales a Adrián, Andy y Génesis por su amistad y constante soporte, especialmente en las interminables noches entre teoremas y algoritmos. Espero que su futuro sea tan brillante como ustedes.

Finalmente, quiero agradecer a todo aquel que en algún momento dejó su huella en mi camino. Gracias por el pedacito de ustedes que estará siempre aquí, junto a todos los recuerdos que recolecté durante esta maravillosa aventura.

Naomi Cedeño Manrique

Abstract

Probabilistic approaches based in derivative modeling are wide-used techniques in the estimation of solutions of systems of differential equations whose numerical solution has an intractable computational complexity or in which the presence of error or infinitesimal perturbations could result in divergences. The quantification of the uncertainty that is produced when estimating the solution in a finite temporal mesh is an open problem and has been addressed from various probabilistic approaches. In this work, uncertainty estimation of solutions of ordinary differential equations by means of a Gaussian process in a space of Lipschitz functions is addressed by implementing an algorithm that allows estimating the solution states $x(t)$ and their derivatives in a sequential way. Besides, the addition of polynomial chaos expansions (PCE) using the resulting distributions of the algorithm is proposed to improve its prediction accuracy. To illustrate the methodology, algorithms were tested on three known systems of ordinary differential equations and their effectiveness was quantified by three performance measures, resulting in an overall improvement in prediction by adding the polynomial chaos expansion.

Keywords: Functional estimation, Uncertainty quantification, Gaussian process, Polynomial chaos expansion.

Resumen

Las aproximaciones probabilísticas basadas en el modelado de derivadas son técnicas ampliamente utilizadas para la estimación de soluciones de sistemas de ecuaciones diferenciales cuya solución numérica tiene una complejidad computacional intratable o en la que la presencia de errores o perturbaciones infinitesimales puede provocar divergencias. La cuantificación de la incertidumbre que se produce al estimar la solución en una malla temporal finita es un problema abierto y ha sido abordado desde diversos enfoques. En este trabajo se aborda la estimación de la incertidumbre de las soluciones de modelos ecuaciones diferenciales ordinarias mediante procesos Gaussianos en espacios de funciones Lipschitzianas, implementando un algoritmo que permite estimar los estados de la solución $x(t)$ y sus derivadas de forma secuencial. Además, se propone la adición de expansiones de caos polinomial (PCE) utilizando las distribuciones resultantes del algoritmo para mejorar la precisión de su predicción. Para ilustrar la metodología, se probaron los algoritmos en tres sistemas de ecuaciones diferenciales ordinarias conocidos y se cuantificó su eficacia a través de tres medidas de rendimiento, obteniendo como resultado una mejora global de la precisión al añadir la expansión de caos polinomial.

Palabras Clave: Estimación funcional, Cuantificación de incertidumbre, Procesos Gaussianos, Expansión de caos polinomial.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Continuous dynamical systems model several phenomena in applied fields of science such as physics, climatology, finance, biology, chemistry, engineering, and medicine. Modeling their dynamic nature can become a highly complex task due to the possible presence of chaos components, and is usually characterized through systems of ordinary differential equations. These models involve a function of given observations with their respective derivatives, describing the probabilistic dependence between each of the partially observed states and their rates of change within a spatio-temporal domain [1].

The solutions of the systems of equations represent the transition of each of the states within a time interval, and while some can be obtained by analytical methods, this is not always possible. Furthermore, many of the ordinary differential equation models exhibit highly structured but clearly unpredictable behavior [2]. To estimate the response of each system, numerical methods are the most commonly used set of techniques. However, the numerical discretization process involves the introduction of perturbations, which, although small, can lead to an exponential divergence of the solution [3] and could represent an intractable computational cost. This error, known as uncertainty, is generally not estimated so its quantification has become the object of study of other types of modeling.

In this work, we study and combine two probabilistic estimation methods of uncertainty quantification in the discretization of solutions by using algorithms based on Gaussian processes, whose effectiveness will be demonstrated through their application in nonlinear dynamical

systems widely known in experimental fields, which will be explained in more detail on subsequent sections.

1.1 Problem Statement

Consider a continuous-time dynamical system relating the d derivatives $\dot{x}(t) \in \mathbb{R}^d$, with respect to space-time varying variables (x, t) in the domain $\mathcal{D} \subset \mathbb{R}^d$, through state functions $x(t) \in \mathbb{R}^d$ in a vector field of smoothed continuous functions differentiable for x and indexed by unknown parameters θ in the parameter set $\Theta \subset \mathbb{R}^p$, denoted as:

$$\mathbf{f} = (f(x_1), \ldots, f(x_d)), \quad f : \mathcal{D} \times \Theta \to \mathbb{R}^d \tag{1.1}$$

such that these satisfy the Lipschitz condition [4].

Definition 1.1.1. (Lipschitz Condition) A function $f \in \mathcal{D} \subset \mathbb{R}^d$ is said to be **Lipschitzian** in $\mathcal{D}$ if there exists a constant K such that

$$|f(x) - f(y)| < K|x - y|$$

for all $x, y \in \mathcal{D}$.

If one considers an initial value problem; that is, an ordinary differential equation (ODE) that satisfies:

$$\begin{aligned} &\frac{dx}{dt} \equiv \dot{x}(t) = f\left(x(t), \theta\right), \quad t \in [0, T] \\ &x(t_0) = x_0 \in \mathbb{R}^d \end{aligned} \tag{1.2}$$

The solution states of the system are $x(t) \equiv (x_1(t), \ldots, x_d(t))$, and evolve according to the model described in (1.2). These solutions are generally not available analytically. Systems involving derivatives of x of order $n > 1$ can be reduced to the expression given in (1.2), by using the following changes of variable:

$$\begin{aligned} x_1 &= x, \\ x_2 &= \dot{x}_1, \\ &\vdots \\ x_n &= \dot{x}_{n-1} \end{aligned}$$

Definition 1.1.1 guarantees the existence and uniqueness of the solutions of (1.2) given an initial value x_0, for the neighborhood $(0, x_0)$. However, most systems does not have defined solutions, which implies a significant increase in the computational cost of the data fitting methodology [4]. Conventionally, the exact solution is replaced by an approximate solution $\hat{x}(t)$ obtained using some numerical technique on a partition of a grid on the domain $\mathcal{D}$, and the inference and prediction processes are based on this approximated solution.

Consider an ordinary differential equation of the type:

$$\frac{dx}{dt} = f\left(x(t)\right), \quad x(0) = x_0 \tag{1.3}$$

where x is a continuous function that takes values from $\mathbb{R}^d$, ϕ_t denotes the flow map of the ordinary differential equation given in (1.3), and $x_t = \phi_t\left(x(0)\right)$ is the solution of the equation. The classical deterministic numerical methods used to find a solution of the equation (1.3) in the time interval $[0, T]$ allow to obtain an approximation of the solution on a temporal mesh $\{t_k = kh\}_{k=1}^{K}$, where $Kh = T$.

In this system, $x_k = x(t_k)$ is the solution on the mesh based on the evaluation of f and possibly on its higher order derivatives in a set of finite points that are generated by some numerical integration technique. This methodology generates a discrete solution $\{x_t\}_{t=1}^{T}$ which is unique but could have a high estimation error rate. This is because although the numerical error analysis provides local and global error bounds within the discretized mesh [5] that allow approximating the asymptotic behavior of the model, the verification error in its inference is still an open problem [6]. This prevents quantifying the uncertainty over the rest of the solution trajectory, causing it to be ignored in practice.

Let $\mathcal{X}_{a,b}$ be the Banach space $\mathcal{C}\left([a, b]; \mathbb{R}^d\right)$, the solution of the equation (1.3) over an interval $[0, T]$ can be interpreted as a Dirac measure δ_x over the space $\mathcal{X}_{0,T}$, where each element x is a solution of the ordinary differential equation. Then, a probability measure μ^h on the space $\mathcal{X}_{0,T}$ can be defined, from which random samples can be generated inside and outside the discretized mesh. In this way, the size of each step in the discretization can be quantified, and through the sampling distribution the uncertainty of the remaining solution of the differential equation can be quantified [7].

From a probabilistic point of view, the integral form of the equation (1.3) is given by:

$$x(t) = x_0 + \int_0^t f\left(x(s)\right) ds \tag{1.4}$$

Discretizing (1.4) into a finite set of steps, the solutions in the mesh satisfy:

$$x_{k+1} = x_k + \int_{t_k}^t f\left(x(s)\right) ds, \quad t \in [t_k, t_{k+1}] \tag{1.5}$$

The equation (1.5) can be rewritten as follows:

$$x_{k+1} = x_k + \int_{t_k}^t g\left(s\right) ds, \quad t \in [t_k, t_{k+1}] \tag{1.6}$$

where $g\left(t\right) = f\left(x(t)\right)$ is an unknown function that evolves over a time point t. The function $g\left(t\right)$ can be approximated by some numerical method so that it is represented as:

$$\begin{aligned} g^h(s) &= \frac{d}{d\tau}\left[\psi_\tau\left(X_k\right)\right]_{\tau=s-t_k}, \quad s \in [t_k, t_{k+1}] \\ X_{k+1} &= \psi_k\left(X_k\right) \end{aligned} \tag{1.7}$$

where $\psi : \mathbb{R}^d \to \mathbb{R}^d$ denotes a deterministic classical numerical integrator of a time step h, a class that includes all methods of numerical integration of ordinary differential equations.

To approximate the function g in stochastic form, equation (1.7) can be modified as follows:

$$\begin{aligned} g^h(s) &= \frac{d}{d\tau}\left[\psi_\tau\left(X_k\right)\right]_{\tau=s-t_k} + \omega_k\left(\tau\right), \quad s \in [t_k, t_{k+1}] \\ \omega_k &\sim N\left(0, K^h(s,t)\right) \end{aligned} \tag{1.8}$$

where $\{\omega_k\}$ is a sequence of functions of Gaussian random variables defined on the interval $[0, h]$. On the other hand, the covariance $K^h(s,t)$ is chosen such that it approaches zero at a set rate of speed h, in order to be able to ensure that $\omega \in \mathcal{X}_{0,T}$. Then, the functions $\{\omega_k\}$ represent the uncertainty about g.

In models that characterize real-world behaviors, the solution states of the system are unknown and only partially observed, with a certain error ϵ_t that does not allow to obtain the exact measurement of x_t. Considering a lattice of discrete points $0 < t_0 < t_1 < \ldots < t_n$ in which measurements $y_1, y_2, \ldots, y_n$ are obtained from observations $x_0, x_1, \ldots, x_n$, with $x_i \equiv x_{t_i}$ y $y_i \equiv y_{t_i}$, the values of y_i can be taken from a probability model of the form:

$$y_i = h\left(x_i, \epsilon_i\right), \quad i \in \{1, \ldots, n\} \tag{1.9}$$

where $h(\cdot)$ is a known vector function of real variable, ϵ_i are independent and identically distributed variables drawn from some probability distribution, and $\{y_i\}_{i=1}^n$ are conditionally independent given $\{x_i\}_{i=1}^n$. For the $k-$th observed state, the n observations are obtained through the following statistical model:

$$\begin{aligned} y(t) &= h\left(x(t)\right) + \epsilon(t) \\ \epsilon(t) &\sim N\left(0, \sigma_\epsilon^2\right) \end{aligned} \tag{1.10}$$

This model is made up of a vector of latent states $X \equiv (x(t_1), x(t_2), \ldots, x(t_n))^T$ where the $k-$th state is given by $x_k = (x_k(t_1), x_k(t_2), \ldots, x_k(t_n))^T$, and by an observation vector $Y \equiv (y(t_1), \ldots, y(t_n))^T$ where the $k-$th observation is $y_k = (y_k(t), \ldots, y_k(t))^T$, for $k \in \{1, 2, \ldots, K\}$. Based on the above, the nonlinear state-space model in compact form would be represented as:

$$\begin{aligned} \dot{x}(t) = \frac{dx}{dt} &= f\left(x(t)\right), \quad x(t_0) = x_0, \quad t \in [0, T] \\ y(t) &= h\left(x(t)\right) + \epsilon(t) \\ \epsilon(t) &\sim N\left(0, \sigma_\epsilon^2\right) \end{aligned} \tag{1.11}$$

The problem (1.11) can be formulated in terms of a discrete-time stochastic dynamical system by the following state-space model:

$$\begin{aligned} x_k &= f\left(x_{k-1}\right) + \nu_{k-1}, \quad (\textit{system state}) \\ \nu_{k-1} &\sim N\left(0, Q\right) \\ y_k &= h\left(x_k\right) + \epsilon_{k-1}, \quad (\textit{system measurement}) \\ \epsilon_k &\sim N\left(0, R\right) \\ x_0 &\sim N\left(m_0^x, k_0^x(s_0, t_0)\right) \end{aligned} \tag{1.12}$$

where $x_k \in \mathbb{R}^{d_x}$, $\nu_{k-1} \in \mathbb{R}^{d_y}$, $y_k \in \mathbb{R}^{d_y}$, $\epsilon_k \in \mathbb{R}^{d_y}$, $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$, and $h : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$.

The Bayesian solution of the filtering problem can be obtained in terms of prediction and updating. Thus, the filtered distribution of the posterior state is given by:

$$p\left(x_k | y_{1:k}\right) = \frac{p\left(y_k | x_k\right) p\left(x_k | y_{1:k-1}\right)}{p\left(y_k | y_{1_k - 1}\right)} \tag{1.13}$$

where the likelihood $p\left(y_k|x_k\right)$ is obtained from the observation equation of (1.12) and $y_{1:k} = (y_1, \ldots, y_k)$. On the other hand, the predictive density is obtained by the Chapman-Kolmogorov equation:

$$p\left(x_k|y_{1:k-1}\right) = \int p\left(x_k|x_{k-1}\right) p\left(x_k|y_{1:k-1}\right) dx_{k-1} \tag{1.14}$$

where the transition density $p\left(x_k|x_{k-1}\right)$ is obtained from the state equation of (1.12). This allows to obtain a model of the observations given by:

$$L\left(Y|X,\sigma^2\right) = \prod_{k=1}^{K_0}\prod_{i=1}^{n} p\left(y_k(t_i)|x_k(t_i),\sigma_k^2\right) \tag{1.15}$$

1.2 Testing Models

The above mentioned issue will be addressed by testing the methods on three differential equation models widely known in different areas of science, such as mathematics, physics, medicine and biology.

1.2.1 Lorenz Attractor Model

The classical Lorenz attractor model, proposed by Eduard Lorenz in 1963, simulates the fluid motion induced by the temperature difference between lower and upper surfaces. Mathematically, it can be described as a coupled system of nonlinear differential equations expressed as:

$$\begin{aligned} \dot{x(t)} &= \frac{dx}{dt} = \sigma(y(t) - x(t)) \\ \dot{y(t)} &= \frac{dy}{dt} = \rho x(t) - y(t) - x(t)z(t) \\ \dot{z(t)} &= \frac{dz}{dt} = -\beta z(t) + x(t)y(t) \end{aligned} \tag{1.16}$$

where $\sigma, \rho, \beta \in \mathbb{R}^+$ are parameters known as the Prandtl coefficient, Rayleigh coefficient and radius aspect ratio, respectively. The state vector $(x(t), y(t), z(t))^T$ represents the position of the particles in phase space, and usually points are chosen at which the attractor exhibits chaotic behavior [8].

In (1.16), the states $x(t)$, $y(t)$ y $z(t)$ are proportional, respectively, to the intensity of the convective motion of the fluids, the temperature difference between the updrafts and downdrafts, and the deviation of the vertical temperature profile from linearity.

1.2.2 Kermack-McKendrick SIR Model

The SIR compartmental epidemiological model, proposed by William Kermack and Anderson McKendrick in 1927, aims at predicting the evolution of infectious diseases transmitted from person to person. Under the assumption of suppression of life dynamics, i.e. that the sample population remains constant, this deterministic system of nonlinear ordinary differential equations is specified as follows:

$$\begin{aligned} \dot{S(t)} &= \frac{dS}{dt} = -\beta S(t)I(t) \\ \dot{I(t)} &= \frac{dI}{dt} = \beta S(t)I(t) - \gamma I(t) \\ \dot{R(t)} &= \frac{dR}{dt} = \gamma I(t) \\ N(t) &= S(t) + I(t) + R(t) \end{aligned} \tag{1.17}$$

where a population of $N(t) \in \mathbb{R}^+$ individuals are classified according to various disjoint statuses by disease pairs at a point in time t. Within the population, $S(t)$ represents the set of individuals susceptible to a given disease, while $I(t)$ represents the infectious population, and $R(t)$ the inhabitants who are considered recovered by developing immunity to the disease [9]. Also, $\beta, \gamma \in \mathbb{R}^+$ are fixed parameters, which represent the infection rate and the recovery rate.

1.2.3 FitzHugh–Nagumo Model

FitzHugh-Nagumo equations, initially proposed by Richard FitzHugh in 1961 and supplemented by Jin-Ichi Nagumo in 1962, are a particularization of the Hodgkin-Huxley generation models of the action potential in the axon of the giant squid [10] and describe the excitation and propagation properties of neurons in the face of electrochemical reactions. These are

characterized as a system of two coupled nonlinear differential equations, given by:

$$\begin{aligned} \dot{v}(t) = \frac{dv}{dt} &= v(t) - \frac{v^3(t)}{3} - w(t) + I \\ \dot{w}(t) = \frac{dw}{dt} &= \phi(v(t) + a - bw(t)) \end{aligned} \tag{1.18}$$

where $v(t)$ describes the evolution of the voltage at the neuronal membrane, which is usually fast; while $w(t)$ simulates the recuperative action between the sodium channel deactivation and potassium channel deactivation currents [10]. In (1.18), $a, b, \phi \in \mathbb{R}^+$ are constant parameters, and $I \in \mathbb{R}^+$ represents the external current applied.

1.3 Thesis Structure

This work is divided into six chapters, which are distributed as follows: Introduction, Theoretical Framework, State of the Art, Methodology, Results, and Conclusions.

Chapter 1 introduces dynamic systems, which are the means of study and testing of the methods to be used. In addition, the problem of uncertainty quantification and the necessary conditions for its solution are determined, as well as the contributions made to the area of functional data analysis through the improvement of existing algorithms. Finally, the general and specific objectives that were met during the writing of this work are determined.

In Chapter 2, the main concepts of this work are detailed, making a formal review about Bayesian inference, functional data analysis, the forward problem, and stochastic processes. Within the latter, special emphasis will be placed on Gaussian processes, whose fundamentals will be discussed, as well as the properties of the mean, kernels and covariance functions. In addition, fundamental concepts related to Polynomial Chaos Expansions, such as orthogonal and orthonormal polynomials and basis, will be introduced.

Chapter 3 presents the historical background of the development of the theory of functional estimation and uncertainty quantification, starting with the evolution of probability theory towards Bayesian statistics, the characterization of systems of ODEs as inference problems and the use of Gaussian processes (GP), moving on to the rise of Bayesian Data Analysis, the extension of probabilistic estimation methods in ODE models to higher dimensionalities and

the problem of computational complexity, and concluding with the use of the aforementioned techniques within artificial neural networks.

In Chapter 4, the methodology is formulated, which include the calculation of the Gaussian process prior distributions, the characterization of the derivatives of the observations vector, the formulation of the Uncertainty Quantification for Differential Equations (UQDES) algorithm, which will be detailed for each iteration, and how the GP-based approaches work together with PCE to compute new distributions for the mean vector and the covariance matrix. Then, the computational requirements of this project, the particularities of the time complexity and performance measures will be defined.

Chapter 5 shows the results obtained from the execution of the UQDES algorithm, as well as those derived from the addition of PCE; defining first the time interval, the vectors of initial conditions, the values of the hyperparameters of each of the systems used. Then, a graphical comparison between the solutions obtained by the differential equations system solver of the *scipy* library with 2000 realizations of each one of the techniques is presented, both in the three-dimensional model and in each one of the solution functions separately; followed by the error rates reported by each one of the performance measures.

Finally, in Chapter 6 the conclusions obtained from the analysis and discussion of the results are presented. Likewise, future work that could be carried out in the area of probabilistic modeling of solutions, including more advanced applications in emerging areas of computer science such as artificial intelligence, is proposed.

1.4 Objectives

1.4.1 General Objective

Implement the UQDES algorithm and consider the addition of Polynomial Chaos Expansions (PCE) to perform functional estimation and uncertainty quantification of solution states in systems of ordinary differential equations.

1.4.2 Specific Objectives

The general objective described above leads to the implementation of the specific objectives detailed below:

- Discretize three systems of ordinary differential equations with known initial conditions and parameters to obtain point solutions of their states.
- Discuss the computational time complexity of UQDES and PCE algorithms.
- Quantify the accuracy of each algorithm through goodness-of-fit measures and compare the efficiency of these methods.

1.5 Contributions

This thesis consists of an extension of the theoretical background of the following publication:

> Cedeño, N. & Infante, S. (2021) Estimation of Ordinary Differential Equations Solutions with Gaussian Processes and Polynomial Chaos Expansion. In: Rodriguez G., Fonseca C., Salgado J., Pérez-Gosende P., Orellana M. (eds) Information and Communication Technologies. TICEC 2021. Communications in Computer and Information Science. Springer, Cham.

This research project performs a review of mathematical methods of functional estimation based on Bayesian methods and Gaussian processes, contributing to the study of the theory behind uncertainty quantification, derivative inference and the prediction of probabilistic solutions, which for practical reasons is not usually detailed in the studies that address their applications. Regarding the experimental part, this work adapts the UQDES algorithm [2] to the use of multidimensional arrays in Python, aiming to improve the execution times presented in the literature. In addition, the addition of the PCE technique to the results obtained is considered, which had not been considered in previous research papers, in order to verify if there are improvements in the prediction rate, which will be quantified by three goodness-of-fit measures that compare the accuracy of the UQDES algorithm with the UQDES+PCE.

Chapter 2

Theoretical Framework

2.1 Bayesian Inference

Bayesian inference is a method of statistical analysis that uses prior knowledge, in the form of probabilistic distributions, about a random phenomenon to infer the probability of a hypothesis being fulfilled by dynamic sequence analysis. This reasoning is entirely based on the assumptions of probability calculus, and in particular of Bayes' theorem.

Theorem 2.1.1. *(Bayes Theorem) Let* $\{A_1, \ldots, A_k\}$ *a set of events that form a partition of the sample space* Ω *and* B *an event with* $p(B) \geq 0$*, we have that:*

$$p(A_i|B) = \frac{p(B|A_i)p(A_i)}{p(B)} \tag{2.1}$$

If we consider $X = (x_1, \ldots, x_n)$ as a data vector and $\Theta = (\theta_1, \ldots, \theta_p)$ as a set of unknown parameters, we can define $\mathcal{F}$ as the respective density functions of X indexed by θ, so that:

$$\mathcal{F} = \{p(x \mid \theta);\ x \in X, \theta \in \Theta\} \tag{2.2}$$

Under the assumption that all variables in X are independent and identically distributed over θ, the model (2.2) can be rewritten in terms of the marginal densities of each element of X:

$$\mathcal{F} = \left\{ p\left(x_1, \ldots, x_n \mid \theta\right) = \prod_{i=1}^{n} f_i\left(x_i \mid \theta\right);\ x \in X, \theta \in \Theta \right\} \tag{2.3}$$

In Bayesian inference, $\mathcal{F}$ is considered as part of the model, i.e., that $p(x_i|\theta)$ will have a subjective interpretation. Applying Theorem 2.1.1 and assuming that the parameters are continuous, it follows that:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int_\theta p(X|\theta)p(\theta)d\theta} \tag{2.4}$$

where $p(\theta)$ is known as a prior distribution, which will be updated by observing the data from $p(\theta|X)$, the standardized posterior distribution.

By omitting $p(X)$ since it does not depend on θ, and fixing the values of y, we can define the non-normalized posterior density as:

$$p(\theta|X) \propto p(\theta)p(X|\theta) \tag{2.5}$$

To make predictive inferences on unknown but observable data, it is necessary to consider the marginal distribution of $X = x$, also known as the predictive prior, which is given by:

$$\begin{aligned} p(x) &= \int p(x, \theta)d\theta \\ &= \int p(\theta)p(x|\theta)d\theta \end{aligned} \tag{2.6}$$

Once x is observed, we can perform the inference process. Let $\tilde{X} = \tilde{x}$ be a set of new data, $p(\tilde{x}|x)$ it is defined as the posterior predictive conditional distribution on x:

$$\begin{aligned} p(\tilde{x}|x) &= \int p(\tilde{x}, \theta|x)d\theta \\ &= \int p(\tilde{x}|\theta, x)p(\theta|x)d\theta \\ &= \int p(\tilde{x}|\theta)p(\theta|x)d\theta \end{aligned} \tag{2.7}$$

According to (2.7), this distribution can be interpreted as the average of the conditional predictions made over the posterior distribution of θ [11].

2.1.1 Space-State Models

State-space models are a class of probabilistic models that provide a framework for the analysis of deterministic and non-deterministic systems observed by stochastic processes [12].

Definition 2.1.2. *(Space-State Model)* Let $\mathbf{x}_t \in \mathbb{R}^m$ be a set of unobserved variables, $\mathbf{y}_t \in \mathbb{R}^m$ a set of observed variables, both evaluated at point in time $t \in \{1, \ldots, T\}$, and m representing the number of states, a state space model is defined as:

$$\begin{aligned} \mathbf{x}_1 &\sim p(\mathbf{x}_1) \\ \mathbf{x}_t|\mathbf{x}_{t-1} &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ \mathbf{y}_t|\mathbf{x}_t &\sim p(\mathbf{y}_t|\mathbf{x}_t) \end{aligned} \tag{2.8}$$

where $p(\mathbf{x}_1)$ is known as the initial distribution, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the transition distribution and $p(\mathbf{y}_t|\mathbf{x}_t)$ is the observation distribution.

The (2.8) model assumes conditional independence between observations and aims to compute an optimal estimate of the unobserved states given the observed data by recursively applying Bayes' theorem [13], resulting in filtering distributions.

There are several cases derived from the general state-space model, and this project will focus particularly on the linear Gaussian state-space model. This model, also known as linear Gaussian dynamical system, is known to result in closed-form filtering and smoothing formulas known as Kalman filter and Kalman smoother [14]. It is described as:

$$\begin{aligned} \boldsymbol{x}_1 &\sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{Q}_1) \\ \boldsymbol{x}_t &= \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{\epsilon}_t \\ \boldsymbol{y}_t &= \boldsymbol{B}\boldsymbol{x}_t + \boldsymbol{w}_t \\ \boldsymbol{\epsilon}_t &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_t) \\ \boldsymbol{w}_t &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}_t) \end{aligned} \tag{2.9}$$

where $\boldsymbol{A}$ y $\boldsymbol{B}$ are dimensionally compatible matrices, and $\boldsymbol{Q}_1$, $\boldsymbol{R}_t$ and $\boldsymbol{S}_t$ are covariance matrices.

2.1.2 Bayesian Filtering

Bayesian filtering, or Bayesian recursive estimation, is a statistical inversion method that seeks to estimate the true value of the solution states of a system from its observations by performing a Bayesian analysis of probability density functions.

Let $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ a vector of unknown values observed by $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$, a set of noisy measurements. Consider $\mathbf{x}_t = (x_t, \dot{x}_t)^\top$ as a two-dimensional vector composed of each observation x_t and its derivative $\dot{x}_t$. We can formulate a discretized system given by:

$$\begin{aligned} \boldsymbol{x}_t &= f_t\left(\boldsymbol{x}_{t-1}, \boldsymbol{\nu}_{t-1}\right) \\ \boldsymbol{y}_t &= g_t\left(\boldsymbol{x}_t, \boldsymbol{\eta}_k\right) \end{aligned} \tag{2.10}$$

where $\nu_t \in \mathbb{R}^n$ is a randomly given noise vector, and $\eta_t \in \mathbb{R}^m$ is the noise vector of the observations. Then, we can define f_t and g_t as the process and observation functions. If we consider that these are invariant with respect to time, then $f_t = f$ and $g_t = g$ [15].

Bayesian filtering models can be performed recursively when the evolution of observations is time-dependent. This method is known as Bayesian Sequential Filtering, and seeks, through the prediction and update steps, to compute the smoothing, filtering and prediction distributions [16, 17] to subsequently estimate $p\left(\boldsymbol{x}_t, \boldsymbol{y}_{1:t}\right)$.

Definition 2.1.3. *(Smoothing)* Let $t < T$, the smoothing distribution is obtained through the marginal distributions of $\mathbf{x}_t$ given $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$, such that:

$$p\left(\mathbf{x}_t | \mathbf{y}_{1:T}\right), \quad t \in \{1, \ldots, T\} \tag{2.11}$$

It is the a posteriori estimation, where the data observed after the time index of interest is used.

Definition 2.1.4. *(Filtering)* The filtering distribution is calculated using the marginal distributions of state $\mathbf{x}_t$ given the observations $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$. Thus, it follows that:

$$p\left(\mathbf{x}_t | \mathbf{y}_{1:t}\right), \quad t \in \{1, \ldots, T\} \tag{2.12}$$

It is the operation that quantifies the information about a value of interest at a given time t using all the observed data in the interval $[0, t]$.

Definition 2.1.5. *(Prediction)* Let $\tau > 0$ and $n \in \mathbb{Z}^+$ the number of steps, the prediction distribution is estimated through the marginal distributions of a future state $\mathbf{x}_{t+n}$ after n steps. Then:

$$p\left(\mathbf{x}_{t+n} \mid \mathbf{y}_{1:t}\right), \quad t \in \{1, \ldots, T\} \tag{2.13}$$

It is the prior estimation of the process, which involves information of the value of interest after a time $t + \tau$ using the observed data over the interval $[0, t]$.

Given $t \in \{1, \ldots, T\}$, any inference made about $x_{0:t}$ will be based on the calculation of the posterior distribution $P\left(dx_{0:t}|y_{1:t}\right)$ [18]. To make a Bayesian sequential estimate on t, it must follow the recursive steps of prediction and update. These are given, respectively, by:

$$p\left(\dot{x}_{0:t}|y_{1:t-1}\right) = p\left(\dot{x}_{0:t-1}|y_{1:t-1}\right) p\left(\dot{x}_t|x_{0:t-1}, y_{1:t-1}\right) \tag{2.14}$$

$$p\left(\dot{x}_{0:t}|y_{1:t}\right) = \frac{g\left(y_t|x_{0:t}, y_{1:t-1}\right) p\left(\dot{x}_{0:t}|y_{1:t-1}\right)}{\int_{X^{(t+1)}} g_t\left(y_t|x_{0:t}, y_{1:t-1}\right) p\left(dx_{0:t}|y_{1:t-1}\right)} \tag{2.15}$$

From (2.14) and (2.15), it follows that:

$$p\left(\dot{x}_{0:t}|y_{1:t}\right) \propto \mu\left(\dot{x}_0\right) \prod_{i=1}^{t} g_i\left(y_i|x_{0:i}, y_{1:i-1}\right) p_i\left(\dot{x}_*|x_{0:i-1}, y_{1:i-1}\right) \tag{2.16}$$

where $\mu\left(\dot{x}_0\right)$ is the probability distribution of $\dot{x}_0$. This recursion may appear to be easy to compute, however $p\left(\dot{x}_{0:t} \mid y_{1:t}\right)$, $p\left(\dot{x}_t|y_{1:t}\right)$ are generally not in closed form, except for cases such as linear Gaussian state-space models and finite state-space Markov chains [18].

2.2 Functional Data Analysis

Functional data analysis (FDA) is one of the statistical approaches that deals with the analysis and modeling of data in the form of functions, as well as the conversion of more general data structures to a functional form. In this way, we express intermittently observed data at discrete time points and smooth them so that their resulting function measure represents an observation [19, 20]. More formally, we will represent data points as a function $x_i(t)$ where the i-th observation takes real values when evaluated over a time interval $t \in [0, T]$.

FDA seeks to extract as much information as possible from the properties of the functional structures in order to then apply concepts of multivariate statistical analysis in infinite-dimensional domains. Thus, by applying multivariate techniques, it is possible to recognize patterns of similarity or clustering in the data, represent high-dimensional data graphically [21], compute statistics such as the mean and variance of functions described as time series based on observable variables, and identify properties of functions, such as continuity and differentiability.

Ramsay and Dalzell [22], explain in further detail the practical reasons for considering functional data over traditional data:

- With the advancement of automation in data collection, functional observations are presented increasingly in applied studies, making them more accessible to researchers.
- There are several contexts in which reasoning in functional terms is more suitable than considering them as finite sets of observations.
- The analysis of functional data can be extended to other estimations, such as derivatives and functional operators.
- Properties of functions, such as continuity, can be significant in their further analysis.

Hence, the schematization of data as functional structures makes it possible to characterize systems by means of smooth dynamics [20], taking advantage of the information that can be obtained implicitly in functions and that is not accessible using traditional methods. One of the main uses of the aforementioned strengths is in the processing of noisy data. Considering that data obtained from real-world problems often include a noise component, which can vary from observation to observation, FDA approaches allow for effective noise reduction through smoothing curves, where the noise can be accommodated by repeating measurements of each observation [19] at non-regular sampling times, i.e. $t_{i+1} - t_i \neq t_{j+1} - t_j$ for some values of i and j. When data are handled as continuously observed functions, it can be assumed that there are no variable error rates, which makes them easier to analyze in methodologies such as stochastic processes [19] and also makes them subject to the application of the theorems for continuous functions.

2.3 Stochastic Processes

A stochastic process, also known as a random process, is one whose outputs are determined by randomness [23]. More formally:

Definition 2.3.1. (Stochastic Process) A stochastic process $\{X(t), t \in T\}$ is a collection of random variables $X(t)$ in which it is satisfied that:

1. $X_t(\omega) \in S$, where $S \subset \mathbb{R}$, known as space state.

2. X_t is indexed by a set of temporal parameters $t \in T$, $T \subset \mathbb{R}^+$, known as parameter space.

These models allow us to explain the structure and predict the future evolution of observed variables over time. Under the assumption that these processes are continuous, the parameter space will be based on a finite interval $T = [0, t]$ or $T = [0, \infty)$.

Then, we can characterize each random variable as X_t a mapping from the sample space Ω to values belonging to $\mathbb{R}$. Taking into account all possible finite dimensional distributions, we can generalize this concept to all the time values $t \in T$, such that the stochastic process $X(w)$ can be represented as:

$$X : \Omega \to \mathbb{R}^T$$
$$\omega \to X_t(\omega)$$

where a mapping is performed from Ω toward a set of real-valued functions indexed by the values of T. In this way, each sample $\omega \in \Omega$ will be associated with a function $X(\omega) \in \mathbb{R}$. $X(\omega)$ will be known as a realization or trajectory of the stochastic process.

The process $X(t) = \{X_t : t \in T\}$ evaluated at a time t has an average:

$$m_X(t) = \mathrm{E}[X(t)] \tag{2.17}$$

In addition, the autocovariance function at two points $(t_1, t_2) \in T^2$ ir written as:

$$\mathrm{C}_X(t_1, t_2) = \mathrm{E}\left(X(t_1) X(t_2)\right) - m_X(t_1) m_X(t_2) \tag{2.18}$$

Supposing that $t_1 = t_2 = t$, the equation (2.18) is called the variance of the stochastic process and is expressed by:

$$\mathrm{Var}(X(t)) = \mathrm{C}_X(t, t) \tag{2.19}$$

On the other hand, we can determine an autocorrelation function by using (2.17) and (2.18), such that:

$$R_X(t_1, t_2) = \mathrm{E}\left(X(t_1) X(t_2)\right) \tag{2.20}$$

Replacing (2.20) in (2.18), it follows that:

$$\mathrm{C}_X(t_1, t_2) = R_X(t_1, t_2) - m_X(t_1) m_X(t_2) \tag{2.21}$$

If a finite time set $\{0 \leq t_1 < t_2 \ldots < t_k\}$ is established and the random variable X is evaluated at each element of it, we will have a random vector of the form $\mathbf{X} = (x(t_1), x(t_2), \ldots, x(t_k))$, to which we can assign a distribution function as well as a mass probability function.

Definition 2.3.2. *(Stochastic Process Distribution and Mass Functions)* Let $\{X(t); t \in T\}$ a stochastic process of order k. The joint distribution function of X is given by:

$$F(x_1, \ldots, x_k; t_1, \ldots, t_k) = p\Big(\{X(t_1) \leq x_1, \ldots, X(t_k) \leq x_k\}\Big) \tag{2.22}$$

On the other hand, the mass probability function of X is:

$$p(x_1, \ldots, x_k; n_1, \ldots, n_k) = p\left\{\Big(x_{n_1} = x_1, \ldots, x_{n_k} = x_k\Big)\right\} \tag{2.23}$$

Under the assumption that $F(x_1, \ldots, x_k; t_1, \ldots, t_k) \in C^1(\mathbb{R})$, we have that:

$$f(x_1, \ldots, x_k; t_1, \ldots, t_k) = \frac{\partial^k}{\partial x_1, \ldots, \partial x_k} F(x_1, \ldots, x_k; t_1, \ldots, t_k) \tag{2.24}$$

2.3.1 Classification of Stochastic Processes

There are several classes of stochastic processes, defined by the behavior of their states and by the discrete or continuous nature of each one. In this work, we will focus on the following:

Markovian Process

The transition between states can be viewed as a process in which the evolution of the future state depends on a known present state and in turn is independent of the past. This characterization is known as a Markov chain.

Definition 2.3.3. *(Markov Chain)* Let S a discrete set with $i, j \in S$, a Markov chain is said to be a sequence of $X_0, \ldots, X_n$ of n random variables that satisfies:

$$P(X_{n+1} = j | X_0 = x_0, \ldots, X_{n-1} = x_{n-1}, X_n = i) = P(X_{n+1} = j \mid X_n = i) \tag{2.25}$$

According to the previous concept, we can also describe Markovian processes.

Definition 2.3.4. *(Markovian Processes)* These are the processes that meet Markov's property, i.e.:

$$P\left(X_{n+1} = x_{n+1} \mid X_0 = x_0, \ldots, X_n = x_n\right) = P\left(X_{n+1} = x_{n+1} \mid X_n = x_n\right) \tag{2.26}$$

When (2.25) depends only on the indexes i and j, Markov chains satisfy the homogeneity property. Thus, the probability described in Theorem 2.3.3 can be rewritten as:

$$P\left(X_{n+1} = j | X_n = i\right) = P\left(X_1 = j | X_0 = i\right) \tag{2.27}$$

Assuming homogeneity in a Markov process, we can represent the probabilities of a stepwise transition from one state to another by means of a matrix of the form $\boldsymbol{P} = (P_{ij}) = P\left(X_1 = j \mid X_0 = i\right)$. This matrix is known as the transition matrix, and it must have the characteristics of a stochastic matrix [23].

Definition 2.3.5. *(Stochastic Matrix)* A stochastic matrix is any square matrix P, with $P_{ij} \geq 0$ for all i, j, which satisfies the condition:

$$\sum_j P_{ij} = 1 \tag{2.28}$$

Gaussian Processes

Gaussian processes are one of the most widely reported stochastic processes in the literature due to their particular importance in the modeling of nonlinear systems with experimental data [24]. This process is based on multivariate normal distribution functions.

Definition 2.3.6. *(Multivariate Normal Distribution)* Let $X = (X_1, \ldots, X_n)$ a random vector of dimension n. It follows a multivariate normal distribution, denoted as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if for some $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$ every linear combination $Y = a_1 X_1 + \ldots + a_k X_k$ is distributed as a univariate normal.

The joint density of a multivariate normal distribution is given by:

$$f\left(x_1, \ldots, x_k\right) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \tag{2.29}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and the covariance matrix associated with X, respectively.

In Gaussian processes, the above concepts are extended to high-dimensional spaces [23], and even to non-finite dimensions.

Definition 2.3.7. *(Gaussian Processes)* Let $X(t)$ be a stochastic process. It is defined as a Gaussian process if for $t_1, \ldots, t_k$, all dimensional distributions $\left(X_{t_1}, \ldots, X_{t_k}\right)$ are finite and normally distributed, i.e., they are governed by the probability distribution:

$$p_X\left(X_1, \ldots, X_k, t_1, \ldots, t_k\right) = \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{|K_k|^{-\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu_k)^T K_k^{-1}(X-\mu_k)} \tag{2.30}$$

where $\mu_k = \left(m_X\left(t_1\right), \ldots, m_X\left(t_k\right)\right)^T$ and K_k has the characteristics of a covariance matrix.

The methodology presented in this work will be based on the use of Gaussian processes, so this topic will be discussed in more detail in the next section.

Gaussian-Markovian Processes

Under certain conditions, Gaussian processes can mix with Markov processes, giving rise to Gaussian Markovian processes.

Definition 2.3.8. *(Gaussian-Markovian Processes)* Let $\{\tilde{x}_t, t \geq 0\}$ a real-valued Gaussian process, y $s, t \geq 0$. For every s and t, $\{\tilde{x}_t\}$ is a Gaussian-Markovian process if it also satisfies the Markov property, so that:

$$\left(\tilde{x}_{t+s} \mid \tilde{x}_u\right) \sim \left(\tilde{x}_{t+s} \mid \tilde{x}_t\right) \tag{2.31}$$

Fixing values for the mean $\tilde{\mu}_t$ and for the covariance function $\tilde{K}_{s,t}$, values can be generated for $(x_1, \ldots, x_n) = (\tilde{x}_{t_1}, \ldots, \tilde{x}_{t_n})$ justified by the properties of the conditional distributions of the multivariate normal distribution [23]. For $i \in \{1, \ldots, n\}$, and given μ_i y $K_{i,i}$ as the expected value and the variance of $\{x_t, t \geq 0\}$ respectively, it follows that:

$$\begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_i \\ \mu_{i+1} \end{pmatrix}, \begin{pmatrix} K_{i,i} & K(i, i+1) \\ K_{i,i+1} & K(i+1, i+1) \end{pmatrix}\right) \tag{2.32}$$

Applying the properties of conditional distributions, it follows that:

$$\left(x_{i+1} \mid x_i = x\right) \sim N\left(\mu_{i+1} + \frac{K_{i+1,i}}{K_{i,i}}\left(x - \mu_i\right); K_{i+1,i+1} - \frac{K_{i+1,i}^2}{K_{i,i}}\right) \tag{2.33}$$

2.4 Gaussian Processes

Bayesian predictive models have the particularity that the target values are closely related to the scalar product of the data vectors [25]. Within these, Gaussian processes use this relationship to constitute a learning and prediction method that does not specifically correspond to any parametric scheme, allowing in particular the combination between functions and their derivatives [24] in a direct and understandable way.

More precisely, a Gaussian process is seen as the generalization of infinite normally distributed variables, where this set of variables is considered random by definition [26]. Considering the existence of noise conditions, it is possible to use these variables to compute conditional normal distributions that estimate the points where the value of the functions to be modeled is uncertain. In this way, the marginalization of all non-observable points in any of the dimensions turns the infinite-dimensional Gaussian processes into a finite set of multivariate normal distributions [27].

Let d be the cardinality of the dimension, $\boldsymbol{\mu}$ a vector of means and $\boldsymbol{\Sigma}$ a covariance matrix of a normal distribution, a Gaussian process can be expressed as the probability of a random function in a d-dimensional space:

$$p(f(\boldsymbol{x})) = \mathcal{GP}\left(\mu(\boldsymbol{x}), k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right) \tag{2.34}$$

where the mean $\mu(\boldsymbol{x})$ and covariance $k\left(\boldsymbol{x}, \boldsymbol{x}'\right)$ parameters are also real functions.

Using this definition, we can directly write $f(\mathbf{x})$ as a random variable of the form:

$$f(\mathbf{x}) \sim \mathcal{G}P\left(\mu(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right) \tag{2.35}$$

The above distribution can be represented in a set of indices $i \in \{1, 2, \ldots, n\}$ by components as $\left(f\left(\mathbf{x}_1\right)_1 f\left(\mathbf{x}_2\right)_1 \ldots, f\left(\mathbf{x}_n\right)\right) \sim \mathrm{N}(\mu, \mathrm{K})$, where $\mu_i = \mu\left(\mathbf{x}_i\right)$ and $\mathrm{K}_{i,j} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$. Due to the fact that the distributions are dimensionally finite, the behavior of $f(\mathbf{x})$ can be described only in terms of the mean and covariance functions [14], since these represent the dependence between each of the evaluations of the observations. Once these have been determined, Gaussian process can be used to sample values from the prior and conditional posterior distributions based on past observations [28].

The notation given in (2.34) is known as kernel formalism; however, there are other approaches from which GPs can be conceptualized. While Gaussian processes are classically taken as a probability functional distribution describing data points, Solin [29] states that they can be viewed as signal processing, as well as random fields when dealing with dimensions higher than two. Thus, multivariate normal distributions deal with a fixed number of elements in their input vector, while a Gaussian process handles a data structure that may whose cardinality may progressively tend to infinity. In particular, this object is in a Hilbert space of infinite dimension $\mathcal{J}\left(\mathrm{E}^{d}\right)$ [27], and goes from being indexed by a finite set to become a variable of continuous values, which keeps the notation x.

The advantage of this transformation is that despite the fact that the probabilistic model considers an input of continuous values, the processing, inference and learning of the data can be performed with the same constraints of a numerable set of inputs [27] without requiring a number of dimensions to be fixed. For this reason, Gaussian processes are generally regarded in the literature as nonparametric.

2.4.1 Covariance Functions and Kernels

As noticed in the previous section, covariance functions are a fundamental part of the inference process of a Gaussian process. In the context of this work, we will use the definition and describe the required properties of multi-output covariance functions, since in the computational setup we deal with multidimensional arrays.

Definition 2.4.1. *(Multi-output Covariance Function)* Let $k\left(\boldsymbol{x}, \boldsymbol{x}'\right) : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}^{D \times D}$ be a multidimensional function. It is called a multi-output covariance function when each (d, d') entry of the output matrix meets the characteristics of a scalar covariance function with inputs $(\boldsymbol{x}, d)$ and $(\boldsymbol{x}', d')$.

Starting at this point, we will assume that in the context of functional estimation, the definition 2.4.1 shares the same mathematical properties as that of kernel, so we can interpret them as a measure of similarity between inputs [30]. Due to the high dependence of the kernels on the inputs and the hyperparameters of the model, it is required to impose certain restrictions and properties in accordance with the regression problem addressed in this work.

Gibbs [31], determines two main properties of covariance functions:

- The correlation between two data points that are neighbors in the input space must be greater than that of two distant points.
- The possibility of data being perturbed by noise must be considered, so noise must be included in the covariance function.

Before detailing the theoretical basis of the square exponential covariance, which is the kernel used in this research work, it is necessary to define one of its main properties, the stationarity.

Definition 2.4.2. (Stationary Covariance Functions) A kernel $k(x, x')$ is said to be stationary if it depends only on the radial distance of the input data, i.e.:

$$k\left(\mathbf{x}, \mathbf{x}'\right) = k\left(\mathbf{x} - \mathbf{x}'\right) \tag{2.36}$$

We also define (2.36) as translation invariant.

When a covariance function is stationary and depends on a distance function described by a suitable norm $\|.\|$, we extend the above-mentioned definition to stationary isotropic or rotation-invariant kernel.

Squared Exponential Covariance Function

The Squared Exponential (SE) covariance function is one of the most widely used and well-known stationary kernels.

Definition 2.4.3. *(Squared Exponential Covariance Function)* Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^D$. Then, the squared exponential function is defined by:

$$k\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \alpha^2 \exp\left(-\frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{x}'\right)^\top \boldsymbol{\Lambda}^{-1}\left(\boldsymbol{x} - \boldsymbol{x}'\right)\right) \tag{2.37}$$

where α^2 is the variance coefficient and $\Lambda = \operatorname{diag}\left(\lambda_1^2, \ldots, \lambda_D^2\right)$ is the length scale matrix and controls the width.

Looking at (2.37) carefully, we can find that it is similar to the normal distribution formula, thus it is also referred as Gaussian kernel [27]. In fact, it is possible to rewrite (2.37) by

considering the following class of functions:

$$\begin{aligned} f(x) &= \sum_{u \in \mathrm{U}} w_u \exp\left(-(x-u)^2\right) \\ &= \sum_{u \in \mathrm{U}} w_u \phi_u(x) \end{aligned} \tag{2.38}$$

which is the class of sums of basis functions $\phi_u(x)$ with finite number of centres $u \in \mathbb{U} \subset \mathbb{R}$ and is scaled by weights distributed with a Gaussian prior $w \sim \mathcal{N}\left(0, \alpha^2\right)$ [26]. The covariance of this class is defined by:

$$\begin{aligned} k\left(x, x'\right) &= \mathrm{E}\left[f(x) f\left(x'\right)\right] \\ &= \alpha^2 \sum_{u \in \mathrm{U}} \phi_u(x) \phi_u\left(x'\right) \end{aligned} \tag{2.39}$$

By approximating a number tending to infinity of bases in $\mathbb{R}$, the sum of the basis functions can be expressed as an integral of the form:

$$\alpha^2 \int_{-\infty}^{\infty} \phi_u(x) \phi_u\left(x'\right) \mathrm{d}u \propto \alpha^2 \exp\left(-\frac{1}{2}\left(x - x'\right)^2\right) \tag{2.40}$$

Since this particular covariance is defined as a stationary function, we can claim that it is equivalent to a regression based on an infinite numerable set of Gaussian basis functions [30].

2.5 Stochastic Differential Equations

Ordinary differential equations are a useful tool in the modeling of diverse systems. However, they do not consider the random components that could be present in the variables. To characterize this behavior, ODEs are modified by adding stochasticity terms in order to obtain solutions that also present stochastic characteristics.

Before conceptualizing stochastic differential equations, it is necessary to define the integral and Ito's lemma, which are fundamental in the calculation of derivatives of a stochastic process [32] and, consequently, in the resolution of SDEs.

Definition 2.5.1. *(Ito's Integral)* Let X_t be a diffusion process, and f_t be a stochastic process adapted to a filtration F. Then, the Ito integral is defined as:

$$Y_t = \int_0^t f_s dX_s \tag{2.41}$$

Theorem 2.5.2. *(Ito's Lemma) Let $W(t)$ be a Brownian motion, and $X(t)$ be an Ito process, and $f \in \mathrm{C}^2$. Then, for $t \geq 0$:*

$$df(X_t) = \left(\mu_t \frac{\partial f}{\partial x}(X_t) + \frac{1}{2}\sigma_t^2 \frac{\partial^2 f}{\partial x^2}(X_t)\right) dt + \sigma_t \frac{\partial f}{\partial x}(X_t) dW_t \tag{2.42}$$

Finally, using the above definition and theorem, we can state the concept of SDE in a formal way.

Definition 2.5.3. *(Stochastic Differential Equation)* Let X_t be a continuous stochastic process and W_t be a Brownian motion. If linear combinations of small perturbations in W_t and in t can compose small changes in X_t, we can define a stochastic differential equation by:

$$dX_t = a(t, W_t, X_t) + b(t, W_t, X_t) dW_t \tag{2.43}$$

which can be rewritten as:

$$X_t = \int_0^t a(s, W_s, X_s) ds + \int_0^t b(s, W_s, X_s) dW_s + X_0 \tag{2.44}$$

2.6 The Forward Problem

The propagation of uncertainty within a probabilistic model is known as the forward problem, and its proper implementation is critical for increasing the accuracy of the solutions that reconstruct the inverse problem. In order to achieve this, it is necessary to perform the calibration of the problem, which consists of quantifying the magnitude and structure of the sampling error rates [2] generated when estimating new inputs from the observable states.

2.6.1 Uncertainty

Uncertainty quantification involves the analysis and choice of mathematical models that represent the uncertainty of an unspecified variable [33], depending on the type and amount of information available on the quantity of interest. It may be implicit within the model, or come from external sources such as prior knowledge of experts, or experimental data from similar studies; and must be incorporated using appropriate statistical methods.

The study of the influence of errors within data structures is of special interest in the area of complex systems, and particularly in those that despite showing quantitative relationships

between inputs and predictions, present a high variability behavior when modifying the parameters that control the equations [34]. In these systems, uncertainty is usually categorized as a random variable, so estimation methods are found in a wide spectrum ranging from probability theory to scientific computing. According to Elishakoff [35], uncertainty quantification is approached mainly from three approaches: probabilistic, possibilistic and anti-optimization, and is usually implemented prior to the simulations and not as an ulterior argument.

2.7 Polynomial Chaos Preliminaries

Polynomial Chaos approaches were introduced as an alternative to modeling techniques of stochastic process with normally distributed variables, and is considered as a subset of the class of polynomial approximations [34]. This set of techniques consists of projecting the model outcomes as a basis of orthonormal polynomials of the observation data [36], which allows estimating the variability of the results and their dependence on the entries.

Let Y be a probabilistic model, and $\hat{Y}$ its polynomial expansion, we want to find:

$$\begin{aligned} Y &\approx \hat{Y}(\boldsymbol{x}, t, \boldsymbol{Q}) \\ &= \sum_{n=0}^{N_p-1} c_n(\boldsymbol{x}, t)\phi_n(\boldsymbol{Q}) \end{aligned} \tag{2.45}$$

where c_n are the coefficients of the expansion, ϕ_n the polynomials and N_p the cardinality of the expansion at order p.

To formulate a polynomial expansion it is necessary to determine the polynomials $\phi_n(\boldsymbol{Q})$, which must be orthogonal with respect to a probability density [34]; and then calculate the coefficients c_n by intrusive or non-intrusive methods.

2.7.1 Basis and Polynomials

As the entire methodology is carried out in a Hilbert space H, we will assume that the bases used in the polynomial expansions will be Hilbert bases, also known as orthonormal basis.

Definition 2.7.1. *(Orthonormal Basis)* Let V be a inner product space with an associated inner product $\langle \cdot, \cdot \rangle$, and $(e_1, \ldots, e_n) \in V$ a list of non-zero vectors. For all $i, j \in \{, \ldots, n\}$,

$(e_1, \ldots, e_n)$ is said to be orthonormal if:

$$\left\langle e_i, e_j \right\rangle = \delta_{ij} \tag{2.46}$$

where δ_{ij} is the Kronecker delta function.

Based on the fact that an orthonormal basis of finite dimension is also orthogonal, and that orthonormal bases must be formed by a list of orthonormal vectors, the family of polynomials that we aim to study is orthogonal.

Definition 2.7.2. *(Orthogonal Polynomials)* Let $\{p_n(x)\}_{n=0}^{\infty}$ be a sequence of polynomials of order n, and $w(x)$ a continuous positive weight function. It is said to be orthogonal with respect to $w(x)$ on (a, b) if:

$$\int_a^b w(x)p_m(x)p_n(x)dx = h_n\delta_{mn} \tag{2.47}$$

where δ_{mn} is the Kronecker delta function.

Using the above definition, an inner product between two orthogonal polynomials f and g can be defined, which is given as:

$$\langle f, g\rangle := \int_a^b w(x)f(x)g(x)dx \tag{2.48}$$

According to Andrews [37], the most frequently used orthogonal polynomials are known as classical polynomials, and are described in the following table:

Polynomial	$p_n(x)$	$w(x)$	(a, b)
Hermite	$H_n(x)$	e^{-x^2}	$(-\infty, \infty)$
Jacobi	$P_n^{(\alpha,\beta)}(x)$	$(1-x)^\alpha(1+x)^\beta$	$(-1, 1)$
Laguerre	$L_n^{(\alpha)}(x)$	$e^{-x}x^\alpha$	$(0, \infty)$
Legendre	$P_n(x)$	1	$(-1, 1)$

Table 2.1: Classical orthogonal polynomials.

The representation of stochastic processes constituted as non-finite linear combinations of orthogonal polynomials, and their parallelism with the Fourier series representations over bounded intervals is established by the Karhunen-Loève theorem.

Theorem 2.7.3. *(Karhunen–Loève) Let $X_t \in (\Omega, F, \mathbf{P})$ be a zero-mean square-integrable stochastic process indexed over $[a, b]$, and $K_X(s, t)$ be its covariance function. Then X_t has a L^2 convergent representation:*

$$X_i = \sum_{k=1}^{\infty} Z_k e_k(t) \tag{2.49}$$

where:

$$Z_k = \int_a^b X_t e_k(t) dt \tag{2.50}$$

is a random variables Z_k with zero-mean, and that is uncorrelated point-wise.

2.7.2 Numerical Integration

Numerical integration is one of the most frequently used numerical methods in stochastic computations, since many systems do not have closed expressions of its solutions. For polynomial estimation, the technique employed is Gaussian quadrature, which integrates polynomials of up to degree $2n-1$ [38] into symmetric intervals using the evaluations of a function at a given number of interior points and a weights function.

Let $f(x)$ be some well-behaved function, and $w(x)$ be a weight function. Then, the integral I is represented in terms of a weighted sum by:

$$\begin{aligned} I &= \int_{-1}^{1} f(x) dx \\ &= \sum_{i=0}^{n-1} w_i f(x_i) \end{aligned} \tag{2.51}$$

The Gauss Legendre quadrature is considered as the standard case, but there exist other types of quadratures on orthogonal polynomials that are subject to restrictions particular to each of these. Following the notation above, we can generalize (2.51) as:

$$\begin{aligned} I &= \int_{-1}^{1} f(x) dx \\ &= \int_{-1}^{1} w(x) g(x) dx \end{aligned} \tag{2.52}$$

where $f(x)$ results from the convolution of $w(x)$ and $g(x)$. Then, under the conditions of each polynomial, we can approximate (2.52) by:

$$\begin{aligned} I &= \int_{-1}^{1} W(x) g(x) dx \\ &\approx \sum_{i=0}^{n-1} w_i g(x_i) \end{aligned} \tag{2.53}$$

In the table below, Venkateshan [38] summarizes the Gaussian quadrature rules for each of the classical orthogonal polynomials:

Quadrature	**Integral**
Gauss-Hermite	$\int_{-\infty}^{\infty} \mathrm{e}^{-x^2} g(x)dx$
Gauss-Jacobi	$\int_{-1}^{1} f(x)(1-x)^{\alpha}(1+x)^{\beta}dx$
Gauss-Laguerre	$\int_{0}^{\infty} \mathrm{e}^{-x} g(x)dx$
Gauss-Legendre	$\int_{-1}^{1} g(x)dx$

Table 2.2: Gauss quadrature rules.

Chapter 3

State of the Art

Arising from the need to understand and ensure accuracy in problem solving, the mathematical development of the theory surrounding uncertainty and its quantification dates back to the late nineteenth century, addressing the theory of probabilities to replace unknown values in statistical mechanics [39]. Since 1960, works such as those of Dempster [40] and Feller [41] show Bayesian inference as a new approach to uncertainty assessment, inferring through prior knowledge the difference between measurable and desired data. Authors such as O'Hagan [42] and Neal [43] suggested materializing the Bayesian formalism through Gaussian processes [44]. On the other hand, Skilling [45], Glimm [46] and Eriksson [47] introduced Bayesian methods as a tool for the reconstruction of numerical solutions of systems of ordinary differential equations with boundary conditions by setting them as an inference problem, while Herzog [48] deepened into the influence of the correlation between parameters in second-order models.

Around the early 2000s, the rapid advancement of scientific computing led to the combination of estimation with Gaussian processes and Bayesian characterization of differential equations, giving rise to Bayesian Data Analysis, where system responses were analyzed and evaluated by means of computational simulations [49]. In the Bayesian computation area, Vanhatalo [50] implemented the algorithms of several techniques such as Monte Carlo Markov Chains (MCMC) methods, covariance kernel approximation and model assessment tools for Gaussian processes [11]. On the other hand, Vyshemirsky and Girolami [51] did one of the first works in Bayesian estimation for ODE systems by assigning a prior distribution θ to generate samples of a posterior and thus approximate its solutions, while Adams [52] approached the problem

from Gaussian processes, conditioning the domain and using the rejection sampling method to avoid intractability by approximating the likelihood normalization term, and easing the solvability of the system [11]. Solin and Sarkka [53] followed along this path by transforming GP regression into finite-dimensional state-space models and improve Bayesian filtering inference, while Sternberg [14] extended their work to high-dimensional inputs through adaptive covariance functions. However, large-scale studies were still limited in numerous cases due to the expensive computational cost of performing multiple runs to evaluate complex systems. In 2013, Chkrebtii [2] proposed an algorithm of reduced computational complexity using derivative inference on a discretized temporal mesh, which was subsequently rewritten by [5] in a tensor-adaptive version. In addition, authors such as Dutta [49], Rendardy [54] and Oparaji [55] studied an alternative to Chkrebtii's work using surrogate models, such as the Radial Basis Function, Support Vector Regression (SVR), and Polynomial Chaos Expansion (PCE); with lower computational cost. Modeling with PCE gained popularity as it was recognized as one of the most efficient in complex systems, both in its intrusive and non-intrusive approaches [49]. This claim was supported by the studies of Sepahvand [33], who made a general exploration of the quantification of uncertainty with this method and its convergence criteria for parameter estimation; and subsequently of Sudret [56], who studied in more depth its application in stochastic analysis.

As could be seen from the evolution of the previously mentioned studies, functional estimation and uncertainty quantification techniques have evolved drastically over the last few decades. With the advancement in data processing, a denser sampling of observations can be performed on different continuous measures such as time and space, as shown by Ullah [20]. These improvements have led to the using of these methods into the artificial intelligence framework, where Li [57] and Khosravi [58] use it to estimate confidence levels in the prediction intervals of neural networks for forecasting tasks. These studies were complemented by Yao [59], who implemented it in Bayesian neural networks. In addition, functional estimation based on GP regression have been studied, as proposed by Lee, as well as Matthews [60] and Huang [61] studied the equivalence between GPs and single-layer deep learning neural networks for prediction tasks by using covariance functions as kernels. Also, Kabir [62, 63] developed an artificial neural network for uncertainty quantification, while Tripathy [64] complemented this work to higher dimensionalities by implementing a surrogate model based on deep learning.

Chapter 4

Methodology

4.1 Gaussian Process Priors

Gaussian processes define a distribution over f functions that map some space of $\mathcal{X}$ entries to $\mathbb{R}$, that is, $f : \mathcal{X} \to \mathbb{R}$. Formally, for some finite set of elements taken from $\mathcal{X}$, f is a GP defined by:

$$f(x) \sim GP\left(m(x), k\left(x, x'\right)\right)$$

The admissible functions for $\cdot(\cdot)$ and $k(\cdot, \cdot)$ satisfy the condition that their marginals are distributed as Gaussian, where $m(\cdot)$ is a parametric function and $k(\cdot, \cdot)$ is a function that admits a positive semi-definite matrix when evaluated at the points $x \in \mathcal{X}$. The mean and covariance functions map the set of indices to the real numbers, so that $m : \mathcal{X} \to \mathbb{R}$ and $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Let x_{new} be a matrix where each row forms a finite sequence of new input points x_i^{new} for $i \in \{1, \ldots, n\}$. Then, the covariance matrix can be estimated as:

$$k\left(x_{new}, x_{new}\right) = \begin{pmatrix} k\left(x_1^{new}, x_1^{new}\right) & \ldots & k\left(x_1^{new}, x_n^{new}\right) \\ k\left(x_2^{new}, x_1^{new}\right) & \ldots & k\left(x_2^{new}, x_n^{new}\right) \\ \vdots & \vdots & \vdots \\ k\left(x_n^{new}, x_1^{new}\right) & \ldots & k\left(x_n^{new}, x_n^{new}\right) \end{pmatrix}$$

The function $k\left(x, x'\right)$ models the dependence between the functional values at different input points x and x'. An usual way to choose $k\left(x, x'\right)$ is to consider a radial basis function, defined

by:

$$k\left(x, x'\right) = \sigma_f^2 \exp\left(-\frac{\|x - x'\|}{2\lambda^2}\right) \tag{4.1}$$

where (4.1) provides a transition kernel that serves to model smooth and stationary functions, λ is a scaling parameter, and σ_f^2 is the variance of the signal. Given $m(x) = 0$, it is possible to sample values of f for x_{new} using a Gaussian process:

$$f_{new}(x_{new}) \sim N\left(0, K\left(x_{new}, x_{new}\right)\right)$$

Defining f_{new}, it follows that:

$$\begin{aligned} f_{new} &= f_{new}(x_{new}) \\ &= \left(f\left(x_1^{new}\right), \ldots, f\left(x_n^{new}\right)\right)^T \end{aligned} \tag{4.2}$$

Consider a set of observations $\mathcal{D}_t = \{x_t, y_t\}$ to make predictions using new data x_{new}, we take $f_{new}(x_{new})$ from the posterior distribution $p\left(f_{new}(x_{new})|\mathcal{D}_t\right)$. The observations $y_t = f(x_t)$ and the function $f_{new}(x_{new})$ follow a multivariate joint normal distribution:

$$\begin{pmatrix} f \\ f_{new} \end{pmatrix} \sim N\left(\begin{pmatrix} m\left(x\right) \\ m(x_{new}) \end{pmatrix}; \begin{pmatrix} K\left(x, x\right) & K\left(x, x_{new}\right) \\ K\left(x_{new}, x\right) & K\left(x_{new}, x_{new}\right) \end{pmatrix}\right) \tag{4.3}$$

with $K\left(x, x\right)$ as the kernel evaluated in x. The marginal distribution $f_{new}|f, x, y$ is still Gaussian, so:

$$f_{new}|f, x, y \sim N\left(m\left(x_{new}, \right)^{post}, K^{post}\left(f(x_{new}), f(x_{new})\right)\right)$$

where:

$$m^{post}\left(x_{new}\right) = m\left(x_{new}\right) + K\left(x_{new}, x\right) K^{-1}\left(x, x\right)\left[f - m\left(x\right)\right] \tag{4.4}$$

$$K^{post}\left(f(x_{new}), f(x_{new})\right) = K\left(x_{new}, x_{new}\right) - K\left(x_{new}, x\right) K^{-1}\left(x, x\right) K\left(x, x_{new}\right) \tag{4.5}$$

Making a generalization, by observing the real values of f with additive noise we obtain the following model:

$$\begin{aligned} y_i &= f(x_i) + \epsilon_i \\ \epsilon_i &\sim N\left(0, \sigma_\epsilon^2\right) \\ y|f &\sim N\left(f(x_i), \sigma_\epsilon^2 I\right) \end{aligned} \tag{4.6}$$

where I represents the identity matrix. Noise can be included in the covariance function by rewriting it as follows:

$$k\left(f(x_i), f(x_j)\right) = k\left(x_i, x_j\right) + \delta_{ij}\sigma_\epsilon^2 \tag{4.7}$$

where δ_{ij} is the Kronecker delta function. Uncertainty is now present in the observations, so the joint distribution over the unknown data and the known data is augmented in the covariance equation by the following expression:

$$\begin{pmatrix} f \\ f_{new} \end{pmatrix} \sim N\left(\begin{pmatrix} m\left(x\right) \\ m(x_{new}) \end{pmatrix}; \begin{pmatrix} K\left(x,x\right) + \sigma_\epsilon^2 I & K\left(x, x_{new}\right) \\ K\left(x_{new}, x\right) & K\left(x_{new}, x_{new}\right) + \sigma_\epsilon^2 I \end{pmatrix}\right)$$

Thus, we can rewrite the expression for f_{new} given above as a conditional:

$$f_{new}|f, x, y \sim N\left(m\left(x_{new},\right)^{post}, K^{post}\left(f(x_{new}), f(x_{new})\right)\right)$$

where:

$$m^{post}\left(x_{new}\right) = m\left(x_{new}\right) + K\left(x_{new}, x\right)\left(K\left(x,x\right) + \sigma_\epsilon^2 I\right)^{-1}\left[f - m\left(x\right)\right] \tag{4.8}$$

$$K^{post}\left(f(x_{new}), f(x_{new})\right) = K\left(x_{new}, x_{new}\right) - K\left(x_{new}, x\right)\left(K\left(x,x\right) + \sigma_\epsilon^2 I\right)^{-1} K\left(x, x_{new}\right) + \sigma_\epsilon^2 I \tag{4.9}$$

4.1.1 Derivative Gaussian process

The flexibility of Gaussian processes allows modeling derivatives of functional data [24], being of major importance in engineering applications, in dynamical systems, and in modeling the solutions of systems of differential equations.

Differentiation is a linear operator, so the derivative of a GP is still a GP [30]. Therefore, a Gaussian process can be used to predict about derivatives, and also use derivative observations to make predictions. The mean of the derivative is equal to the derivative of the mean of the latent process, which means that:

$$\mathrm{E}\left[\frac{\partial f(x)}{\partial x_d}\right] = \frac{\partial\, \mathrm{E}\, f(x)}{\partial x_d} \tag{4.10}$$

The derivatives of the function covariance involve cross-covariance functions between derivatives. For indices $i \in \{1, 2, \ldots, K\}$ and $j \in \{1, 2, \ldots, d\}$, each of the covariance matrices is

expressed as:

$$\text{Cov}\left[f(x_i), \frac{\partial f(x_j)}{\partial x_{j,d}}\right] = \frac{\partial k\left(x_i, x_j\right)}{\partial x_{j,d}} \tag{4.11}$$

$$\text{Cov}\left[\frac{\partial f(x_j)}{\partial x_{j,d}}, f(x_i)\right] = \frac{\partial k\left(x_j, x_i\right)}{\partial x_{i,d}} \tag{4.12}$$

$$\text{Cov}\left[\frac{\partial f(x_i)}{\partial x_{i,k}}, \frac{\partial f(x_j)}{\partial x_{j,d}}\right] = \frac{\partial^2 k\left(x_i, x_j\right)}{\partial x_{j,k}\partial x_{j,d}} \tag{4.13}$$

where $x_{i,k}$ denotes the k-th element of x_i. This result allow us to define a prior distribution over the derivative GP in terms of the GP prior, i.e.:

$$f \sim GP\left(m(.), k_{(f,f)}(.,.)\right)$$
$$\nabla f \sim GP\left(m'(.), k'(.,.)\right)$$

The joint distribution of the GP with the addition of uncertainty and its derived process is given by:

$$\begin{pmatrix} f \\ \nabla f \end{pmatrix} \sim GP\left[\begin{pmatrix} m(.) \\ m'(.) \end{pmatrix}, \begin{pmatrix} k\left(x_i, x_j\right) & \frac{\partial k\left(x_i, x_j\right)}{\partial x_{j,d}} \\ \frac{\partial k\left(x_j, x_i\right)}{\partial x_{i,d}} & \frac{\partial^2 k\left(x_i, x_j\right)}{\partial x_{i,k}\partial x_{j,d}} \end{pmatrix}\right]$$

To predict new observed values $y_* = f_*(x_*) + \epsilon$, $\epsilon \sim N\left(0, \sigma_\epsilon^2\right)$, the mean of the derivative of the function $f_*(x_*) = \left(f_*(x_{*,1}), \ldots, f_*(x_{*,d})\right)^T$ with respect to dimension d is:

$$\begin{aligned} \text{E}\left[\frac{\partial f_*}{\partial x_{*,d}}\right] &= \frac{\partial \text{E}\left(f_*\right)}{\partial x_{*,d}} \\ &= \frac{\partial m'\left(x_*\right)}{\partial x_{*,d}} \\ &= \frac{\partial k\left(x_*, x\right)}{\partial x_{*,d}} \times \left[k\left(x, x\right) + \sigma_\epsilon^2 I\right]^{-1} y \end{aligned} \tag{4.14}$$

The calculation of the variance is part of the work of Riihimä and Vehtari [65], and is given as follows:

$$\text{Var}\left[\frac{\partial f_*}{\partial x_{*,d}}\right] = \frac{\partial^2 k\left(x_*, x_*\right)}{\partial x_{*,d}\partial x_{*,d}} - \frac{\partial k\left(x_*, x\right)}{\partial x_{*,d}} \times \left[k\left(x, x\right) + \sigma_\epsilon^2 I\right]^{-1} \times \frac{\partial k\left(x, x_*\right)}{\partial x_{*,d}} \tag{4.15}$$

where:

$$x = \left(x_1, \ldots, x_d\right)^T \tag{4.16}$$

$$x_* = \left(x_{*,1}, \ldots, x_{*,d}\right)^T \tag{4.17}$$

$$k\left(x_*, x\right) = \left(k\left(x_{*,d}, x_1\right), \ldots, k\left(x_{*,d}, x_d\right)^T\right) \tag{4.18}$$

The posterior distribution of the derivative process of the observations can be written as:

$$\frac{\partial f_*}{\partial x_{*,d}} \sim GP\left(\mathrm{E}\left[\frac{\partial f_*}{\partial x_{*,d}}\right], \mathrm{Var}\left[\frac{\partial f_*}{\partial x_{*,d}}\right]\right)$$

Considering the particular case in which the covariance is of exponential form, we have that:

$$\begin{aligned}
\mathrm{Cov}\left(f(x_i), f(x_j)\right) &= k\left(x_i, x_j\right) \\
&= \sigma_f^2 \exp\left(-\frac{1}{2}\frac{\|x^{(i)} - x^{(j)}\|^2}{\lambda^2}\right) \\
&= \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{1}{\lambda_d^2}\left(x_d^{(i)} - x_d^{(j)}\right)^2\right)
\end{aligned} \tag{4.19}$$

where σ_f^2 and $\lambda = (\lambda_1, \ldots, \lambda_d)$ are the hyperparameters of the model. The derivatives of the observations are:

$$\frac{\partial}{\partial x_d^{(i)}} k\left(x_i, x_j\right) = \sigma_f^2 \times \left(-\frac{1}{\lambda_d^2}\left(x_d^{(i)} - x_d^{(j)}\right)\right) \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{1}{\lambda_d^2}\left(x_d^{(i)} - x_d^{(j)}\right)^2\right) \tag{4.20}$$

$$\begin{aligned}
\frac{\partial^2}{\partial x_g^{(i)} \partial x_h^{(j)}} k\left(x_i, x_j\right) = \sigma_f^2 \times \frac{1}{\lambda_g^2} \times \left(\delta_{gh} - \frac{1}{\lambda_h^2}\left(x_h^{(i)} - x_h^{(j)}\right)\left(x_g^{(i)} - x_g^{(j)}\right)\right) \\
\times \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{1}{\lambda_d^2}\left(x_d^{(i)} - x_d^{(j)}\right)^2\right)
\end{aligned} \tag{4.21}$$

where $\delta_{gh} = 1$ if $g = h$, and $\delta_{gh} = 0$ if $g \neq h$ [65].

4.2 Probabilistic Solver Model

We want to introduce a probabilistic model that allows to characterize the error of the solution of an initial-valued ODE, where the solution states are not explicitly defined and are not available in closed form. To do so, we replace the exact solution $x(t)$ with a finite dimensional representation $x^n(t, \theta)$, so that:

$$\begin{aligned}
\dot{x}(t) &= f\left(x(t), t\right) \\
x(t_0) &= x_0
\end{aligned} \tag{4.22}$$

and:

$$y(t) = h\left(x^n(t,\theta)\right) + \epsilon(t) \tag{4.23}$$
$$\epsilon(t) \sim N\left(0, \sigma_\epsilon^2\right)$$

Equation (4.22) represents an unobserved theoretical model, while (4.23) represents an observational model. We formulate the approximation of (4.22) at discrete points $\{t_i\}_{i=1}^n$ as a Bayesian inference problem, where it is required to know a prior probability measure and a likelihood function. From this information, a posterior probability measure is defined using Bayes' Theorem.

Let $y_{1:t} := (y_1, y_2 \ldots, y_n)$ be the observed data, $x_{1:t} := (x_1, x_2 \ldots, x_n)$ the unknown states, $p\left(y_{1:t}|x_{1:t}\right)$ the likelihood, and $p_y\left(\theta\right)$ the prior distribution over the solution space. Then, the posterior distribution is obtained by:

$$p\left(y_{1:t}|x_{1:t}\right) = \frac{p\left(y_{1:t}|x_{1:t}p_y\left(\theta\right)\right)}{\int p\left(y_{1:t}|x_{1:t}p_y\left(\theta\right)\right)d\theta} \tag{4.24}$$

Following the theory developed in the works of Skilling [45] and Chkrebttii et al. [2], it is proposed to model the uncertainty of the solution of an ODE by means of a Gaussian process in a space of smoothed functions characterized on time interval $[0, T]$, by implementing an efficient algorithm that allows estimating the solution states $x(t)$ and the derivative of the solution sequentially under the conditions established in the model defined in (1.2).

In this method, the existence of the solutions is well grounded [66], the initial condition $x(0) = x_0$ represents the exact solution at the boundary conditions and the vector field $f\left(x(t), t, \theta\right)$ provides an approximation to the derivative $\dot{f}(x)$ in a domain $[0, T]$.

For an index $i \in \{1, 2, \ldots, n\}$, let (x_i, y_i) be the observations where x_i is the input variable, $y_i = f(x_i) + \epsilon_i$ is the response variable, with $\epsilon_i \sim N\left(0, \sigma_\epsilon^2\right)$. It is of interest to estimate $\dot{f}(x)$ without making any prior assumptions. First, let:

$$f(x) \sim GP\left(m(.), k\left(.,.\right)\right)$$

Yaglom [67] proved that if $k\left(.,.\right)$ is twice differentiable at the origin, $f(x)$ is differentiable in root mean square. Then:

$$\dot{f}(x) \sim GP\left(0, \frac{dk\left(x, x'\right)}{dxdx'}\right)$$

Given the parametric form of $k(.,.)$, the data given in (x_i, y_i) is useful to estimate the parameters of $k(.,.)$, and then use:

$$k''(.,.) = \frac{dk(x, x')}{dxdx'} \tag{4.25}$$

to make inferences about $\dot{f}(x)$.

An alternative method is to assume that the derivative process $\dot{y} = \dot{f}(x)$ is a GP. Then, the process with noisy observations is:

$$f(x) = \int_0^x f(s)ds \tag{4.26}$$

The covariance function of $f(x)$ can be obtained using a double integration over the covariance of $\dot{f}(x)$. We assume that:

$$\dot{y} = \dot{f}(x) \sim GP\left(0, k\left(x, x'\right)\right) \tag{4.27}$$

Then, equation (4.27) implies that:

$$y = f(x) \sim GP\left(0, \int_0^x \int_0^{x'} k\left(s, s'\right) dsds'\right) \tag{4.28}$$

In this paper, we consider an approximation of the solutions of the ODE at n points $\{t_1, t_2, \ldots, t_n\}$, denoted by:

$$\hat{x}(t) = (\hat{x}(t_1), \hat{x}(t_2), \ldots, \hat{x}(t_n))^T \tag{4.29}$$

We evaluate the vector field $y = f(\hat{x}(t))$, so that:

$$f(x) = \left(f(\hat{x}(t_1)), f(\hat{x}(t_2)) \ldots, f(\hat{x}(t_n))\right)^T \tag{4.30}$$

The likelihood of the model depends on (4.28), which implies that for each pair of locations $\left(x(t_i), x(t_j)\right)$ a double integral calculation is required. When n is sufficiently large, this implies a high computational cost. An alternative to this operation, is to consider a set of time points $\{\tau_1, \tau_2, \ldots, \tau_n\}$, thus obtaining:

$$\hat{x}^*(t) = (\hat{x}^*(\tau_1), \hat{x}^*(\tau_2), \ldots, \hat{x}^*(\tau_n))^T \tag{4.31}$$

Evaluating (4.31) over the vector field $\dot{y} = \dot{f}(\hat{x}^*(t))$, it follows that:

$$\dot{f}(x) = \left(\dot{f}(\hat{x}(\tau_1)), \dot{f}(\hat{x}(\tau_2)) \ldots, \dot{f}(\hat{x}(\tau_n))\right)^T \tag{4.32}$$

Therefore, by conditioning the joint distribution $(y, \dot{y})^T$ over $\dot{y}$ the computational complexity is simplified [2]. To make inference, the parameters θ of the continuous model are set, as well as the hyperparameters $\Sigma = \left(\sigma_f^2, \lambda, \sigma_\epsilon^2, x_0\right)$ that are associated with the estimation errors. Finally, the model likelihood, which represents the probabilistic solution of the system, is written as:

$$p\left(x\left(t,\theta\right), y, \dot{y}, \theta, \Sigma\right) \propto p\left(x\left(t,\theta\right)|y, \dot{y}, \theta, \Sigma\right) p\left(y|\dot{y}, \theta, \Sigma\right) p\left(\dot{y}|\Sigma\right) \tag{4.33}$$

The posterior distribution of the model is estimated by:

$$p\left(x\left(t,\theta\right), \theta, \Sigma|y(t), \dot{y}(t)\right) \propto p\left(x\left(t,\theta\right)|y, \dot{y}, \theta, \Sigma\right) p\left(y|\dot{y}, \theta, \Sigma\right) p\left(\dot{y}|\Sigma\right) p\left(\theta, \Sigma\right) \tag{4.34}$$

To determine the solution of an ODE as a statistical inference problem defined in the structure of the state-space models (4.22) and (4.23), we consider a prior distribution that models it using a Gaussian process and its $d-1$ derivatives:

$$\left(x(t), \dot{x}(t), x^{(2)}(t), \ldots, x^{(q-1)}(t)\right) : [0, T] \to \mathbb{R}^d \tag{4.35}$$

where $x = \left(x_t^{(1)}, \ldots, x_t^{(d)}\right)^T$ for $t \in [0, T]$ is a d-times integrated Wiener process. Here, the dynamics of x_t represents the solutions of the differential equation.

We consider prior distributions given by the Gaussian process:

$$x(t) \sim GP\left(m(\cdot), k(t, t')\right)$$

where $m(.)$ is the mean function and $k(t, t')$ is the covariance function. We can represent $x(t)$ by components:

$$x(t) = \left[x^{(1)}(t), x^{(2)}(t), \ldots, x^{(q-1)}(t)\right]^T \tag{4.36}$$

where $x^{(1)}(t)$ and $x^{(2)}(t)$ model $x(t)$ and $\dot{x}(t)$, respectively. The remaining $q-1$ subvectors in $u(t)$ can be used to model the higher order derivatives in $x(t)$ [68, 69].

The introduction of prior measurements through a Gaussian random vector field reflects uncertainty about the model approximations, includes information about the quantities of interest, and allows the implementation of computationally efficient algorithms. Chkrebtii et al. [2] and, more recently, Tronarp et al. [70] proposed a joint initial prior distribution of the uncertainty based on the solution and its derivative, with mean vectors $\dot{m}_0$ and m_0, covariance matrices $\dot{k}_0(t, t')$ and $k_0(t, t')$, and cross-covariance matrices $\tilde{k}_0(t, t')$ and $\tilde{k}_0(t', t)$.

In order to satisfy the condition $x(0) = x_0$, the following restrictions are imposed:

$$m_0(t) = \int_0^t \dot{m}_0(z)dz + x_0 \tag{4.37}$$

$$\dot{k}_0(0,0) = 0 \tag{4.38}$$

The initial joint prior distribution and its derivative are evaluated at time point vectors t_i and t_j, which may be different. These are defined as:

$$\left(\dot{x}(t_i), x(t_j)\right)^T | f_0 \sim GP\left(\begin{pmatrix} \dot{m}_0(t_i) \\ m_0\left(t_j\right) \end{pmatrix}; \begin{pmatrix} \dot{k}_0(t_i,t_i) & \tilde{k}_0\left(t_i,t_j\right) \\ \tilde{k}_0\left(t_j,t_i\right) & k_0\left(t_j,t_j\right) \end{pmatrix}\right) \tag{4.39}$$

The matrix $\dot{k}_0\left(t_i,t_j\right)$ has entries given by:

$$\dot{k}_0\left(t_i,t_j\right) = \int_0^{t_i}\int_0^{t_j} \dot{k}_0(z,w)dzdw \tag{4.40}$$

On the other hand, the entries of the cross-covariance matrices are given by:

$$\tilde{k}_0\left(t_i,t_j\right) = \int_0^{t_j} \dot{k}_0\left(t_i,z\right)dz \tag{4.41}$$

$$\tilde{k}_0\left(t_j,t_i\right) = \int_0^{t_j} \dot{k}_0\left(z,t_i\right)dz \tag{4.42}$$

The solution $x(t)$ and its derivative $\dot{x}(t)$ at (4.39) can be updated by conditioning the model information over the partition $\tau = (\tau_1, \ldots, \tau_n)$. Evaluating the model in:

$$f_n = f\left(\tau_n, x(\tau_n), \theta\right) = \dot{x}(\tau_n) \tag{4.43}$$

the marginal posterior predictive distribution $x\left(\tau_n\right)|\dot{x}(\tau_{n-1})$ is obtained. Since the marginal distributions follow the distribution of a GP [2, 5, 24], the solution vector is distributed as:

$$x(t_j)|\dot{x}(t_i) \sim GP\left(\dot{m}_0, \dot{k}(.,.))\right)$$

Analogously, the marginal distribution of the derivative is obtained:

$$\dot{x}(t_i)|x(t_j) \sim GP\left(m_0, k_0(.,.)\right)$$

The updating of (4.39) is performed sequentially, so that the derivative of the exact solution at the initial condition $\dot{x}_0^*\left(\tau_1\right)$ in $\tau_1 = 0$ is obtained by evaluating the function:

$$\begin{aligned} f_1 &= f\left(\tau_1, x_0^*(\tau_1), \theta\right) \\ &= \dot{x}^*\left(\tau_1\right) \end{aligned} \tag{4.44}$$

In accordance with the aforementioned, the next iteration will be given by:

$$\left(\dot{x}(t_i), x(t_j)\right)^T | f_1 \sim N\left(\begin{pmatrix} \dot{m}_1(t_i) \\ m_1\left(t_j\right) \end{pmatrix}; \begin{pmatrix} \dot{k}_1(t_i,t_i) & \tilde{k}_1\left(t_i,t_j\right) \\ \tilde{k}_1\left(t_j,t_i\right) & k_1\left(t_j,t_j\right) \end{pmatrix}\right) \tag{4.45}$$

Means and covariances are updated in the time vectors t_i and t_j, so that:

$$\dot{m}_1(t_i) = \dot{m}_0(t_i) + \dot{k}_0\left(t_i,\tau_1\right)\dot{k}_0\left(\tau_1,\tau_1\right)^{-1}\left[f_1 - \dot{m}_0(\tau_1)\right] \tag{4.46}$$

$$m_1(t_j) = m_0(t_j) + \dot{k}_0\left(\tau_1,\tau_1\right)^{-1}\tilde{k}_0(t_j,\tau_1)\left[f_1 - \dot{m}_0(\tau_1)\right] \tag{4.47}$$

$$\dot{k}_1(t_i,t_i) = \dot{k}_0(t_i,t_i) - \dot{k}_0\left(t_i,\tau_1\right)\dot{k}_0\left(\tau_1,\tau_1\right)^{-1}\dot{k}_0\left(\tau_1,t_i\right) \tag{4.48}$$

$$\dot{k}_1(t_j,t_j) = \dot{k}_0(t_j,t_j) - \tilde{k}_0\left(t_j,\tau_1\right)\dot{k}_0\left(\tau_1,\tau_1\right)^{-1}\left(\tilde{k}_0\left(t_j,\tau_1\right)\right)^T \tag{4.49}$$

Following a similar process, the second realization f_2 is obtained by simulating $x(\tau_2)$ at time τ_2 of the posterior predictive distribution:

$$x\left(\tau_2\right)|f_1 \sim GP\left(m_1\left(\tau_2\right), \dot{k}_1(\tau_1,\tau_2)\right)$$

Then, f_2 evaluates to:

$$f_2 = f\left(\tau_2, x_1(\tau_2), \theta\right) = \dot{x}(\tau_2) \tag{4.50}$$

The simulated value $x\left(\tau_2\right)|f_1$ does not guarantee that f_2 is equal to its derivative over τ_2. This implies the existence of an error $\epsilon^2_{\tau_2} = \left(f_2 - \dot{x}(\tau_2)\right)^2$. In general, the total error can be quantified as follows:

$$\sum_i^d \epsilon^2_{\tau_i} = \sum_i^d \left(\dot{x}(\tau_i) - f\left(\tau_i, x(\tau_i), \theta\right)\right)^2 \tag{4.51}$$

A logical way to obtain new data is to simulate it from a Gaussian process of the form:

$$f_2|f_1 \sim GP\left(\dot{m}\left(\tau_2\right), \Lambda_1(\tau_2)\right)$$

where the mean is $\dot{m}\left(\tau_2\right) = \dot{x}\left(\tau_2\right)$, and the covariance is $\Lambda_1(\tau_2) = \dot{k}_1(\tau_2,\tau_2)$.

The iterative process continues, and in the next step (4.45) is updated. Thus:

$$\left(\dot{x}(t), x(t)\right)^T | f_2, f_1 \sim N\left(\begin{pmatrix} \dot{m}_2(t_i) \\ m_2\left(t_j\right) \end{pmatrix}; \begin{pmatrix} \dot{k}_2(t_i,t_i) & \tilde{k}_2\left(t_i,t_j\right) \\ \tilde{k}_2\left(t_j,t_i\right) & k_2\left(t_j,t_j\right) \end{pmatrix}\right) \tag{4.52}$$

The marginal means and covariances evaluated on the time vectors t_i and t_j are updated as:

$$\dot{m}_2(t_i) = \dot{m}_1(t_i) + \dot{k}_1\left(t_i,\tau_2\right)\left(\dot{k}_1\left(\tau_2,\tau_2\right) + \Lambda_1(\tau_2)\right)^{-1}\left[f_2 - \dot{m}_1(\tau_2)\right] \tag{4.53}$$

$$m_2\left(t_j\right) = m_1(t_j) + \tilde{k}_1\left(t_j, \tau_2\right)\left(\dot{k}_1\left(\tau_2, \tau_2\right) + \Lambda_1(\tau_2)\right)^{-1}\left[f_2 - \dot{m}_1(\tau_2)\right] \tag{4.54}$$

$$\dot{k}_2(t_i, t_i) = \dot{k}_1(t_i, t_i) - \dot{k}_1\left(t_i, \tau_2\right)\left(\dot{k}_1\left(\tau_2, \tau_2\right) + \Lambda_1(\tau_2)\right)^{-1}\dot{k}_1\left(\tau_2, t_i\right) \tag{4.55}$$

$$\dot{k}_2(t_j, t_j) = \dot{k}_1(t_j, t_j) - \tilde{k}_1\left(t_j, \tau_2\right)\left(\dot{k}_1\left(\tau_2, \tau_2\right) + \Lambda_1(\tau_2)\right)^{-1}\left(\tilde{k}_1\left(t_j, \tau_2\right)\right)^T \tag{4.56}$$

After n steps, $x(\tau_n)$ is generated at τ_n from the predicted marginal posterior distribution:

$$x\left(\tau_n\right)|f_{n-1}, \ldots, f_1 \sim GP\left(\dot{m}\left(\tau_n\right), \Lambda_{n-1}\left(\tau_n\right)\right)$$

where $\Lambda_{n-1}(\tau_n) = \dot{k}_{n-1}(\tau_n, \tau_n)$.

The n-th update of (4.45) is carried out by evaluating t_i and t_j as follows:

$$\left(\dot{x}(t), x(t)\right)^T|f_n, \ldots, f_1 \sim N\left(\begin{pmatrix} \dot{m}_n(t_i) \\ m_n\left(t_j\right) \end{pmatrix}; \begin{pmatrix} \dot{k}_n(t_i, t_i) & \tilde{k}_n\left(t_i, t_j\right) \\ \tilde{k}_n\left(t_j, t_i\right) & k_n\left(t_j, t_j\right) \end{pmatrix}\right)$$

Finally, the marginal means and covariances evaluated at t_i and t_j are updated by:

$$\dot{m}_n(t_i) = \dot{m}_{n-1}(t_i) + \dot{k}_{n-1}\left(t_i, \tau_n\right)\left(\dot{k}_{n-1}\left(\tau_n, \tau_n\right) + \Lambda_{n-1}(\tau_n)\right)^{-1}\left[f_n - \dot{m}_{n-1}(\tau_n)\right] \tag{4.57}$$

$$m_n\left(t_j\right) = m_{n-1}(t_j) + \left(\dot{k}_{n-1}\left(\tau_n, \tau_n\right) + \Lambda_{n-1}(\tau_n)\right)^{-1}\tilde{k}_n\left(t_j, t_i\right)\left[f_n - \dot{m}_{n-1}(\tau_n)\right] \tag{4.58}$$

$$\dot{k}_n(t_i, t_j) = \dot{k}_{n-1}(t_i, t_j) - \dot{k}_{n-1}\left(t_i, \tau_n\right)\left(\dot{k}_{n-1}\left(\tau_n, \tau_n\right) + \Lambda_{n-1}(\tau_n)\right)^{-1}\left(\tilde{k}_{n-1}\left(\tau_n, t_j\right)\right)^T \tag{4.59}$$

$$\dot{k}_n(t_i, t_j) = \dot{k}_{n-1}(t_i, t_j) - \left(\dot{k}_{n-1}\left(\tau_n, \tau_n\right) + \Lambda_{n-1}(\tau_n)\right)^{-1}\tilde{k}_{n-1}\left(t_i, \tau_n\right)\left(\tilde{k}_{n-1}\left(t_j, \tau_n\right)\right)^T \tag{4.60}$$

Chkrebtii et al. [2] and Overstall et al. [5] sum all the steps above in a sequential algorithm for updating and sampling, at the times $\mathbf{t} = (t_1, \ldots, t_n)^T$, a bivariate Gaussian process between $x(t) = (x(t_1), \ldots, x(t_n))^T$ and its derivative $\dot{x}(t) = f\left(t, x(t), \theta\right)^T$, also using and a time grid $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)^T$ to evaluate $\dot{x}(t) = (\dot{x}(\tau_1), \ldots, \dot{x}(\tau_n))^T$.

Algorithm 01: Sequential sampling and updating of GP-based solutions of a ODE system.

---

1. Let $\mathbf{x}(0) = \mathbf{x}_0$, $\tau_1 = t_0$, $\boldsymbol{\Lambda}_1 = 0$, y $f_1 = f\left(x_0, t_0, \theta\right)$
2. For $r = 1, \ldots, n-1$
   (a) Let $\boldsymbol{\tau}_r = (\tau_1, \ldots, \tau_r)^T$, $\mathbf{t}_r = (t_1, \ldots, t_r)^T$

(b) Compute

$$\mathbf{B}_r = \left(\dot{k}_0\left(\boldsymbol{\tau}_r, \boldsymbol{\tau}_r\right) + \boldsymbol{\Lambda}_r\right)^{-1}$$
$$\mathbf{a}_r = \mathbf{B}_r \times \tilde{k}_0\left(\boldsymbol{\tau}_r, \tau_{r+1}\right)$$
$$k_r = k_0\left(\tau_r, \tau_r\right) - \tilde{k}_0\left(\tau_{r+1}, \boldsymbol{\tau}_r\right) \times \mathbf{B}_r \times \tilde{k}_0\left(\boldsymbol{\tau}_r, \tau_{r+1}\right)$$
$$\dot{k}_{r+1} = \dot{k}_0\left(\tau_{r+1}, \tau_{r+1}\right) - \dot{k}_0\left(\tau_{r+1}, \boldsymbol{\tau}_r\right) \times \mathbf{B}_r \times \dot{k}_0\left(\boldsymbol{\tau}_r, \tau_{r+1}\right)$$
$$\boldsymbol{\Lambda}_{r+1} = diag\left(\boldsymbol{\Lambda}_r, \dot{k}_{r+1}\right)$$

(c) Compute

$$\mathbf{m}_r = \mathbf{x}_0 + \mathbf{F}_r^T \times \mathbf{a}_r$$

where $\mathbf{F}_r \in \mathcal{M}_{rs}$ composed by elements of f, with the i-th row given by f_i for $i = 1, \ldots, n-1$

(d) Sample

$$\mathbf{x}\left(\tau_{r+1}\right) \sim GP\left(m_r, k_r I_S\right)$$

(e) Evaluate

$$f_{\tau+1} = f\left(\mathbf{x}(\tau_{r+1}), \tau_{r+1}, \theta\right)$$

3. Compute:

$$\mathbf{B}_n = \left(\dot{k}_0\left(\boldsymbol{\tau}_n, \boldsymbol{\tau}_n\right) + \boldsymbol{\Lambda}_n\right)^{-1}$$
$$\mathbf{A}_n(\mathbf{t}) = \mathbf{B}_n \times \tilde{k}_0\left(\boldsymbol{\tau}, \mathbf{t}\right)$$
$$\mathbf{M}_n(\mathbf{t}) = 1_m \times \mathbf{x}_0^T + \mathbf{A}_n(\mathbf{t}) \times \mathbf{F}_n$$

where 1_m is a m-vector with all entries equaling to 1, $F_n \in \mathcal{M}_{ns}$ with k-th row given by f_k for $k = 1, \ldots, n$, and

$$k_n\left(\mathbf{t}, t\right) = k_0(\mathbf{t}) - \tilde{k}_0\left(t, \boldsymbol{\tau}\right) \mathbf{B}_N \tilde{k}_0\left(\boldsymbol{\tau}, \mathbf{t}\right)$$

4. For $l = 1, \ldots, s$

$$x_l(t_1), \ldots, x_l(t_n) | \dot{x}_l(\tau_1), \ldots, \dot{x}_l(\tau_n) \sim N\left(\mathbf{M}_N(t) \times \mathbf{e}_l, k_n\left(\mathbf{t}, \mathbf{t}\right)\right)$$

where $\mathbf{e}_l$ is the l-th unitary vector

Given an initial solution $\mathbf{x}_0$ and parameters θ, the prior GP can update the solutions using the derivative evaluations over the grid of points $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)^T$ by implementing Algorithm 1, sequentially conditioning on $f(x, \tau_{r+1}; \theta)$ and calculating the solution state x_l from sampling the posterior distribution over x_r. The marginal GP distribution for $x_l(t)$ is given by:

$$m_{nl}(\mathbf{t}) = m_{0l} + \tilde{k}_0(\mathbf{t}, \boldsymbol{\tau}) \times \mathbf{B}_n \times \mathbf{F}_n \times \mathbf{e}_l \tag{4.61}$$

$$k_n\left(\mathbf{t}, \mathbf{t}'\right) = k_0\left(\mathbf{t}, \mathbf{t}'\right) - \tilde{k}_0(\mathbf{t}, \boldsymbol{\tau}) \times \mathbf{B}_n \times \tilde{k}_0(\boldsymbol{\tau}, \mathbf{t}) \tag{4.62}$$

for $l \in \{1, \ldots, s\}$. In (4.61) and (4.62), $\mathbf{F}_n$ is the matrix of the evaluations of the derivatives and $\mathbf{B}_n$ is the covariance matrix of the updated derivatives.

4.3 Polynomial Chaos Expansion (PCE) and Gaussian Processes (GP)

Consider a dynamic system whose behavior is represented by the following mathematical model:

$$Y = f(X), \quad X \in \mathcal{D}_X \subset \mathbb{R}^M \tag{4.63}$$

where $X = (X_1, \ldots, X_M)^T$ represents the input parameters of the system, while Y is the quantity of interest.

Once f has been chosen, the sources of uncertainty must be quantified. The available information is processed to build the probabilistic model that fits the input data, represented by a random vector X and characterized by a probability density $f_X(\cdot)$. If data are assumed to be independent, the joint distribution can be defined by the following set of marginal distributions:

$$f_X(x_1, \ldots, x_M) = \prod_{i=1}^{M} f_{X_i}(x_i) \tag{4.64}$$

Consequently, $Y = f(X)$ becomes a random variable whose properties are implicitly defined by the propagation of uncertainty and which is described by the joint probability distribution $f_X(.)$.

Let Y be a second order stochastic process. If Y has finite variance, then it belongs to a Hilbert space and can be represented by:

$$f_{PCE}(X) = \sum_{i=0}^{\infty} \beta_i Z_i \tag{4.65}$$

where Y is an infinite series, $\{Z_i\}_{i=0}^{\infty}$ denotes a numerable set of random variables which forms a Hilbert space basis, and $\{\beta_i\}$ are the coefficients of the series.

Hilbert spaces guarantee the existence of such bases and their representation, however, there are many ways to represent them. In the case of polynomial chaos expansions, the terms of bases $\{Z_i\}_{i=0}^{\infty}$ are multivariate orthonormal polynomials with an input vector X, i.e, $Z_i = \Psi_i(X)$. The approximation (4.65) can be rewritten by truncating the series into M terms:

$$f_{PCE}(X) = \sum_{i=0}^{M} \beta_i \Psi_i(X) \tag{4.66}$$

For the construction of the basis, we assume the existence of a random vector of independent components denoted by $\{X_i, \quad i = 1, \ldots, n\}$, and of two arbitrary functions $\phi_1, \phi_2 : x \in \mathcal{D}_{x_i} \to \mathbb{R}$. A functional inner product is defined as:

$$\begin{aligned} \langle \phi_1, \phi_2 \rangle_i &= \int_{\mathcal{D}_{x_i}} \phi_1(x)\phi_2(x) f_{X_i}(x) dx \\ &= \mathrm{E}_{f_{X_i}(x)} \left(\phi_1(x)\phi_2(x) \right) \end{aligned} \tag{4.67}$$

Functions of this type are said to be orthogonal with respect to the probability measure $\mathbb{P}(dx) = f_{X_i}(x)dx$, if:

$$\mathrm{E} \left(\phi_1(x)\phi_2(x) \right) = 0 \tag{4.68}$$

Using the previous notation and classical algebra operations, we can construct a family of orthogonal polynomials $\left\{ \pi_k^{(i)}, \quad k \in \mathbb{N} \right\}$ that satisfies:

$$\begin{aligned} \langle \pi_j^{(i)}, \pi_k^{(i)} \rangle &= \mathrm{E} \left\{ \pi_j^{(i)}(x_i) \pi_k^{(i)}(x_i) \right\} \\ &= \int_{\mathcal{D}_{x_i}} \pi_j^{(i)}(x_i) \pi_k^{(i)}(x_i) f_{X_i}(x) dx \\ &= a_j^i \delta_{ik} \end{aligned} \tag{4.69}$$

where the subscript k denotes the degree of the polynomial $\pi_k^{(i)}$, δ_{ik} is the Kronecker delta function, and $a_j^i = \langle \pi_j^{(i)}, \pi_j^{(i)} \rangle_i$.

The family of polynomials can be obtained by applying Gram-Schmidt orthogonalization to the canonical form of monomials $\left\{1, x, x^2, \ldots\right\}$. For standard distributions as the associated family of orthogonal polynomials is known, being Legendre, Hermite, Laguerre or Jacobi polynomials. However, the family obtained is usually not orthonormal, so the following normalization technique is applied:

$$\Psi_j^{(i)} = \frac{\pi_j^{(i)}}{\sqrt{a_j^i}} \tag{4.70}$$

for the indexes $i \in \{1, \ldots, M\}$ and $j = \{1, \ldots, \mathbb{N}\}$.

To extend the computation of bases as seen in (4.65) to the case of multivariate polynomials, tensor products of orthonormal univariate polynomials are constructed by defining tuples or multiple indices $\alpha \in \mathbb{N}^M$, which represent ordered lists of natural numbers of the form:

$$\alpha = (\alpha_1, \ldots, \alpha_M)^T \tag{4.71}$$

A multivariate polynomial Ψ_α can be associated by any multiple index α by means of:

$$\Psi_\alpha(x) = \prod_{i=1}^{M} \Psi_{\alpha_i}^{(i)}(x_i) \tag{4.72}$$

where the polynomials $\left\{\Psi_k^{(i)} : k \in \mathbb{N}\right\}$ are defined according to the i-th marginal distribution given in (4.69) and (4.70). Moreover, multivariate polynomials with input vector X are also orthonormal. Consequently, for all $\alpha, \theta \in \mathbb{N}^M$ we have that:

$$\begin{aligned} \mathrm{E}\left\{\Psi_\alpha(X)\Psi_\theta(X)\right\} &= \int \Psi_\alpha(X)\Psi_\theta(X) f_X(x) dx \\ &= \delta_{\alpha\theta} \end{aligned} \tag{4.73}$$

where $\delta_{\alpha\theta}$ is the Kronecker delta function.

Given the above, it can be proved that the set of all multivariate polynomials with an input random vector X forms a basis in Hilbert spaces, in which $Y = f(X)$ is represented as:

$$\hat{f}_{PCE}(X) = \sum_{\theta \in \mathbb{N}^M} \beta_\theta \Psi_\theta(X) \tag{4.74}$$

By the orthogonal property of the polynomial chaos basis (4.73), each expansion coefficient

can be computed as follows:

$$\begin{aligned}\mathrm{E}\left\{f(X)\Psi_\alpha(X)\right\} &= \mathrm{E}\left\{\beta_\theta \sum_{\theta\in\mathbb{N}^M} \Psi_\alpha(X)\Psi_\theta(X)\right\} \\ &= \mathrm{E}\left\{\sum_{\theta\in\mathbb{N}^M} \beta_\theta \delta_{\alpha\theta}\right\} \\ &= \beta_\alpha \end{aligned} \tag{4.75}$$

where $\alpha = \theta$. Equivalently:

$$\begin{aligned}\beta_l &= \mathrm{E}\left\{f(X)\Psi_l(X)\right\} \\ &= \int f(X)\Psi_l(X) f_X(x) dx \\ &\approx \sum_{i=1}^{N} \xi_i f_{X_i}(x_i)\Psi_l(X_i) \end{aligned} \tag{4.76}$$

for $l \in \{0,\ldots,M\}$. En (4.76), the second equation is obtained by numerical integration techniques, such as the Gaussian quadrature rule; where $\{X_i\}$ in $i \in \{1,\ldots,N\}$ and $\{\xi_i\}$ in $i \in \{1,\ldots,N\}$ represent the nodes and weights, respectively. This equation can also be obtained using a Monte Carlo type algorithm, such as importance sampling:

$$\begin{aligned}\beta_l &= \mathrm{E}\left\{f(X)\Psi_l(X)\right\} \\ &= \int f(X)\Psi_l(X) f_X(x) dx \\ &\approx \frac{1}{n}\sum_{i=1}^{N} \Psi_l(x_i)\frac{f_{X_i}(x_i)}{q_{X_i}(x_i)} \\ &= \frac{1}{n}\sum_{i=1}^{N} \xi_i \Psi_l(x_i) \end{aligned} \tag{4.77}$$

with $\xi_i = \dfrac{f_{X_i}(x_i)}{q_{X_i}(x_i)}$ and $l \in \{0,\ldots,M\}$. Here, $q_X(x)$ is known as an importance function, which is generally easy to sample and has a support containing $f_X(x)$.

The series (4.74) is infinite, and in order to work with it it must be truncated to M terms, so that:

$$\begin{aligned}\hat{f}_{PCE}(X) &= \sum_{i=0}^{M}\left\{\sum_{i=1}^{N}\xi_i f(X_i)\Psi_i(X_i)\right\}\Psi_i(x) \\ &= \sum_{i=0}^{M}\beta_i\Psi_i(x) \end{aligned} \tag{4.78}$$

In practice, experimental data include an error rate that is mathematically represented as a random variable $\epsilon_i \sim N\left(0, \sigma^2\right)$. Then, for each component $i \in \{1, \ldots, N\}$, the response is written as:

$$Y_i = \hat{f}_{PCE}\left(X_i\right) + \epsilon_i \tag{4.79}$$

with $\hat{f}_{PCE}\left(X_i\right)$ as a function that remains unknown.

Now, we can connect the polynomial chaos expansion technique to Gaussian processes. Consider an input $X = \{X_i, i = 1, \ldots, N\}$ and an output response $Y = \{Y_i, i = 1, \ldots, N\}$, where:

$$\begin{aligned} Y &= f(X) + \epsilon \\ \epsilon &\sim N\left(0, \sigma_\epsilon^2 \mathbf{I}\right) \\ f(X) &\sim GP\left(m_{prior}(X), k_{prior}(X, X)\right) \end{aligned} \tag{4.80}$$

Using Bayes' Theorem, new data with its related error can be predicted using the posterior distribution:

$$f(x)|Y, X, x, \theta \sim N\left(m_{post}(x), k_{post}(x, x)\right)$$

where:

$$m_{post}(x) = K_x^T \left(K + \sigma_\epsilon^2 \mathbf{I}\right)^{-1} Y \tag{4.81}$$

$$k_{post}(x, x) = K_{xx} - K_x^T \left(K + \sigma_\epsilon^2 \mathbf{I}\right)^{-1} K_x \tag{4.82}$$

where the covariance matrices are denoted as $K = k\left(X, X\right) \in \mathbb{R}^{N \times N}$, $K_{ij} = k\left(X_i, X_j\right)$, $K_x = k\left(X, x\right) \in \mathbb{R}^{N \times 1}$ and $K_{xx} = k(x, x) \in \mathbb{R}$.

On the other hand, the equation (4.81) can be represented as a combination of N kernels of functions:

$$\begin{aligned} m_{post}(x) &= \hat{f}_{GP}(x) \\ &= \sum_{i=1}^{N} \beta_i k\left(X_i, x\right) \end{aligned} \tag{4.83}$$

with $\beta = \left(K + \sigma_\epsilon^2 \mathbf{I}\right)^{-1} Y$.

Polynomial chaos expansions and Gaussian processes can be studied under the same structure and combined to improve estimation by linking through the reproduction of kernels of

covariance functions in Hilbert spaces. Other approximation methods have been developed in recent literature using the structure of reduced rank models combined with Gaussian processes [53, 71]. It consists mainly in defining the isotropic covariance function $k\left(x,x'\right)=k\left(\|\mathbf{r}\|\right)$, where $\mathbf{r}=\mathbf{x}-\mathbf{x}'$ is a distance measurement under the Euclidean norm $\|.\|$.

If $k\left(\mathbf{r}\right)$ is continuous and positive definite, it is defined as:

$$k\left(x,x'\right)=\frac{1}{\left(2\pi\right)^{d}}\int\exp\left(i\omega^{T}\mathbf{r}\right)\mu\left(d\omega\right) \tag{4.84}$$

where μ is a positive measure. When the measure $\mu(d\omega)=S(\omega)d\omega$ has a spectral density, $S(\omega)$, associated with the covariance $k(\mathbf{r})$. The definition allows establishing a Fourier duality between the covariance and the spectral density, formalized as the Wiener-Khinchin theorem.

The application of this theorem results in the following identities:

$$k\left(\mathbf{r}\right)=\frac{1}{\left(2\pi\right)^{d}}\int S(\omega)\exp\left(i\omega^{T}r\right)d\omega \tag{4.85}$$

$$S\left(\omega\right)=\int k\left(\mathbf{r}\right)\exp\left(-i\omega^{T}r\right)d\omega \tag{4.86}$$

In (4.84) and (4.85), $k\left(\mathbf{r}\right)$ is the inverse Fourier transform and $S\left(\omega\right)$ represents the Fourier transform.

Solin and Särkkä [53] define a covariance operator $\mathcal{K}$ as a pseudo differential operator associating to each covariance function $k\left(x,x'\right)$, which satisfies:

$$\mathcal{K}\phi=\int k\left(\cdot,x'\right)\phi\left(x'\right)dx' \tag{4.87}$$

The definition given in (4.87) allows the covariance to be approximated in similarly to the methods used to approximate differential and pseudo differential operators of partial differential equations in Hilbert spaces.

When covariance is isotropic, the Fourier representation is a transfer function with spectral density $S(\omega)$ following a Gaussian distribution, with an associated covariance operator:

$$S(\omega)\overset{\triangle}{=}S\left(\|\omega\|\right) \tag{4.88}$$

If a polynomial expansion is applied to the density, we have that:

$$S\left(\|\omega\|\right)=a_{0}+a_{1}\|\omega\|^{2}+a_{2}\left(\|\omega\|^{2}\right)^{2}+a_{3}\left(\|\omega\|^{2}\right)^{3}+a_{4}\left(\|\omega\|^{2}\right)^{4}+\ldots \tag{4.89}$$

Since the transfer function corresponds to Laplace's operator $\nabla^2 = -\|\omega\|^2$, for a regular f function it is satisfied that:

$$\mathcal{F}[\nabla^2 f](\omega) = -\|\omega\|^2 \mathcal{F}[f](\omega) \tag{4.90}$$

where $\mathcal{F}[.]$ denotes the Fourier transform of its argument.

Taking the inverse of the Fourier transform, it gives the covariance operator $\mathcal{K}$:

$$\mathcal{K} = a_0 + a_1\left(-\nabla^2\right) + a_2\left(-\nabla^2\right)^2 + a_3\left(-\nabla^2\right)^3 + a_4\left(\| - \nabla^2\right)^4 + \ldots \tag{4.91}$$

Now, let us define the pseudo-differential operator as a series of differential operators that approaches the Laplacian operator through Hilbert's method. In order to carry out the approximation of the covariance operator in Hilbert space, we consider a covariance function $k\left(x, x'\right)$ and the inner product:

$$\langle f(x), g(x)\rangle = \int_\Omega f(x)g(x)\omega(x)dx \tag{4.92}$$

where $\Omega \subset \mathbb{R}^d$ is a compact set. The inner product induces a Hilbert space of random functions.

If a basis $\left\{\phi_j(x)\right\}$ is fixed, a Gaussian process $f(x) = \left(f(x_1), \ldots, f(x_n)\right)^T$, can be expanded in terms of an infinite series:

$$f(x) = \sum_{j=1}^{\infty} \zeta_j \phi_j(x) \tag{4.93}$$

where each of the ζ_j components is iid in a multivariate Gaussian distribution, while $\phi_j(x)$ is obtained from the eigenfunctions of $k\left(x, x'\right)$ with respect to the inner product given in (4.92). Then, (4.93) becomes into a Karhunen-Loéve series.

To carry out the approximation of the Laplacian operator in a Hilbert space, the eigenvalue problem for Laplacian operators is considered:

$$-\nabla^2\phi_j(x) = \lambda_j\phi_j(x), \quad x \in \Omega \tag{4.94}$$

$$\phi_j(x) = 0, \qquad x \in \partial\Omega \tag{4.95}$$

Since the negative version of the Laplacian operator is positive definite, the set of eigenfunctions $\phi_j(.)$ are orthonormal with respect to (4.92), that is:

$$\int_\Omega \phi_i(x)\phi_j(x)dx = \delta_{ij} \tag{4.96}$$

where δ_{ij} is the Kronecker delta function, and all the eigenvalues λ_j are positive real numbers. Then, the negative Laplacian operator can be expressed in terms of an expected value:

$$-\nabla^2 f(x) = \int l\left(x, x'\right) f\left(x'\right) dx' \tag{4.97}$$

where $l\left(x, x'\right)$ is a kernel of the form:

$$l\left(x, x'\right) = \sum_{j=1}^{\infty} \lambda_j \phi\left(x\right) \phi\left(x'\right) \tag{4.98}$$

For $n \in \{1, 2, \ldots\}$, the development in powers of the operator defined in (4.98) is given by:

$$l^n\left(x, x'\right) = \sum_{j=1}^{\infty} \lambda_j^n \phi\left(x\right) \phi\left(x'\right) \tag{4.99}$$

where the negative Laplacian operator is:

$$\left(-\nabla^2\right)^n f(x) = \int l^n\left(x, x'\right) f\left(x'\right) dx' \tag{4.100}$$

Then, working (4.100) on both sides, we have that:

$$\begin{aligned} \mathcal{K}f(x) &= \left[a_0 + a_1\left(-\nabla^2\right) + a_2\left(-\nabla^2\right)^2 + a_3\left(-\nabla^2\right)^3 + \ldots\right] f(x) \\ &= \int \left[a_0 + a_1 l^1\left(x, x'\right) + a_2 l^2\left(x, x'\right) + a_3 l^3\left(x, x'\right) + \ldots\right] f\left(x'\right) dx' \end{aligned} \tag{4.101}$$

Comparing (4.101) with (4.87), and using the formula given in (4.99), we approximate the covariance function as:

$$\begin{aligned} k\left(x, x'\right) &\approx a_0 + a_1 l^1\left(x, x'\right) + a_2 l^2\left(x, x'\right) + a_3 l^3\left(x, x'\right) + \ldots \\ &= \sum_{j=1}^{\infty} \left[a_0 + a_1\lambda_j^1 + a_2\lambda_j^2 + a_3\lambda_j^3 + \ldots\right] \phi_j\left(x\right) \phi_j\left(x'\right) \end{aligned} \tag{4.102}$$

Evaluating the spectral density series at $\|\omega\|^2 = \lambda_j$, it follows that:

$$S\left(\sqrt{\lambda_j}\right) = a_0 + a_1\lambda_j^1 + a_2\lambda_j^2 + a_3\lambda_j^3 + \ldots \tag{4.103}$$

Substituting (4.103) into (4.102), an approximation of the form follows:

$$k\left(x, x'\right) \approx \sum_{j=1}^{\infty} S\left(\sqrt{\lambda_j}\right) \phi_j\left(x\right) \phi_j\left(x'\right) \tag{4.104}$$

where $S(.)$ is the spectral density of the covariance function, λ_j is the j-th eigenvalue, and $\phi_j(.)$ is the eigenvalue function of the Laplace operator.

Once the covariance approximation is calculated, it can be combined with Gaussian processes to make predictions. Recall that Gaussian processes can be formulated as a prediction of an unknown output $f(X_{new})$ associated with the entry of a new known fact $X_{new} \in \mathbb{R}^n$, given a set of training data $\mathbb{D} = \{(X_i, Y_i), \quad i = 1, \ldots, n\}$.

Evaluating $f(X_{new}) = \left(f\left(x_{new}^{(1)}\right) \ldots, f\left(x_{new}^{(n)}\right)\right)^T$ on a vector of new data X_{new} is considered as a realization of a GP, such that $f(x_{new}) \sim GP(0, k(x, x'))$. If an error rate $\epsilon_i \sim N\left(0, \sigma_\epsilon^2\right)$ is added to the process, we have that:

$$Y_i = f(X_i) + \epsilon_i \tag{4.105}$$

Rasmussen and Williams [30] showed that the posterior distribution of the new data is:

$$f(X_{new}) \,|\mathbb{D} \sim N\left(\mathrm{E}\left(f(X_{new})\right), \mathrm{Var}\left(f(X_{new})\right)\right)$$

where:

$$\mathrm{E}\left(f(X_{new})\right) = k_{new}^T \left(K + \sigma_\epsilon^2 \mathbf{I}\right)^{-1} Y \tag{4.106}$$

$$\mathrm{Var}\left(f(X_{new})\right) = k(x_{new}, x_{new}) - k_{new}^T \left(K + \sigma_\epsilon^2 \mathbf{I}\right)^{-1} k_{new} \tag{4.107}$$

with $K_{ij} = k\left(x_i, x_j\right)$, k_{new} is an n-dimensional vector with the j-th entry of $k\left(x_{new}, x_j\right)$.

To avoid the computation of inverse matrices in (4.106) and (4.107), we use the approximation obtained in (4.104) by performing a projection of the GP onto a truncated set of m basis functions such that:

$$f(x) = \sum_{j=1}^{m} \zeta_j \phi_j(x) \tag{4.108}$$

where $\zeta_j \sim N\left(0, S\left(\sqrt{\lambda_j}\right)\right)$.

From the above, it is concluded that it is possible to find an approximate decomposition of the covariance matrix $K \approx \Phi\Lambda\Phi$, with $\Lambda_{jj} = S\left(\sqrt{\lambda_j}\right)$ as a diagonal matrix of the eigenvalues for $j \in \{1, \ldots, m\}$.

Using the matrix inversion equations given in (4.106) and (4.107), we obtain:

$$\mathrm{E}\left(f(x_{new})\right) \approx \phi_{new} \left(\Phi^T\Phi + \sigma_\epsilon^2 \Lambda^{-1}\right)^{-1} \Phi^T y \tag{4.109}$$

$$\mathrm{Var}\left(f(x_{new})\right) \approx \sigma_\epsilon^2 \phi_{new}^T \left(\Phi^T\phi + \sigma_\epsilon^2 \Lambda^{-1}\right)^{-1} \phi_{new} \tag{4.110}$$

where $\phi_{new}(X_{new}) = \left(\phi_1(x_{new}^{(1)}), \ldots, \phi_m(x_{new}^{(n)})\right)^T$ is an m-dimensional vector with the j-th entry given by $\phi_j(x_{new})$ [53].

4.4 Experimental Setup

4.4.1 Computational Requirements

This research project is performed using the following hardware and software resources:

- **Hardware:** Computer with Intel® Core™ i5-7200U processor at 2.50GHz and 8 GB RAM, running under Windows 10 64-bit operating system.
- **Software:** Jupyter Notebooks 6.4.0 with Python 3 kernel, using the libraries *numpy* [72] and *scipy* [73] for the UQDES algorithm, *chaospy* [74] for the PCE addition, and *time* and *tensorflow* [75] for the metrics computation.

4.4.2 Computational Complexity

To analytically calculate the efficiency of the UQDES and PCE algorithms, a computational complexity study was carried out in [2] and [74] in order to analyze the amount of computational resources required to execute them. This was be interpreted through the statistical measure Big-O notation, through which the relationship between the size of the input and the steps that are executed in the algorithms will be estimated by means of mathematical notation that describes the behavior of the functions when the limit of the arguments tends to infinity.

In Algorithm 1, a Markovian assumption was used for make the modelling very fast-changing dynamics more flexible, reducing the computational cost compared to deterministic numerical methods. Also, considering the using of isotropic covariance structures, it requires O(n) operations for the mean and covariance updates [2] by the truncation of weight matrices. On the other hand, the polynomial chaos implementation offered by *chaospy* replaces the generation of multivariate pseudo-random realizations for the probability distribution used in most uncertainty quantification algorithms by a series of univariate realizations based on the assumption of stochastic independence of the dimensions [74], which considerably reduces the computational complexity of the algorithm in both pseudo-spectral projection and point

collocation methods, especially in high-dimensional cases.

In this work, the execution time will be used as the measure that will give a idea about the performance and, consequently, the efficiency of the algorithms.

4.4.3 Performance Measures

To evaluate the goodness of fit of the different algorithms, performance measures that allow us to make quantitative comparisons about the prediction rate of the applied techniques will be used. Literature shows a large number of metrics that are based on the difference between true and predicted values to measure the accuracy of a model. In this work, the metrics considered are closely related to the loss functions of L_1 and L_2 in the sense that the loss of L_1 minimizes the error, which is defined as the sum of all the absolute differences between true and predicted values, while the loss of L_2 minimizes the squared value of these differences. Measures based on squared differences place more emphasis on outliers: because of quadrature, predictions that are far from the true values are penalized more strongly than closer predictions.

Given a probabilistic model and a set of latent states, we denote $\hat{y}_t$ as a data estimate for a time point t, and y_t^e as the true data. Then, we define the following performance measures:

1. Root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{y}_t - y_{*t}\right)^2} \tag{4.111}$$

2. Standardized mean squared error (SMSE):

$$SMSE = \sum_{t=1}^{n_*}\frac{\left(y_{*t} - \mu_{*t}\right)^2}{\mathrm{Var}\left(y\right)} \tag{4.112}$$

where:

$$\mu_{*t} = \mathrm{E}\left(f(x_{*t})\right) \tag{4.113}$$

$$\sigma_{*t}^2 = \mathrm{Var}\left(f(x_{*t})\right) + \sigma_\epsilon^2 \tag{4.114}$$

are the predictive mean and variance for test sample $t = 1, 2, \ldots, n_*$, and y_{*t} is the actual test value. The real data variance is denoted by $\mathrm{Var}(y)$.

3. Mean standardized log loss (MSLL):

$$MSLL = \frac{1}{n_*} \sum_{t=1}^{n_*} \frac{(y_{*t} - \mu_{*t})^2}{\sigma_{*t}^2} + \ln 2\pi\sigma_{*t}^2 \tag{4.115}$$

whose definition follows the notation given in (4.113) and (4.114).

Chapter 5

Results and Discussion

To exemplify the methodology proposed above, three coupled differential equation models which are widely used in physics, epidemiology and biology have been proposed. In these systems, uncertainty is evaluated by comparing the estimated solutions with the numerical solution provided by the Python dependency *scipy.integrate*, and the effectiveness of the proposed solvers is quantified by performance measures such as root mean squared error (RMSE), standardized mean squared error (SMSE) and mean standardized log loss (MSLL).

5.1 Lorenz Attractor Model

In this work we will use $(\sigma, \beta, \rho) = (10, 8/3, 28)$ since at these points the attractor exhibits chaotic behavior [8]. A vector of initial conditions $(x_0, y_0, z_0) = (-12, 15, 38)$ in an interval of $T = [0, 20]$ was considered. After running 2000 realizations of the UQDES algorithm, the three-dimensional model was constructed.

Here, the blue trajectory represents the numerical solution and the light blue spectrum are the model realizations with uncertainty, noting that the perturbed trajectories represent a modification of the attractor orientation but do not change radically its overall behavior. On the other hand, a generalized polynomial chaos sampling was performed, with Legendre polynomials of order 4 that generated 50 basis functions fitted by Gaussian quadrature. Due to the geometry of the attractor, approximate it by this method can be complicated in initial

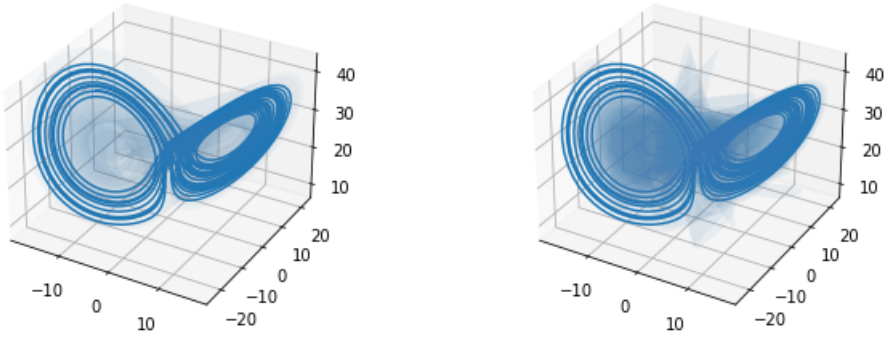

Figure 5.1: 3D reconstructed solutions for the Lorenz model.
Left, GP-Based vs. Numerical solutions. Right, GP+PCE vs. Numerical solutions.

conditions in which the dynamics of the attractor results in chaos. In fact, note that the calculated realizations become spare and depart from the expected trajectory as time goes by, presenting an increase in the standard deviation and therefore in the variance of the realizations.

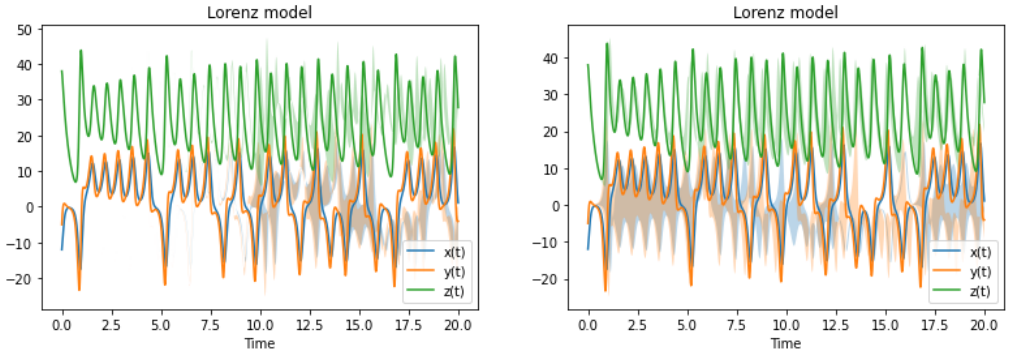

Figure 5.2: Individual solutions for Lorenz model.
Left, GP-Based vs. Numerical solutions. Right, GP+PCE vs. Numerical solutions.

Comparing the numerical solution with the sampling of the solutions with the GP-based method and with addition of the polynomial chaos expansion, we observe that the polynomial chaos simulation has a rather more unstable behavior with a clear variability in the functions $x(t)$ and $y(t)$, which is noticeable from the beginning of the time interval.

Taking the mean of the realizations to compute the performance measures, we can observe that a considerably higher error rate is present in the functions $x(t)$ and $y(t)$, according with what was observed in the plot of the trajectories. In general, polynomial chaos expansions did not help to estimate the model adequately, perhaps due to its complex structure [76].

Function	Technique	RMSE	SMSE	MSLL
x(t)	GP-Based	11.7409	5566.6898	8.5237
	GP+PCE	10.0168	9242.2935	9.5369
y(t)	GP-Based	13.1429	5578.0299	8.7529
	GP+PCE	11.2242	9166.5040	9.7348
z(t)	GP-Based	11.3804	5566.4833	8.4612
	GP+PCE	6.9450	2570.9264	6.7482

Table 5.1: Performance Measures for Lorenz model

After running each of the programs 10 times, we obtained an average of the execution time, where we can notice a considerable difference between the two programs. This may be due to the high sensitivity of the PCE-based methods to the chaotic dynamics of the attractor.

Technique	Execution Time (s)
GP-Based	3.6883
GP+PCE	13.2190

Table 5.2: Execution Time for Lorenz model.

5.2 Kermack-McKendrick SIR Model

For this coupled model, we will use the set of parameters $(\beta, \gamma) = (0.22, 0.1)$ and initial conditions $(S(0), I(0), R(0)) = (1, 0, 0)$ in an interval $T = [0, 200]$.

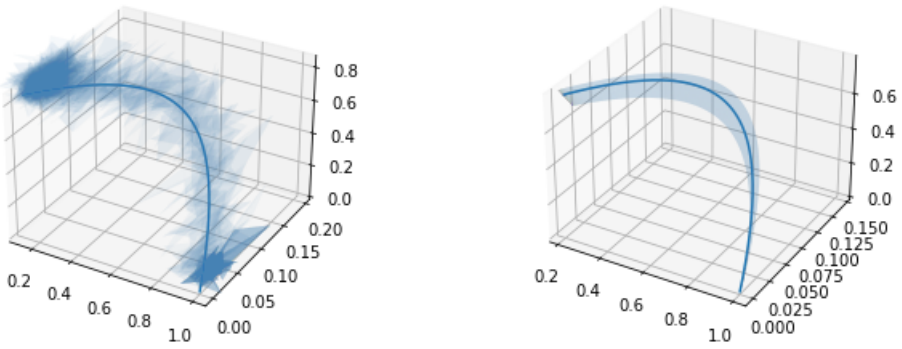

Figure 5.3: 3D reconstructed solutions for SIR model.
Left, GP-Based vs. Numerical solutions. Right, GP+PCE vs. Numerical solutions.

A wide range of disturbances can be observed in the estimation of the GP technique, with unstable and periodic trajectories, which are concentrated at the beginning of the time interval. On the other hand, the trajectories returned by adding the polynomial chaos fit are smoother and have a mean closer to the numerical estimated paths of the solution. The method was applied with Legendre polynomials of order 3 resulting in 50 basis functions. The effect of the variability of the trajectories has a better understanding in the three-dimensional projection of the model.

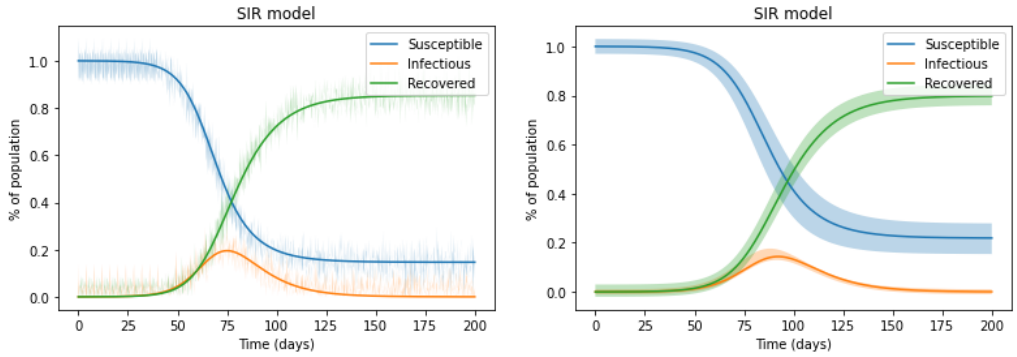

Figure 5.4: Individual solutions for SIR model.
Left, GP-Based vs. Numerical solutions. Right, GP+PCE vs. Numerical solutions.

On the other hand, the mean squared error rates do not differ considerably between methods, but the standardized error shows a lower error in the method based on GPs. This may occur because the mean of the realizations is closer to the original solution; however, the high variability of data must be considered. Although as the polynomial chaos increases the error is higher, the sampled functions are smoother which allows a more adequate prediction of the function at all its points.

Function	**Technique**	**RMSE**	**SMSE**	**MSLL**
S(t)	GP-Based	0.0325	8.1571	0.1750
	GP+PCE	0.1480	193.7416	0.1471
I(t)	GP-Based	0.0355	420.2386	1.7746
	GP+PCE	0.0403	736.1283	0.2806
R(t)	GP-Based	0.0337	8.8735	0.1035
	GP+PCE	0.1404	176.2106	0.1750

Table 5.3: Performance Measures for SIR model.

Finally, after ten runs, the average run time increases slightly with the addition of polynomial

chaos, which does not imply an unusual result given the distribution domain, the type of polynomial and the quadrature used.

Technique	Execution Time (s)
GP-Based	02.9691
GP+PCE	03.2970

Table 5.4: Execution Time for SIR model.

5.3 FitzHugh–Nagumo Model

For the simulations, the values of the parameters used in [77] will be used, which are $(a, b, \tau) = (0.5, 0.7, 0.8)$ and a current amount $I = 12.5$. In this study, we have considered the initial conditions $(v(0), w(0) = (0, 0)$ to run simulations in a interval time $T = [0, 200]$. For the implementation of polynomial chaos, 56 basis functions were generated from Legendre polynomials of degree 3, using the Gaussian quadrature rule.

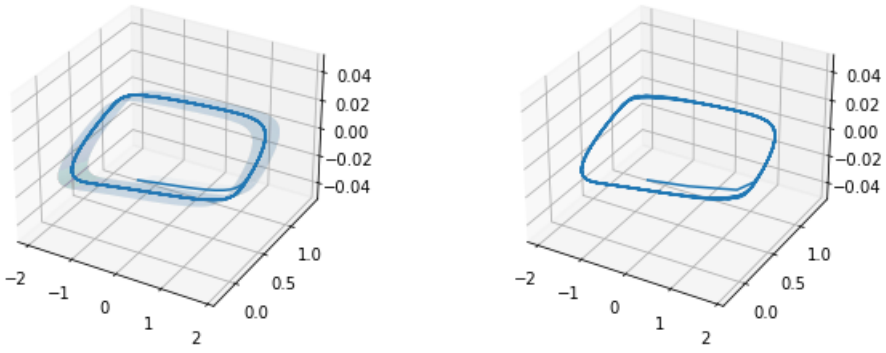

Figure 5.5: 3D reconstructed solutions for FitzHugh–Nagumo model.
Left, GP-Based vs. Numerical solutions. Right, GP+PCE vs. Numerical solutions.

A good overall estimation can be observed using the GP-based algorithm, where the distance between the simulated trajectories and the numerical solution is small. Moreover, the simulations follow very closely the shape of the numerical trajectory so we can affirm that its variability is low. On the other hand, the addition of the chaos component considerably improves the prediction, resulting in trajectories that overlap the numerical solution.

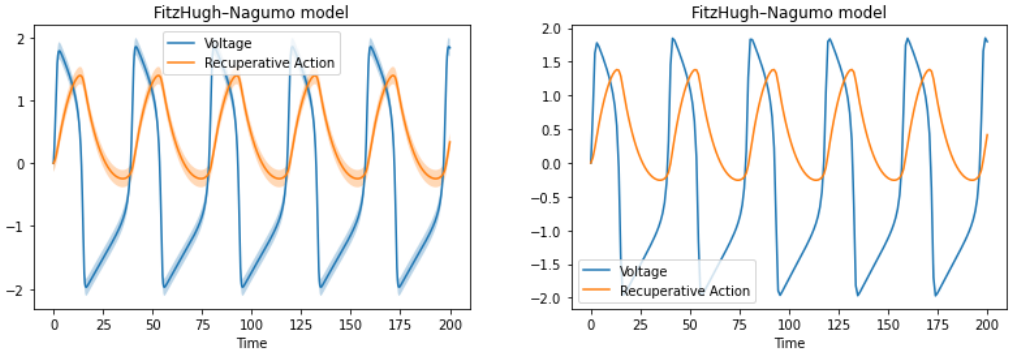

Figure 5.6: Individual solutions for FitzHugh–Nagumo model.
Left, GP-Based vs. Numerical solutions. Right, GP+PCE vs. Numerical solutions.

The above-mentioned observations are confirmed by the performance measures, which show a much lower error than in the rest of the models. Both methods yielded very small error rates, indicating that in this model there is no substantial modification of the solutions when simulations are generated.

Function	**Technique**	**RMSE**	**SMSE**	**MSLL**
v(t)	GP-Based	$3.5874e^{-17}$	$1.4399e^{-30}$	2.4187
	GP+PCE	0.1153	1.4772	2.4335
w(t)	GP-Based	$1.0268e^{-17}$	$6.3392e^{-31}$	0.7372
	GP+PCE	0.0310	0.5823	0.7359

Table 5.5: Performance Measures for FitzHugh-Nagumo model.

In a similar way to the SIR model, by calculating the mean of the execution time after ten runs of each algorithm for FitzHugh-Nagumo model, an increase in run time is found when implementing the polynomial chaos expansion addition.

Technique	**Execution Time (s)**
GP-Based	03.0034
GP+PCE	03.4850

Table 5.6: Execution Time for FitzHugh-Nagumo model.

Chapter 6

Conclusions

In real life, uncertainty is present in almost all systems modeling physical, biological, chemical and even social and economic behaviors. For this reason, quantifying its effect within the estimation of the solutions of these models helps experts to predict divergences that could occur in the presence of perturbations. This can cause undesired effects, but due to the computational complexity that it usually has, it is ignored. In this work, this problem is addressed by using a GP-based algorithm and its combination with recently studied components, such as polynomial chaos expansions.

To check the outcome of both techniques, they were tested on models which can be challenging case studies due to their dynamics. The algorithm proposed by Chkrebtii showed a mean relatively close to the numerical solution in all the cases, but the randomness of the samplings resulted in the existence of several sparse trajectories. This shows that the uncertainty rate in the estimation of the solutions is significant, especially in the SIR model where constant perturbations are shown.

On the other hand, the polynomial chaos addition yielded good results in terms of similarity to the mean of the GP-based simulations, as it was expected, and approached toward the numerical solutions. However, simulated trajectories of the Lorenz model depart from the solutions as time progressed, which may suggest that this technique is sensitive to the chaotic structure of the model. Finally, a comparison between the two approaches versus the numerical solution given by Python shows a better estimation when polynomial chaos is added in

two systems. As a consequence, the error rate is lower, specially in the FitzHugh-Nagumo model.

However, the tests on these models do not represent a decisive opinion about the contrast of both methods in terms of performance, so future studies on other systems of chaotic behavior are suggested that may provide a better approach to the interpretation of the results of the techniques. On the other hand, it is recommended to make modifications in the basis functions of the polynomial chaos, such as changing the type of polynomial, their order or their quadrature fitting, in order to have a broaden perspective of the results obtained.

Bibliography

[1] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, 01 2009.

[2] O. Chkrebtii, D. Campbell, M. Girolami, and B. Calderhead, "Bayesian solution uncertainty quantification for differential equations," *Bayesian Analysis*, vol. 11, 06 2013.

[3] L. Berliner, "Likelihood and bayesian prediction of chaotic systems," *Journal of The American Statistical Association - J AMER STATIST ASSN*, vol. 86, pp. 938–952, 12 1991.

[4] J. Ramsay, G. Hooker, D. Campbell, and J. Cao, "Parameter estimation for differential equations: A generalized smoothing approach," *Journal of the Royal Statistical Society Series B*, vol. 69, pp. 741–796, 11 2007.

[5] A. Overstall, D. Woods, and B. Parker, "Bayesian optimal design for ordinary differential equation models with application in biological science," *Journal of the American Statistical Association*, vol. 115, 04 2019.

[6] W. Oberkampf and C. Roy, *Verification and Validation in Scientific Computing*. Cambridge University Press, 01 2010.

[7] P. Conrad, M. Girolami, S. Särkkä, A. Stuart, and K. Zygalakis, "Probability measures for numerical solutions of differential equations," 06 2015.

[8] E. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Science*, vol. 20, pp. 130–141, 03 1963.

[9] D. Emile, R. Maritz, and J. Munganga, "Some properties of kermack – mckendrick epidemic model with fractional derivatives and nonlinear incidence," *Advances in Difference Equations*, 09 2014.

[10] W. E. Sherwood, *FitzHugh–Nagumo Model*. New York, NY: Springer New York, 2013, pp. 1–11. [Online]. Available: https://doi.org/10.1007/978-1-4614-7320-6_147-1

[11] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin, *Bayesian Data Analysis*, 11 2013.

[12] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, pp. 35–45, 01 1960.

[13] Z. Chen and E. N. Brown, "State space model," *Scholarpedia*, vol. 8, no. 3, p. 30868, 2013, revision #189565.

[14] W. Sternberg and M. Deisenroth, *Identification of Gaussian Process State-Space Models*, 2017.

[15] A. Dore, M. Pinasco, and C. Regazzoni, *Multi-Modal Data Fusion Techniques and Applications*, 05 2009, pp. 213–238.

[16] S. Särkkä, *Bayesian Filtering and Smoothing*, 10 2013.

[17] Z. Chen, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics*, vol. 182, 01 2003.

[18] A. Doucet, N. Freitas, K. Murphy, and S. Russell, "Sequential monte carlo methods in practice," 01 2013.

[19] J.-L. Wang, J.-M. Chiou, and H.-G. Mueller, "Review of functional data analysis," 07 2015.

[20] S. Ullah and C. Finch, "Applications of functional data analysis: A systematic review," *BMC medical research methodology*, vol. 13, p. 43, 03 2013.

[21] B. W. S. a. J. O. Ramsay, *Functional Data Analysis*, 2nd ed., ser. Springer Series in Statistics. Springer-Verlag New York, 2005.

[22] J. Ramsay and C. Dalzell, "Some tools for functional data analysis," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 53, pp. 539–561, 07 1991.

[23] R. P. Dobrow, *Introduction to stochastic processes with R.* Hoboken, New Jersey: John Wiley Sons, Inc., 2016.

[24] E. Solak, R. Murray-Smith, W. Leithead, D. Leith, and C. Rasmussen, "Derivative observations in gaussian process models of dynamic systems," *Appear Adv Neural Inf Process Syst*, vol. 16, 02 2003.

[25] D. Barber and A. Cemgil, "Graphical models for time-series," *Signal Processing Magazine, IEEE*, vol. 27, pp. 18 – 28, 12 2010.

[26] S. Olofsson, "Probabilistic feature learning using gaussian process auto-encoders," 2016.

[27] O. Martin, *Bayesian Analysis with Python: Introduction to statistical modeling and probabilistic programming using PyMC3 and ArviZ, 2nd Edition.* Packt Publishing, 2018. [Online]. Available: https://books.google.com.ec/books?id=1Z2BDwAAQBAJ

[28] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 08 2018.

[29] A. Solin, "Stochastic differential equation methods for spatio-temporal gaussian process regression," pp. 68 + app. 72, 2016. [Online]. Available: http://urn.fi/URN:ISBN:978-952-60-6711-7

[30] C. Rasmussen and C. Williams, *Gaussian Process for Machine Learning*, 01 2006.

[31] M. Gibbs, *Bayesian Gaussian Processes for Regression and Classification.* University of Cambridge, 1997.

[32] R. van Handel, "Stochastic calculus, filtering and stochastic control," 2007.

[33] K. K. Sepahvand, S. Marburg, and H.-j.hardtke, "Uncertainty quantification in stochastic systems using polynomial chaos expansion," *International Journal of Applied Mechanics*, vol. 02, 04 2012.

[34] D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, 12 2010.

[35] I. Elishakoff, R. Haftka, and J. Fang, "Structural design under bounded uncertainty—optimization with anti-optimization," *Computers Structures*, vol. 53, pp. 1401–1405, 12 1994.

[36] T. Crestaux, O. Le Maître, and J.-M. Martinez, "Polynomial chaos expansion for sensitivity analysis," *Reliability Engineering System Safety*, vol. 94, pp. 1161–1172, 05 2009.

[37] G. E. Andrews, R. Askey, and R. Roy, *Special Functions*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1999.

[38] S. Venkateshan and S. Prasanna, *Computational Methods in Engineering*, 12 2013.

[39] J. Booker and T. Ross, "An evolution of uncertainty assessment and quantification," *Scientia Iranica*, vol. 18, p. 669–676, 06 2011.

[40] A. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society. Series B*, vol. 30, 11 1967.

[41] W. Feller, *An Introduction to Probability Theory and Its Applications*. Wiley, January 1968, vol. 1. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20{&}path=ASIN/0471257087

[42] A. O'Hagan, "Curve fitting and optimal design for prediction," *Journal of the Royal Statistical Society. Series B*, vol. 40, 09 1978.

[43] R. Neal, "Regression and classification using gaussian process priors," *Bayesian Statistics*, vol. 6, 01 1999.

[44] I. Bilionis and N. Zabaras, *Bayesian Uncertainty Propagation Using Gaussian Processes*, 06 2017, pp. 555–599.

[45] J. Skilling, *Bayesian Solution of Ordinary Differential Equations*. Dordrecht: Springer Netherlands, 1992, pp. 23–37. [Online]. Available: https://doi.org/10.1007/978-94-017-2219-3_2

[46] J. Glimm and D. Sharp, "Prediction and the quantification of uncertainty," *Physica D: Nonlinear Phenomena*, vol. 133, no. 1-4, pp. 152–170, 1999.

[47] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, "Introduction to adaptive methods for differential equations," *Acta numerica*, vol. 4, pp. 105–158, 1995.

[48] K. Herzog, M. D. Morris, and T. Mitchell, "Bayesian approximation of solutions to linear ordinary differential equations," Oak Ridge National Lab., TN (USA), Tech. Rep., 1990.

[49] S. Dutta and A. Gandomi, *Design of experiments for uncertainty quantification based on polynomial chaos expansion metamodels*, 10 2019.

[50] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari, "Gpstuff: Bayesian modeling with gaussian processes," *The Journal of Machine Learning Research*, vol. 14, pp. 1175–1179, 04 2013.

[51] V. Vyshemirsky and M. Girolami, "Bayesian ranking of biochemical system models," *Bioinformatics (Oxford, England)*, vol. 24, pp. 833–9, 04 2008.

[52] R. Adams, I. Murray, and D. Mackay, "Nonparametric bayesian density modeling with gaussian processes," 12 2009.

[53] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank gaussian process regression," *Statistics and Computing*, vol. 30, 03 2020.

[54] M. Renardy, T.-M. Yi, D. Xiu, and C.-S. Chou, "Parameter uncertainty quantification using surrogate models applied to a spatial model of yeast mating polarization," *PLOS Computational Biology*, vol. 14, p. e1006181, 05 2018.

[55] B. Oparaji, "Robust surrogate models for uncertainty quantification and nuclear engineering applications," Ph.D. dissertation, 11 2017.

[56] B. Sudret, *Polynomial chaos expansions and stochastic finite element methods*, 12 2014, p. 624.

[57] Y. Li, W. Xiao, and P. Wang, "Uncertainty quantification of artificial neural network based machine learning potentials," 11 2018, p. V012T11A030.

[58] A. Khosravi, S. Nahavandi, and D. Creighton, "Quantifying uncertainties of neural network-based electricity price forecasts," *Applied Energy*, vol. 112, pp. 120–129, 12 2013.

[59] J. Yao, W. Pan, S. Ghosh, and F. Doshi-Velez, "Quality of uncertainty quantification for bayesian neural network inference," 06 2019.

[60] A. Matthews, M. Rowland, J. Hron, R. Turner, and Z. Ghahramani, "Gaussian process behaviour in wide deep neural networks," 04 2018.

[61] W. Huang, D. Zhao, F. Sun, H. Liu, and E. Chang, "Scalable gaussian process regression using deep neural networks," in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI'15. AAAI Press, 2015, p. 3576–3582.

[62] H. M. D. Kabir, A. Khosravi, S. Nahavandi, and D. Srinivasan, "Neural network training for uncertainty quantification over time-range," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 11 2020.

[63] H. M. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE Access*, vol. PP, 06 2018.

[64] R. Tripathy and I. Bilionis, "Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification," *Journal of Computational Physics*, vol. 375, 02 2018.

[65] J. Riihimäki and A. Vehtari, "Gaussian processes with monotonicity information," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 645–652, 01 2010.

[66] J. Butcher, "Numerical methods for ordinary differential equations," pp. i–xxiv, 08 2016.

[67] J. Bram and A. Yaglom, "An introduction to the theory of stationary random functions," *Mathematics of Computation*, vol. 17, p. 468, 10 1963.

[68] M. Schober, S. Särkkä, and P. Hennig, "A probabilistic model for the numerical solution of initial value problems," *Statistics and Computing*, vol. 29, 01 2019.

[69] H. Kersting and P. Hennig, "Active uncertainty calibration in bayesian ode solvers," 05 2016.

[70] F. Tronarp, H. Kersting, S. Särkkä, and P. Hennig, "Probabilistic solutions to ordinary differential equations as nonlinear bayesian filtering: a new perspective," *Statistics and Computing*, vol. 29, 11 2019.

[71] A. Lindholm, A. Solin, S. Särkkä, and T. Schön, "Computationally efficient bayesian learning of gaussian process state space models," 06 2015.

[72] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 09 2020.

[73] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. [Online]. Available: https://rdcu.be/b08Wh

[74] J. Feinberg and H. P. Langtangen, "Chaospy: An open source tool for designing methods of uncertainty quantification," *Journal of Computational Science*, vol. 11, 08 2015.

[75] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[76] E. Blanchard, C. Sandu, and A. Sandu, "A polynomial chaos-based kalman filter approach for parameter estimation of mechanical systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 132, 11 2010.

[77] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical Journal*, vol. 1, no. 6, pp. 445–466, 1961. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0006349561869026

Appendices

A.1 UQDES Algorithm Code

```
# 1. Libraries

import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import math
import random
import time

from scipy.integrate import odeint
from numpy.matlib import repmat
from scipy import special
from sklearn.metrics import r2_score

# 2. Probabilistic Solver
# 2.1. Convolution Kernels

def QQ1d_se(u1, v1, w, a, b):   # QQ Squared Exponential
  u1 = u1.reshape(len(u1),1)
  v1 = v1.reshape(len(v1),1)

  u = repmat(u1,1,len(v1))
  v = repmat(np.transpose(v1),len(u1),1)

  qqt = (np.pi*w*w*(u-a)*special.erf((u-a)/(2*w)))\
  +(2*np.sqrt(np.pi)*w*w*w*np.exp(-((u-a)*(u-a))/(4*w*w)))\
  -(np.pi*w*w*(v-u)*special.erf((v-u)/(2*w)))\
  -(2*np.sqrt(np.pi)*w*w*w*np.exp(-((v-u)*(v-u))/(4*w*w)))\
```

```
    +(np.pi*w*w*(v-a)*special.erf((v-a)/(2*w)))\
    +(2*np.sqrt(np.pi)*w*w*w*np.exp(-((v-a)*(v-a)))/(4*w*w))\
    -(2*np.sqrt(np.pi)*w*w*w)

    return qqt

def RR1d_se(u1, v1, w, a, b):    # RR Squared Exponential
    u1 = u1.reshape(len(u1),1)
    v1 = v1.reshape(len(v1),1)

    u = repmat(u1,1,len(v1))
    v = repmat(np.transpose(v1),len(u1),1)

    rrt = np.exp(-((u-v)*(u-v))/(4*w*w))*np.sqrt(np.pi)*w

    return rrt

def RQ1d_se(u1, v1, w, a, b):    # QR Squared Exponential
    u1 = u1.reshape(len(u1),1)
    v1 = v1.reshape(len(v1),1)

    u = repmat(u1,1,len(v1))
    v = repmat(np.transpose(v1),len(u1),1)

    rqt = (np.pi*w*w*special.erf((v-u)/(2*w)))+(np.pi*w*w*special.erf((u-a)
        /(2*w)))

    return rqt

def QR1d_se(u1, v1, w, a, b):
    qrt = np.transpose(RQ1d_se(v1,u1,w,a,b))

    return qrt

# 2.2. UQDES Algorithm

def uqdes(sspan, nsolves, N, kernel, alpha, u0, theta):
    tic = time.time() #initial execution time

    M = np.size(u0)
    B = nsolves
    s = np.linspace(sspan[0], sspan[1], N)
    t = s
    ds = s[1]-s[0]
    lmbd = 0.5*ds
```

```
u0 = np.transpose(u0)

if kernel == 'sqexp':
  trim = N
else:
  print('Unknown kernel -- try again')

uensemble = np.tile(u0,(B,N,1))

f = np.tile(u0,[B,1,1])
f[0] = lorentz(uensemble[0,0],s[0],theta)

m_deriv_svec = np.tile(np.zeros(len(u0)),[B,N,1])
m_state_svec =  uensemble + np.multiply(m_deriv_svec, np.tile(s.reshape(N
    ,1),[B,1,len(u0)]))

C_deriv_ssmat = RR1d_se(s,s,lmbd,sspan[0],sspan[1])/alpha
C_state_ssmat = QQ1d_se(s,s,lmbd,sspan[0],sspan[1])/alpha
C_cross1_ssmat = QR1d_se(s,s,lmbd,sspan[0],sspan[1])/alpha

kinv = 1/(C_deriv_ssmat[0,0])
f_diff = kinv*(f[:,0,:]-m_deriv_svec[:,0,:])
randnNums  = np.random.rand(B,N,M)
counter = 0

for n in np.arange(N):
  if n > 0:
    ind = np.arange(N)
    endind = np.arange(N)

    m_state_svec[:,endind,:] = m_state_svec[:,endind,:]+(repmat(np.array(
        C_cross1_ssmat[endind,0]).reshape(N,1),1,3)*repmat(f_diff
        [:,:],400,1)) m_deriv_svec[:,ind,:] = m_deriv_svec[:,ind,:]+(repmat
        (np.array(C_deriv_ssmat[ind,0]).reshape(N,1),1,3)*repmat(f_diff
        [:,:],400,1))

    C_state_ssmat[endind,endind] = C_state_ssmat[endind,endind]-kinv*
        C_cross1_ssmat[endind,n]*np.transpose(C_cross1_ssmat[endind,n])
    C_cross1_ssmat[endind,ind] = C_cross1_ssmat[endind,ind]-kinv*
        C_cross1_ssmat[endind,ind]*C_deriv_ssmat[n,ind]
    C_deriv_ssmat[ind,ind] = C_deriv_ssmat[ind,ind]-kinv*C_deriv_ssmat[ind
        ,n]*C_deriv_ssmat[n,ind]

    uensemble[:,n,:] = m_state_svec[:,n,:] +
```

```
        randnNums[:,n-1,:]*np.sqrt(C_state_ssmat[n][n]);
        kinv = 1/(C_deriv_ssmat[n][n]+C_deriv_ssmat[n-1][n-1])
        f_diff = kinv*(lorentz(uensemble[0,0],s[n],theta)-m_deriv_svec[:,n,:])

    toc = time.time() #final execution time
    logfntime= toc-tic
    print('This algorithm took',logfntime, 'sec to run completely.')
    return uensemble, t
```

A.2 Polynomial Chaos Expansion Addition Code

```
# Libraries

import chaospy
import time

# Polynomial Expansions

polynomial_order = 3
polynomial_expansion = chaospy.generate_expansion(polynomial_order,
    distribution)
polynomial_expansion[:5].round(5)

# Quadrature, Evaluations and Weights

quadrature_order = 8
abscissas, weights = chaospy.generate_quadrature(quadrature_order,
    distribution, rule="gaussian") #distribution is given by the joint
    distribution of the model constant parameters

evaluations = [model_solver(abscissa) for abscissa in abscissas.T] #
    model_solver is given by uensemble, an output of A.1

# Model Fitting

def fitting(polynomial_expansion, abscissas, weights, evaluations):
  tic = time.time() #initial execution time

  model_approx = chaospy.fit_quadrature(polynomial_expansion, abscissas,
      weights, evaluations)
```

```
    expected = chaospy.E(model_approx, distribution)
    std = chaospy.Std(model_approx, distribution)

    toc = time.time() #final execution time
    logfntime= toc-tic
    print('This algorithm took',logfntime, 'sec to run completely.')
    return expected, std
```