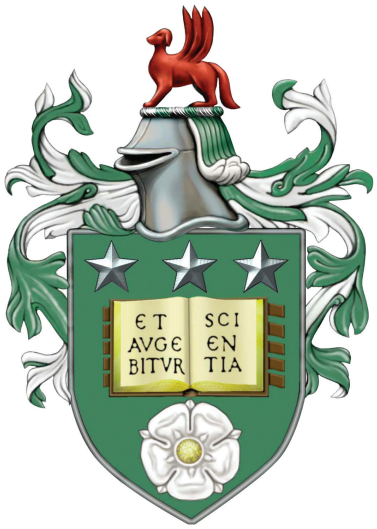
Gaussian Process Emulation: Theory and Applications to the Problem of Past Climate Reconstruction



Dario Domingo

The University of Leeds
School of Mathematics

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

October 2019

Intellectual Property Statement

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

To my parents
who have always supported my studies and ambitions, in spite
of the geographical distance that these have imposed.

A Giovanna e Nino
che mi hanno sempre sostenuto nel perseguire le mie ambizioni,
nonostante la distanza che queste hanno imposto.

Acknowledgements

There are several people whom I wish to thank for their contribution to my PhD and, as a result, to this work. First and foremost my supervisor, Jochen Voss. Jochen, you have been a precious guide for me throughout the PhD. Thanks for giving the right piece of advice when asked for, for believing in my potential and always encouraging my own development, but also for listening to and sharing with me ideas of any sort. Two special persons also deserve my heartfelt thanks: Alan Haywood and Aisling Dolan, my co-supervisors from the School of Earth and Environment. Alan, Aisling, you have both been remarkably kind to me. I have always been able to rely on your explanations, your support and most importantly your affection, which I genuinely reciprocate. My gratitude goes as well to Louise Sime. Almost by chance we started a collaboration three years ago, which I could not wish to be more pleasant and successful, and which has taught me a lot about our planet and its precious ice sheets.

Thanks Harry (Dowsett), your prompt and detailed emails have been most helpful to answer my numerous questions on geological records, and your efforts to scrutinise old archives most appreciated. Thanks John Paul (Gosling), for the mathematical conversations we had in your office, for reading parts of this thesis and especially for the support and advice you have given me throughout the job-search period. Similarly, thanks a lot Elena (Issoglio), for your support and for always enjoyable conversations and valuable exchange of views we had on the most diverse topics.

There are also several people who may not have directly contributed to this work, but whom I nonetheless feel part of it. It would be impossible to list them all. They are the numerous friends I made in Leeds, with whom I shared several experiences, trips and countless dinners. To them all goes my gratitude for making this a great journey. Last but not least, a special thanks to Anastasia, for understanding my last months of hard work and never complaining of the little time this has left during most days.

Abstract

The dynamics of complex systems are commonly explored via the use of computer simulators. To ensure an understanding of the phenomena they model, simulators are usually run at a sequence of inputs, to explore different scenarios. This, however, often requires a prohibitive amount of time and computational resources. In such a case, the Bayesian framework of Gaussian process emulation allows to build a fast and reliable statistical surrogate of the simulator, called an emulator. This provides not only predictions of the simulator outputs, but also information on the uncertainty of these predictions. This work investigates the framework of Gaussian process emulation, and provides two separate examples of application to climate problems.

In Part I of the thesis, Gaussian process emulation is introduced and investigated in depth. We employ a formal probabilistic setting, allowing us to see the derivation as an example of Bayesian analysis in an infinite-dimensional space, and to recover the formulas commonly used as a limit case. Further analyses are carried out, and the case of a chaotic simulator is investigated. In relation to the problem of emulating climate simulators, we also propose a dimension-reduction technique that accounts for the Earth's spherical geometry.

In Part II of the thesis, we employ the emulation framework to tackle problems of past climate reconstruction, key to understanding the dynamics and potential consequences of current global warming. In a first application, we explore the mismatch between simulated mid-Pliocene ocean temperatures and geological records. By sampling from the emulator trajectories, we reproduce the way records are extracted and account for orbitally-induced changes in temperature. In a second application, we explore the morphology of the Greenland ice sheet during the last Interglacial, to locate areas prone to melting under warm temperatures. The context provides an example of non-standard emulation setting, where the emulator input space consists of ice shapes.

Preface

This thesis summarises the work I have carried out during my PhD. I feel lucky to have had the opportunity to combine what is a strong academic passion, the one of mathematics, with my personal interest in climate and in the current climate crisis that we are witnessing, which is now being brought more and more to the attention of politicians and decision-makers. At the beginning of my PhD, I understood that an effective way we have to predict the consequences of the current warming is, however strange this may sound at first, to study the past. In particular, to study periods in Earth's history that appear (and in fact are) temporally remote, but whose climate shares much more than we think with the one we have, now, unnaturally caused.

In this challenge, rigorous statistics and mathematics play a prominent role. I here provide the details of my personal experience in this context: using the statistical framework of Gaussian process emulation to draw inference that is both statistically sound and of relevance to the applied problem. While the first aim calls for direct confrontation with, and feedback from, experts in the field (climate scientists in my case), the second is usually in the hands of the statistician or mathematician. I believe that a sound theoretical investigation is key to achieve robust conclusions. The way this work is structured and developed reflects my interest for both a mathematically sound setting and for the relevance that this can have to efficiently tackle problems of collective interest. My wish is that the reader, whether motivated by a purely statistical and mathematical interest, by an interest in climate and in the current crisis, or by both, may find the coming chapters pleasant and enjoy their reading.

Leeds,
October 2019

Dario Domingo

Contents

List of Abbreviations

CC:	Camp Century (Greenland ice-core site).
DMC:	Data-Model Comparison.
GCM:	General Circulation Model.
GP:	Gaussian Process.
GrIS:	Greenland Ice Sheet.
HadCM3:	Hadley Centre Coupled Model - version 3.
HPC:	High Performance Computing.
IPCC:	Intergovernmental Panel on Climate Change.
kya:	Thousand of Years ago.
LHS:	Left-Hand Side.
LIG:	Last Interglacial.
LOOCV:	Leave-One-Out Cross-Validation.
MAP:	Maximum a Posteriori.
MS:	Mean-Square.
NIG:	Normal-Inverse-Gamma.
PC(A):	Principal Component (Analysis).
PI:	Pre-Industrial.
PRISM:	Pliocene Research, Interpretation, and Synoptic Mapping.
RHS:	Right-Hand Side.
SST:	Sea Surface Temperature.
SVD:	Singular Value Decomposition.
UQ:	Uncertainty Quantification.
WPA:	Warm Peak Average.

List of Notation

The followings list provides a reference to some of the notation used throughout this work. For easier reference, the list is divided into three classes.

Sets and General Mathematical Notation	
$\mathbb{N}$	Set of positive natural numbers: $\mathbb{N} = \{1, 2, 3, \dots\}$.
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}^d$	Set of real vectors of length d.
$\mathbb{R}^{d_1 \times d_2}$	Set of real matrices of dimension $d_1 \times d_2$.
$\mathbf{1_d}$	Vector in $\mathbb{R}^d$ whose components are all equal to 1.
$\boldsymbol{I_d}$	Identity matrix of order d.
$\mathbb{1}_A(\cdot)$	Indicator function of the set A: $\mathbb{1}_A(x) = 1$ if $x \in A$, $\mathbb{1}_A(x) = 0$ otherwise.
S^2	Unit sphere in $\mathbb{R}^3$: $S^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}$.
Probability-Related Notation	
$\mathbb{P}(A)$	Probability of the set A.
$\mathbb{E}[X]$	Expectation of a random variable (or vector) X.
$X \sim \nu$	The random variable/vector X has distribution given by ν.
$N(\mu, \sigma^2)$	Univariate Normal distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \geq 0$.
$N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^q$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{q \times q}$.
$NIG(\boldsymbol{b}, \boldsymbol{B}, a, s)$	Normal-Inverse-Gamma distribution with parameters $\boldsymbol{b}, \boldsymbol{B}, a, s$ (see page 48).
$\mathcal{GP}\big(m(\cdot), v(\cdot, \cdot)\big)$	Gaussian process with mean function $m(\cdot)$ and covariance function $v(\cdot, \cdot)$.

Emulation-Related Notation

$\mathcal{P} \subseteq \mathbb{R}^p$	Input space of emulator and simulator.
$p \in \mathbb{N}$	Dimension of emulator/simulator input space.
$\boldsymbol{x} \in \mathcal{P}$	Generic input to emulator/simulator.
$\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \in \mathcal{P}$	Design points (inputs at which the simulator is run).
$n \in \mathbb{N}$	Number of design points.
$\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} \in \mathcal{P}$	Collection of k generic points in $\mathcal{P}$ at which the emulator is evaluated.
$y \in \mathbb{R},\ \boldsymbol{y} \in \mathbb{R}^s$	Univariate (*e.g.*, Chapter 2) or multivariate (*e.g.*, Chapter 3) simulator outputs.
$s \in \mathbb{N}$	Dimension of space ($\mathbb{R}^s$) on which PCA is performed before GP emulation can be applied.

Introduction to the Thesis

This thesis explores the statistical framework of Gaussian process (GP) emulation and provides details of two applications to past climate reconstruction problems. Accordingly, the work is divided into two parts. In Part I, a thorough discussion of the theory of GP emulation is carried out and further developments within this and related fields are proposed. In Part II, two climate problems are introduced and investigated in detail, in the light of both applied considerations and the theory previously laid out.

While, especially in Part I, some of the sections necessarily delve into the mathematical theory or technicalities, it has been my (the author's) effort to make any reasoning or explanation as clear as possible. Where appropriate and possible, I have used illustrations and diagrams to accompany theoretical explanations. In addition, throughout the work, the reader will find infoboxes, like the following one.

Infoboxes

An infobox may add an historical note that is contextually relevant, clarify the notation used, discuss a technical caveat, or more. The title will provide a guide. The aim of all infoboxes is to concisely present information that favours the understanding of the context or gives a slightly different perspective on it, while not being strictly necessary to understand the rest of the section or chapter.

The two parts into which the thesis is divided should not be considered strictly separate entities. The theoretical investigations of Part I are motivated by the problems arising in applied contexts, and allow to gain a level of insight into these which would be otherwise difficult to achieve. On the other side, the context of each of the two problems tackled in Part II, both arising in relation to the current climate change issue, plays a

primary role in guiding the statistical and mathematical choices which are undertaken in this part of the thesis. The climate context also provides the starting point to develop methodologies which have wider applicability and their own interest from a purely mathematical point of view.

Some remarks about conventions used in this work are as follows. I generally devote the first section of each chapter to introduce in plain language the problem dealt with, be this of pure or applied nature. Especially in Part II, this aims to introduce the reader to potentially unfamiliar settings, in order to favour a solid understanding of the context and motivations behind the work that follows. Moreover, unless otherwise stated, throughout the work I use lowercase plain letters to denote scalars, lowercase bold letter to denote vectors, and uppercase bold letters to denote matrices. As an example, consider the following:

$$x \in \mathbb{R}, \quad \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{X} \in \mathbb{R}^{p \times q}, \qquad n, p, q \geq 2\,.$$

The same will hold for functions, whose values are respectively scalars, vectors, or matrices. Finally, the i^{th} component of a vector $\boldsymbol{x}$ is denoted as x_i.

I would like to conclude with a remark. Unlike the Preface and this very first part, most of the coming work, mainly of descriptive character, is written in first person plural. This choice conforms to the style widely used to report scientific research, where the pronoun "we" may be used with the purpose of including the reader, or on behalf of the whole research community. I am nonetheless the sole author of this work and, as such, I am as well the only person who is to be blamed for any inaccuracy or incorrectness that this work may contain.

Thesis Outline

In the rest of this work, each chapter is introduced by its own abstract. An overview of the structure of the thesis is as follows.

In Part I of the thesis, Chapter 1 introduces general mathematical and statistical tools, at the basis of GP emulation: in particular stochastic and Gaussian processes, and the ideas underlying Bayesian modelling. Chapter 2 is devoted to the emulation framework. While most of the chapter's results are soundly established in the relevant literature, the way the framework is presented and the results are derived, within the probability formalism introduced in the preceding chapter, often departs from the standard approach. Part I terminates with Chapter 3, where a classical dimension-reduction technique (PCA) is adapted to a the spherical setting. This has potential relevance for a number of statistical applications.

Part II of the thesis approaches two problems of climate reconstruction. In Chapter 4, the period we focus on is the mid-Pliocene (about 3.3 to 3 million years ago). This represents the last time in Earth's history with atmospheric carbon dioxide concentrations comparable to today, and global temperatures warmer than today. We investigate the mismatch between mid-Pliocene ocean temperature simulated via complex climate simulators, and temperature reconstructions from geological records. The emulation framework allows us to replace the expensive simulator with a reliable and fast-to-run statistical model, which can be used as its surrogate. In Chapter 5, we employ the framework of GP emulation, together with information available from Greenland ice cores, to study the shape and extent of the Greenland ice sheet during the last Interglacial (approximately 125 thousand years ago). At the time, polar temperatures are estimated to have been up to 4°–5°C warmer than today. Besides being of primary interest within the climate community, due to the drastic consequences produced by the Greenland ice sheet's current melting, the topic provides the opportunity to develop emulation in a non-standard setting, specifically on ice shapes.

The Matlab code that I have developed to tackle the problems in Part II can be downloaded from https://github.com/dariod89. Part of the code is reported in Appendix.

Part I

Theory of Gaussian Process Emulation

1. Bayesian Statistics and Gaussian Processes

Abstract: The aim of the chapter is to introduce concepts and tools that are of particular importance in the treatment of Gaussian process emulation, and of general relevance within the fields of statistics and probability. Specifically, after providing the context and motivation behind the introduction of emulators, we discuss: i) the Bayesian framework for statistical inference; ii) stochastic and Gaussian processes. We report and discuss classical results linking the regularity of a stochastic process to the one of its covariance function, both in mean square and pathwise, and introduce the concept of correlation lengths. The exposition also allows to set the notation and terminology used in the next chapter, where the framework of Gaussian process emulation is investigated in detail.

1.1. Motivation for the Introduction of Emulators

1.1.1. The Use of Computer Simulators in Science

Mathematical models are employed to study the dynamics of various phenomena, for which direct experimentation is too complex or impossible to perform. The primary aim of a model is to allow to gain a deeper insight into the system that it describes. While only representing an approximation of the real phenomenon, most models of practical interest incorporate a level of complexity that makes purely theoretical investigations of their properties unfeasible. For this reason, scientists resort to numerical approximations of the model, implemented in the form of computer code. These are known as computed models, or simulators.

Nowadays, computer simulators are used virtually in every area of science. Craig et al. [1997] provide an example of application to hydrocarbon reservoirs; Vernon et al. [2010] employ a complex simulator to study the large-scale physical phenomenon of Galaxy formation and evolution; Zhou et al. [2010] model crowd behaviour and dynamics; Alden et al. [2013] present a simulator designed for lymphoid tissue organogenesis modelling, alongside other biological modelling tasks; Kavetski [2019] focus on simulation of hydrological systems. A vast literature concerning the use of computer simulators in weather and climate modelling is also available (see for example Rougier and Goldstein [2014], Tran et al. [2016], and also Menemenlis et al. [2005] for an illustration of the use of a NASA supercomputer in ocean climate research); we will expand more on this in Part II of this work.

As already pointed out, computer simulators are characterised by a complex structure, which accounts for the interactions between processes developing in different compartments or components of the system. Such a high complexity necessarily introduces a large number of parameters in the simulator. Some of these are parameters of the mathematical model, whose "best" value may be unknown. Others represent external influences on the system, which tailor the behaviour of the latter to a specific instance: these are known as forcing parameters. We can here denote by $\boldsymbol{x}$ the collection of all the model and forcing parameters that can be varied between different runs of the simulator, and by $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ the state of the simulated system corresponding to the

choice of input parameters $\boldsymbol{x}$. This explicitly allows us to view a simulator as a mathematical function, associating outputs $\boldsymbol{y}$ to inputs $\boldsymbol{x}$.

To ensure a robust inference on the problem of interest, the simulator must usually be run at a sequence of inputs, with the aim of exploring different scenarios. This task, however, can be challenging to perform. This is primarily due to three, partially interconnected reasons:

i) The high dimensionality of the input space;

ii) The computational power that is required to perform each simulation;

iii) The amount of time that is required to perform each simulation.

The above challenges therefore call for the development of appropriate statistical methodologies that enable the study of a simulator's dynamics and that can properly handle the uncertainties associated with the analysis.

1.1.2. The Need to "Emulate" Simulators

During the 1980's and 1990's, the increasingly important role played by computer simulators in studying a variety of problems, and the difficulty to perform an exhaustive search of their parameter space, gave rise to a new field: the one of the design and analysis of computer experiments, as it is referred to in the seminal work Sacks et al. [1989]. The term "computer experiments" refers to a sequence of runs of the simulator; the term "design" concerns the choice of the inputs at which the simulator should be run, to make best use of the information that these will provide. The term "analysis" (of computer experiments) has a wider, and at the same time deeper, interpretation. It refers to the way in which the information provided by the design runs should be processed to make robust inference about the dynamics of the simulator and, even more importantly, about the dynamics of the phenomenon that the simulator describes. In this regard, let us quote an excerpt of Lionel Galway's and Thomas Lucas' comments to Craig et al. [1997].

> *The computer models [...] tend to be very large, often with thousand of parameters and the run times are correspondingly lengthy, so relatively few*

> *computer runs are feasible. As a result, relatively sparse "data" exist [...]. However, [...] important decisions must be made, e.g., setting standards and regulations for carbon-dioxide emissions and nuclear waste sites or efficiently managing hydrocarbon production. The first question is how to use such models rigorously and how to account for our uncertainty about the models' relationships to reality.*

The field of Gaussian process (GP) emulation provides the statistical framework to tackle the problem of the analysis of computer experiments. It is now recognised to be a prominent part of a more general field, the one of Uncertainty Quantification (UQ). In simple words, an emulator is a statistical model of the simulator, whose notable advantage over the simulator is the ability to provide predictions of the response at untried inputs in a significantly reduced amount of time (milliseconds rather than weeks, for instance). Most importantly, the emulator predictions take the form of probabilistic statements, which do not simply provide a "best guess" of the simulator response, but attach probabilistically quantified levels of uncertainty to it. If suitably calibrated, an emulator may also be able to account for the discrepancy between the simulator and reality.

The idea behind GP emulation is to model the simulator as a stochastic process, rather than as a deterministic function. One of the motivations behind this choice is that the simulator output corresponding to a particular input $\boldsymbol{x}$ is essentially unknown, till a notable amount of time and computational resources are invested in running the simulator at the particular input configuration $\boldsymbol{x}$ of interest: Kennedy and O'Hagan [2001] coined the term "code uncertainty" to refer to this kind of uncertainty. It is therefore clear that the theory of GP emulation relies on the ones of stochastic processes. Moreover, it is classically developed within a Bayesian setting: "beliefs" are initially expressed on the stochastic process modelling the simulator, and they are subsequently "updated" in light of the simulator response at a small sample of input configurations.

It is the author's wish to ensure that the ideas underlying Bayesian inference, and the main concepts and results concerning stochastic processes, are not left unclear in the mind of the potentially unfamiliar reader. For this reason, we defer the treatment of GP emulation to Chapter 2, and devote this chapter to the exposition of the previous

topics. Presenting them here also gives us the opportunity to set the notation for the framework that will be used in Chapter 2 to present and investigate the theory of GP emulation.

The structure of this chapter is as follows. Section 1.2 is devoted to the illustration of the central ideas of Bayesian statistics. Section 1.3 introduces stochastic and Gaussian processes, and related results. Section 1.4 examines how properties of covariance functions and correlation lengths affect a stochastic process. While some of the results shown in these last two sections may appear of a theoretical nature, they are of fundamental importance in applied contexts, as we will see in Part II of the thesis.

1.2. Introduction to Bayesian Inference

In this section, we provide an introduction to the framework of Bayesian inference, starting with the illustration of its key principles via a brief and easy example. Even if currently unfamiliar with Bayesian statistics, the reader will then be able to recognise that very similar principles and steps characterise the construction of the emulator (Chapter 2), although the mathematics will necessarily be more involved.

For a more complete and detailed introduction, we refer the reader to the book of Peter Lee, Lee [2012]. This is one of the most popular introductory texts to Bayesian Statistics, and covers a variety of both theoretical and computational topics. A possibly simpler, and easily-accessible introduction is provided in Bolstad and Curran [2016]. This text accompanies most of the theory with illustrations and numerical examples, and also carries out a detailed comparison between the performances of the Bayesian and the more classical frequentist approach.

1.2.1. Illustrative Example

Let us consider the following elementary example.

Setting: Suppose you would like to know whether you are affected by a given rare disease, D. This affects 1% of the population. A test is available, but it is expensive to carry out and the results take a long time to be available. The doctor therefore

suggests to go, firstly, through a simpler route: to take a blood test to check whether you possess a given enzyme, E. The enzyme is known to be produced under the disease, but it can also be produced as a consequence of a number of other factors. In fact, it is present in 40% of the total population. Assuming that the test reveals that you possess the enzyme, and assuming you are a very rational person, how likely do you believe it is that you are actually affected by the disease?

Answer: Using an intuitive notation and elementary probability rules, the answer is easily computed. We have:

$$\mathbb{P}(D) = 0.01, \quad \mathbb{P}(E\,|\,D) = 1, \quad \mathbb{P}(E) = 0.4. \tag{1.1}$$

We are then interested in computing $\mathbb{P}(D\,|\,E)$, the probability of being affected by the disease, if the enzyme test is positive. Through Bayes' rule, we get:

$$\mathbb{P}(D\,|\,E) = \frac{\mathbb{P}(E\,|\,D)\,\mathbb{P}(D)}{\mathbb{P}(E)} = \frac{1 \times 0.01}{0.4} = 0.025\,. \tag{1.2}$$

Albeit elementary, the example provides a typical illustration of the paradigm underlying Bayesian statistics: in light of the observation (the test), you have updated your information from a prior one (having 1% chance of being affected by the disease) to a posterior one (having 2.5% chance of being affected by the disease).

Notice that, from the law of total probability, we have the following relation:

$$\mathbb{P}(E) = \mathbb{P}(E\,|\,D)\,\mathbb{P}(D) + \mathbb{P}(E\,|\,D^c)\,\mathbb{P}(D^c)\,. \tag{1.3}$$

In equation (1.1), we can therefore replace the information about $\mathbb{P}(E)$ with $\mathbb{P}(E|D^c) = (0.4 - 1 \cdot 0.01)/0.99 = 13/33$, hence rewriting (1.1) as:

$$\mathbb{P}(D) = 0.01, \quad \mathbb{P}(E\,|\,D) = 1, \quad \mathbb{P}(E\,|\,D^c) = 13/33. \tag{1.4}$$

In order to easily analyse the example within the Bayesian framework, it is convenient to rephrase it in terms of random variables. We can consider the following:

- X: the Bernoulli random variable denoting whether a person is affected by the disease or not.

- Y: the Bernoulli random variable denoting whether a person possesses the enzyme or not.

For convenience of interpretation, we denote by D (*disease*) and H (*health*) the two possible outcomes of X, and by E (*enzyme*) and F (*free*) the two possible outcomes of Y. Within this formulation, equation (1.4) becomes as follows:

$$\mathbb{P}(X=D) = 0.01\,; \tag{1.5.a}$$

$$\mathbb{P}\big(Y=E\,|\,X=D\big) = 1, \quad \mathbb{P}\big(Y=E\,|\,X=H\big) = 13/33\,. \tag{1.5.b}$$

Equation (1.5.a) specifies the distribution of X, and equation (1.5.b) specifies the conditional distribution of Y given X. Through Bayes' rule, we can easily compute the conditional distribution of X given Y, as done in (1.2). Within a Bayesian setting, these distributions have precise names.

1. The distribution of X, equation (1.5.a), is called **prior distribution**. This is the distribution we associate to the random variable of interest, according to the knowledge available before any observation is made.

2. The conditional distribution of Y given X, equation (1.5.b), is called **likelihood**. It reflects the way in which the random variable X affects the observations Y.

3. The conditional distribution of X given Y is called **posterior distribution**. It reflects the updated distribution of X, given that a specific instance of Y has been observed.

1.2.2. General Setting and Notation

The simple illustration in Subsection 1.2.1 introduces the reader new to Bayesian statistics to the basic concepts and terminology. Here we expand on this by providing a more general and unifying framework, that applies indifferently to random objects of different nature.

In a statistical inference framework, we are interested in the value of an unknown quantity Θ, not directly observable. However, we know that the value of Θ affects the

outcome of a random quantity, Y, which we may be able to observe. In our previous example, Θ tells us whether we are affected by the disease (not directly observable), and Y tells us whether we possess the enzyme E. The key question is as follows:

> "*Provided we observe an instance of Y, how can we use this information to draw inference on Θ?*"

In the classical frequentist approach, Θ is considered to be a fixed, but unknown value. Given the observation $Y = y$, the most common approach is to estimate Θ with the value that maximises the likelihood of seeing y as instance of Y. For example, if $Y \sim N(\Theta, 1)$ and we observe $Y = y$, then we would estimate $\Theta = y$. What instead characterises the Bayesian approach, is that the unknown parameter Θ is itself considered a random variable, about which we are asked to specify a distribution. The latter will encode any external information that is available (for example, from experts' judgements) and can be "updated" in light of the observation $Y = y$.

We have essentially seen this in Subsection 1.2.1 (equation (1.2)), but we make it explicit here, in the simple case where Θ and Y are discrete random variables. In this case, we define:

$$\pi_{\Theta}(\theta) \;:=\; \mathbb{P}\big(\Theta = \theta\big)\,, \tag{1.6}$$

$$\pi_{Y|\Theta}(y \,|\, \theta) \;:=\; \mathbb{P}\big(Y = y \,|\, \Theta = \theta\big)\,. \tag{1.7}$$

The function $\pi_{\Theta}(\cdot)$ is the prior distribution of Θ. The function $\pi_{Y|\Theta}(\cdot \,|\, \theta)$ specifies what distribution Y follows, if $\Theta = \theta$: this is the likelihood function of θ. If we observe $Y = y^*$, then Bayes' rule yields the following posterior distribution for Θ:

$$\pi_{\Theta|Y}(\theta|\, y^*) = \frac{\pi(y^*|\,\theta)\,\pi(\theta)}{\pi(y^*)} = \frac{\pi(y^*|\,\theta)\,\pi(\theta)}{\sum_j \pi(y^*|\,\theta_j)\,\pi(\theta_j)}\,. \tag{1.8}$$

We have omitted the right-hand side (RHS) subscripts, since the argument(s) of each density allow to identify the distribution we refer to.

If Θ and Y are continuous real variables, then we replace the definitions (1.6) and (1.7), which specify probability masses, with probability densities. If $\pi(\theta)$ is the prior density of Θ, and $\pi(y|\,\theta)$ is the density of Y for $\Theta = \theta$, then the posterior density of

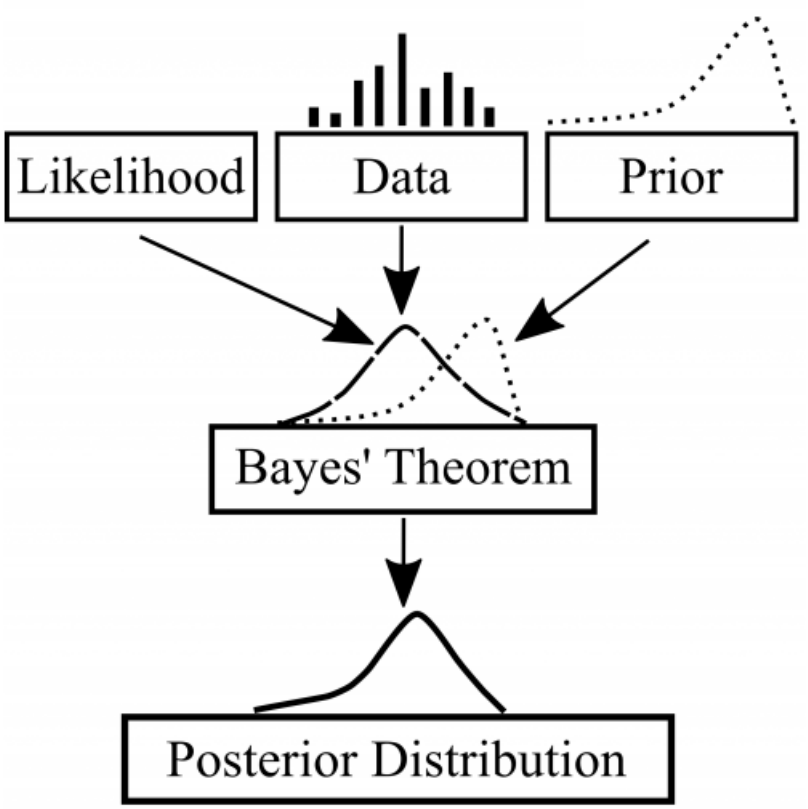


Figure 1.1: Illustration of the classical Bayesian principle: given a likelihood function and observed data, prior information about a parameter Θ is updated into posterior information for the same parameter. Compare with equation (1.10). Permission to use the above illustration from Doll and Jacquemin [2018] has been kindly granted by The American Fisheries Society.

Θ given the observation $Y = y^*$ is computed as:

$$\pi(\theta|\, y^*) = \frac{\pi(y^*|\,\theta)\,\pi(\theta)}{\int \pi(y^*|\,\tilde{\theta})\,\pi(\tilde{\theta})\,d\tilde{\theta}}\,. \tag{1.9}$$

The procedure generalises to random objects of any nature: either Y, or Θ, or both, can be random variables, random vectors, or even random functions. This last case is of particular interest in GP emulation, as we will see in Chapter 2. We can therefore summarise the above in one simple and fundamental formula, at the heart of any Bayesian procedure:

$$\pi(\theta\,|\,y^*) \;\propto\; \pi(y^*|\,\theta) \times \pi(\theta)\,. \tag{1.10}$$

The proportionality sign is due to having neglected the factor $\pi(y^*)^{-1}$: this is independent of θ, the only variable the posterior is a function of. In an informal but easily memorable way, we may write the following:

$$\text{Posterior} \;\propto\; \text{Likelihood} \times \text{Prior}\,. \tag{1.11}$$

Figure 1.1 provides a schematic illustration of the way information from different sources is merged within a typical Bayesian framework: equation (1.10) represents the mathematical formulation of this. In Example 1.1, we illustrate an application of the formula to a simple context, where Θ and Y are both real random variables, respectively continuous and discrete. Before doing that, we make a more technical note on the case dealt with in equation (1.9).

Θ and Y Continuous: Technical Note

Suppose both Θ and Y are continuous real random variables. Then, the event $\{\Theta = \theta\}$ has probability zero, hence the quantity $\mathbb{P}\big(Y \in A \,|\, \Theta = \theta\big)$, for an interval $A \subseteq \mathbb{R}$, is not defined in terms of classical conditional probabilities. Even more, one may wonder what precise meaning to associate to the density of Y given Θ: we cannot define it as the derivative of $\mathbb{P}\big(Y \in [0, y] \,|\, \Theta = \theta\big)$ with respect to y, since this last term is undefined. Within a measure-theoretic setting, it is however possible to define $\mathbb{P}\big(Y \in A \,|\, \Theta = \theta\big)$, and have it satisfy all the intuitive properties that one would expect. The interested reader may for example consult the book of the famous Russian mathematician Albert Shiryaev, Shiryaev [1996] (Chapter II, Section 7). The author first defines the following function of θ:

$$m_{_X}(\theta) = \mathbb{E}\big[X \,|\, \Theta = \theta\big]\,, \tag{1.12}$$

for any random variable X defined on the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$ of Θ. Then, the desired definition of conditional probability follows naturally, by considering X to be the appropriate indicator function:

$$\mathbb{P}\big(Y \in A \,|\, \Theta = \theta\big) = \mathbb{E}\big[\mathbb{1}_{\{Y \in A\}} \,|\, \Theta = \theta\big]\,, \quad A \in \mathcal{F}\,. \tag{1.13}$$

Here, we do not go into the details of definition (1.12), which would first require the

introduction of appropriate tools (partially overlapping with the ones presented in Subsection 1.3.2). It was however worth pointing out the caveat, and reassure the reader that a legitimate definition of conditional density can be given, and that this satisfies Bayes' rule (1.9).

Let us now conclude this introductory section on Bayesian statistics with the simple example mentioned prior to the previous remark.

Example 1.1. Let Y be the number of heads in N coin tosses. The coin is potentially biased: the probability $\Theta \in [0,1]$ of obtaining head in a single toss is unknown. Supposing to have no reasons to favour one value or the other, we put a uniform prior on Θ. Therefore, we have:

$$\Theta \sim \mathcal{U}(0,1)\,, \quad Y\,|\,\Theta \sim B(N,\Theta)\,.$$

We have denoted with $B(n,p)$ the binomial distribution with parameters $n \in \mathbb{N}$ and $p \in [0,1]$. We now toss the coin N times, and obtain $y^* \in \{0,\dots,N\}$ heads in total. How does this observation modify our prior (flat, in this case) belief on Θ? From Bayes' rule (1.10), we obtain:

$$\begin{aligned}\pi(\theta\,|\,Y=y^*) &\propto \pi(y^*|\,\theta)\times\pi(\theta)\\ &\propto \theta^{y^*}(1-\theta)^{N-y^*}\times \mathbb{1}_{[0,1]}(\theta)\,.\end{aligned} \tag{1.14}$$

Defining for simplicity $a = y^*+1$ and $b = N-y^*+1$, and taking into account the normalising factor, we find:

$$\pi(\theta\,|\,Y=y^*) \;=\; \frac{\theta^{a-1}(1-\theta)^{b-1}}{B(a,b)}\,, \qquad \theta\in[0,1]\,. \tag{1.15}$$

This is the density of a Beta distribution with parameters a and b; the function $B(a,b)=\int_0^1 x^{a-1}(1-x)^{b-1}dx$ is the Beta function.

Having seen exactly y^* heads out of N trials, we expect the posterior density of Θ to assign higher probabilities to values near $\Theta = y^*/N$. This is indeed the case, with the mode of (1.15) being exactly $(a-1)/\big((a-1)+(b-1)\big) = y^*/N$. Notice that this posterior mode coincides with the maximum likelihood estimate of Θ: maximising

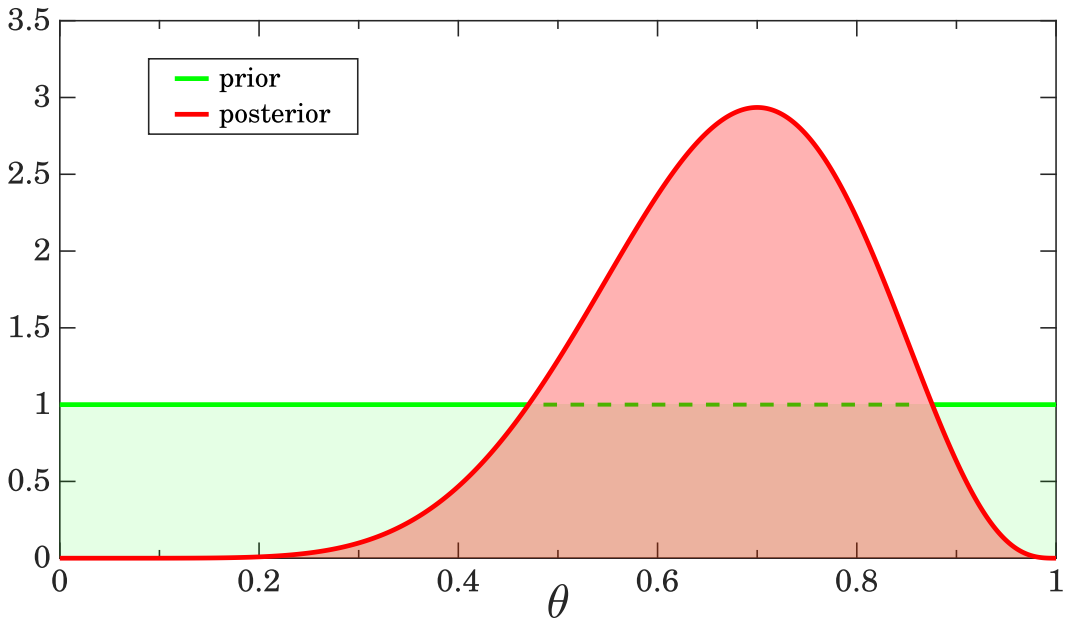


Figure 1.2: Illustration of the prior and posterior densities of a random variable Θ, within the simple setting of Example 1.1. Θ represents the probability of tossing a head in a single coin flip. A uniform prior for Θ is chosen (green), which is updated to the posterior (red) after observing 7 head tosses in 10 trials. The posterior mode is, as expected, 0.7.

the likelihood or the posterior is in this case equivalent, since the prior is constant. The Bayesian approach returns however a full distribution for Θ, rather than a single-valued estimate. We illustrate in Figure 1.2 the change from the prior to the posterior distribution of Θ, in the case where $y^* = 7$ heads are tossed out of $N = 10$ trials.

1.3. Gaussian Processes

As explained in Section 1.1, the main idea behind GP emulation is to create a fast and reliable statistical surrogate of an expensive computer simulator. To this aim, the simulator is modelled as a GP, a particular kind of stochastic process. In this section we present the basic definitions and results about stochastic and Gaussian processes. We do this within a relatively formal probabilistic setting which we introduce in Subsection 1.3.2, after providing a brief intuition in Subsection 1.3.1. For a wider treatment, we refer the reader to Rasmussen and Williams [2006], providing a good and practical overview on GPs, with special focus on their multiple uses in Machine Learning.

1.3.1. Intuition Behind Stochastic Processes

Gaussian processes are stochastic processes. A stochastic process is informally described as a collection of random variables, indexed via the elements of a given set. The most common example is the one where the index set is time, in which case the process is naturally thought of as describing the (random) time evolution of a given quantity of interest. In this case, it is natural to denote the process as follows:

$$\big(X(t)\big)_{t\geq 0}\,. \tag{1.16}$$

$X(t)$ is the random variable denoting the state of the process at time t. The distribution of $X(t)$ at the different times t clearly needs to be specified. However, this is not enough to completely determine the stochastic nature of X. As a minimal further requirement, we have to specify the dependence between the random variables at different times t: for example, if we assume the process continuous, we expect the correlation between $X(t)$ and $X(t+\varepsilon)$ to be high for $\varepsilon \ll 1$. More generally, we need to specify the full joint distribution of the process at any finite collection of times $t_1, \ldots, t_k \geq 0$, that is, the distribution of the random vector:

$$\big(X(t_1), \ldots, X(t_k)\big) \in \mathbb{R}^k\,. \tag{1.17}$$

This will identify what is referred to as the law of the stochastic process (formally defined in the following Subsection 1.3.2). GPs are particular stochastic processes, the ones for which the distribution of (1.17) is multivariate Gaussian. The following section formalises the previous concepts within an appropriate framework.

1.3.2. Formal Definitions and Properties

In order to treat randomness in a formal and consistent manner, we need to introduce the concept of probability space. This is a triple $(\Omega, \mathcal{F}, \mathbb{P})$, where:

- Ω is the so-called sample space. The elements in Ω represent all the possible outcomes of the random experiment being considered.

- $\mathcal{F} \subseteq 2^\Omega$ is a set of events (2^Ω denotes the power set of Ω), an event being a collection of single outcomes for which a probability is defined.
- $\mathbb{P}$ is the probability measure assigning a probability to each event in $\mathcal{F}$.

In order to satisfy the standard probability axioms (*i.e.*, $\mathbb{P}(A) \geq 0$, $\mathbb{P}(\Omega) = 1$, and $\mathbb{P}(\cup_i A_i) = \sum_i \mathbb{P}(A_i)$ for mutually disjoint sets A_i) we need to impose some requirements on $\mathcal{F}$, the set of events. In particular, we need to request that the following three properties hold:

1. $\Omega \in \mathcal{F}$;
2. $A \in \mathcal{F} \;\Rightarrow\; \Omega \setminus A \in \mathcal{F}$ (closure of $\mathcal{F}$ under complements);
3. $\{A_n\}_{n\in\mathbb{N}} \subseteq \mathcal{F} \;\Rightarrow\; \bigcup_n A_n \in \mathcal{F}$ (closure of $\mathcal{F}$ under countable unions).

A set $\mathcal{F} \subseteq 2^\Omega$ satisfying properties 1–3 is called a σ-algebra. In a probability space, the elements of the σ-algebra $\mathcal{F}$ are often referred to as measurable events: we can measure their mass through $\mathbb{P}$. If no probability measure is specified, a pair $(\Omega, \mathcal{F})$ of a set and a corresponding σ-algebra is simply called a measurable space. Notice that, if Ω is a topological space (a space for which the notion of open set is defined), there is a natural σ-algebra associated with Ω. This is the Borel σ-algebra, denoted $\mathcal{B}(\Omega)$, defined as the smallest σ-algebra containing all open sets of Ω.

We now provide the formal definition of random variable, and immediately after the one of stochastic process which we will use in the rest of this work.

Definition 1.3.1. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and $(E, \mathcal{E})$ a measurable space. A random variable between $(\Omega, \mathcal{F}, \mathbb{P})$ and $(E, \mathcal{E})$ is a function

$$X \colon \Omega \to E\,, \tag{1.18}$$

such that the pre-image of an element of $\mathcal{E}$ through X is an element of $\mathcal{F}$. That is:

$$X^{-1}(A) := \big\{\omega \in \Omega \mid X(\omega) \in A\big\} \in \mathcal{F} \quad \forall\, A \in \mathcal{E}\,. \tag{1.19}$$

This way, for any $A \in \mathcal{E}$, the probability $\mathbb{P}(X \in A) = \mathbb{P}\big(X^{-1}(A)\big)$ is well defined.

We now introduce the definition of stochastic process. We limit our treatment to the case of real output with associated Borel σ-algebra, since this is essentially the only case of interest in this work. The following definition (but not all of the following results) remains valid if the pair $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ is replaced by any measurable space $(E, \mathcal{E})$.

Definition 1.3.2. Let $\mathcal{I}$ be a set. A stochastic process with index set $\mathcal{I}$ and underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a function

$$X\colon\ \Omega \times \mathcal{I}\ \to\ \mathbb{R}\,, \tag{1.20}$$

such that, for each $t \in \mathcal{I}$, the function $X_t(\cdot) := X(\cdot, t)$ is a random variable. That is, for any $A \in \mathcal{B}(\mathbb{R})$, we have:

$$X_t{}^{-1}(A) := \big\{\omega \in \Omega \mid X(\omega, t) \in A\big\} \in \mathcal{F}\,. \tag{1.21}$$

We have denoted with t a general element of $\mathcal{I}$, since the latter may often have an interpretation as set of times. However, both in the previous definition and in the following results, $\mathcal{I}$ is to be considered just a set, which needs not have any particular additional structure.

According to Definition 1.3.2, it is natural to see a stochastic process as a collection of random variables, indexed by the elements of $\mathcal{I}$. However, notice that for each sample element $\omega \in \Omega$, the process returns a function from $\mathcal{I}$ to $\mathbb{R}$:

$$X(\omega, \cdot)\colon\ \mathcal{I}\ \to\ \mathbb{R}\,. \tag{1.22}$$

This is often called a sample trajectory, path, or realisation of the process. Hence, it is also convenient to see a stochastic process as a random variable (function of $\omega \in \Omega$ only) taking values in the space of functions from $\mathcal{I}$ to $\mathbb{R}$, denoted by $\mathbb{R}^{\mathcal{I}}$:

$$\begin{aligned} X\colon\ \Omega\ &\longrightarrow\ \mathbb{R}^{\mathcal{I}} \\ \omega\ &\mapsto\ X(\omega, \cdot) \end{aligned} \tag{1.23}$$

The probability measure $\mathbb{P}$ on $(\Omega, \mathcal{F})$ can be "transported" forward to the space $\mathbb{R}^{\mathcal{I}}$ endowed with its Borel σ-algebra (whose existence and construction is not obvious, see point 1 just below). This allows to identify the different elements of $\mathbb{R}^{\mathcal{I}}$ as more, or

less, likely realisations of the process. This can all be made rigorous, but it is not the aim of the present section, nor of the present work, to go into such measure-theoretic details. For the purposes of this work, it suffices to know the following:

1. The Borel σ-algebra $\mathcal{B}(\mathbb{R}^{\mathcal{I}})$ is well-defined even for general uncountable sets $\mathcal{I}$ (this is constructed through the so-called cylinder sets, see Shiryaev [1996, Chap II.2, Thm 3]);

2. A central result, due to the eminent probabilist Andrey Kolmogorov, ensures that the probability measure induced by the process X on $\big(\mathbb{R}^{\mathcal{I}}, \mathcal{B}(\mathbb{R}^{\mathcal{I}})\big)$ is uniquely determined by a set of measures on $\mathbb{R}^k$, $k \in \mathbb{N}$. These are the finite-dimensional distributions of the process, introduced below.

Definition 1.3.3. Let X be a stochastic process on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with index set $\mathcal{I}$. For fixed $t_1, \dots, t_k \in \mathcal{I}$, call $\mu_{t_1,\dots,t_k}$ the distribution of the random vector

$$\big(X_{t_1}, \dots, X_{t_k}\big) \colon\ \Omega \ \to\ \mathbb{R}^k\,. \tag{1.24}$$

The family

$$\Big\{ \mu_{t_1,\dots,t_k} \ \Big|\ k \in \mathbb{N},\ t_1, \dots, t_k \in \mathcal{I} \Big\} \tag{1.25}$$

is called the family of finite-dimensional distributions of the process X.

Within this setting, the famous result of Kolmogorov, informally introduced in point 2 above, can be formulated as follows.

Theorem 1.3.4 (Kolmogorov)**.** *Let X be a stochastic process on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with index set $\mathcal{I}$. Then the law of the process, that is, the probability measure μ on $\big(\mathbb{R}^{\mathcal{I}}, \mathcal{B}(\mathbb{R}^{\mathcal{I}})\big)$ defined by*

$$\mu(A) \ =\ \mathbb{P}\big(\,\omega \in \Omega \mid X(\omega, \cdot) \in A\,\big) \quad \forall\, A \in \mathcal{B}\big(\mathbb{R}^{\mathcal{I}}\big), \tag{1.26}$$

is uniquely identified by the set of all finite dimensional distributions $\{\mu_{t_1,\dots,t_k}\}$ of the process.

Remark 1.2. It is worth noting that the original result of Kolmogorov, known as Kolmogorov's extension theorem, is slightly more general than the one stated above:

see, for example, Shiryaev [1996, Chap II.3, Thm 4], Shiryaev [1996, Chap II.9, Thm 1], or Øksendal [1998, Thm 2.1.5]. It states that a family of finite-dimensional distributions on $\left(\mathbb{R}^k, \mathcal{B}(\mathbb{R}^k)\right)_{k\in\mathbb{N}}$ can be uniquely extended to a measure on $\left(\mathbb{R}^{\mathcal{I}}, \mathcal{B}(\mathbb{R}^{\mathcal{I}})\right)$, provided that the family satisfies some relatively natural conditions known as consistency conditions. These are clearly satisfied in the case where the distributions are the finite-dimensional distribution of a given stochastic process, hence our statement.

In light of the solid mathematical ground laid by Kolmogorov's theorem, we can define a Gaussian process as follows.

Definition 1.3.5 (Gaussian Process)**.** A Gaussian process is a stochastic process X, for which all finite-dimensional distributions are Gaussian: that is, the random vectors

$$\left(X_{t_1}, \dots, X_{t_k}\right) \in \mathbb{R}^k, \quad t_1, \dots, t_k \in \mathcal{I}, \tag{1.27}$$

are all jointly Gaussian distributed.

The previous definition allows to uniquely identify the law of the process X, thanks to Kolmogorov's theorem. In Figure 1.3 we show some sample trajectories of what is arguably the most famous Gaussian process, the Brownian motion. The parameter set $\mathcal{I}$ is a set of times, $\mathcal{I} = [0, 25]$ in Figure 1.3. The finite dimensional distributions are instead as follows:

$$\left(X_{t_1}, \dots, X_{t_k}\right) \sim N(\mathbf{0}, \boldsymbol{C}_{\boldsymbol{t_1},\dots,\boldsymbol{t_k}}), \tag{1.28}$$

where $\mathbf{0} \in \mathbb{R}^k$ is the zero k-dimensional vector, and the element (i, j) of the covariance matrix $\boldsymbol{C}_{\boldsymbol{t_1},\dots,\boldsymbol{t_k}} \in \mathbb{R}^{k\times k}$ is given by

$$\left(\boldsymbol{C}_{\boldsymbol{t_1},\dots,\boldsymbol{t_k}}\right)_{ij} = \min\{t_i, t_j\}, \quad i, j = 1, \dots, k. \tag{1.29}$$

As Figure 1.3 shows, the Brownian motion has relatively irregular paths. In the next section, we see that the regularity of the paths of a GP is affected by an important property of the process, its covariance function.

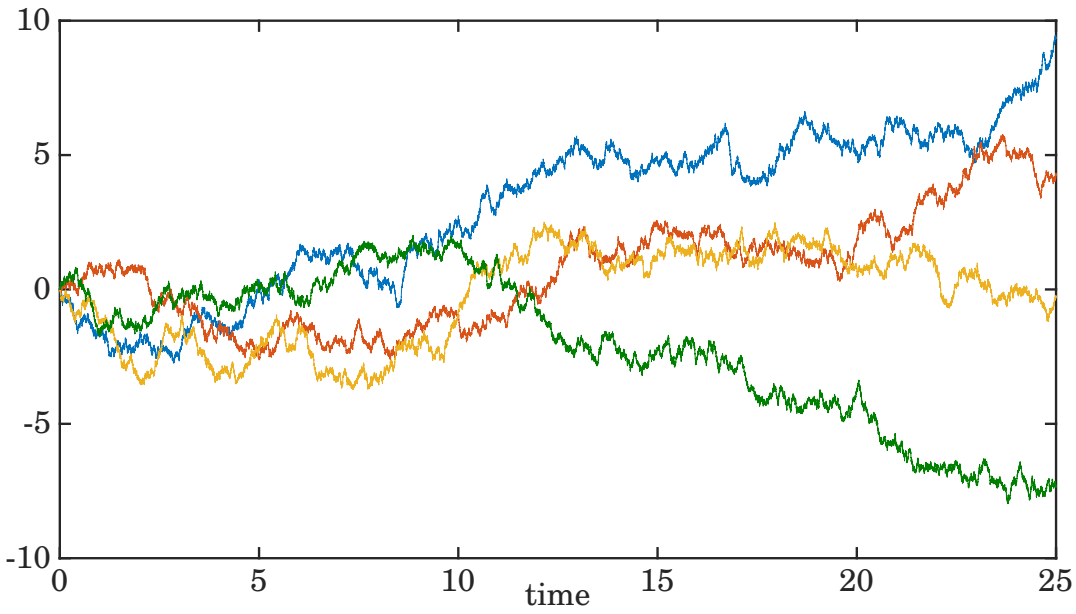


Figure 1.3: Plot of four different trajectories of Brownian Motion, the canonical example of a Gaussian process. Each trajectory corresponds to a different ω of the sample space Ω.

1.4. Covariance Functions

Let us consider a stochastic process X with index set $\mathcal{I}$. Throughout this section, we need the technical hypothesis that the process has finite second-order moments. That is, we assume the following:

$$\mathbb{E}\big[X_t^2\big] < \infty \quad \forall\, t \in \mathcal{I}\,. \tag{1.30}$$

If the previous condition holds, we say in short that X is a second-order process. From (1.30), it follows that the mean of all X_t is as well finite. Indeed:

$$\big|\mathbb{E}[X_t]\big| \;=\; \big|\mathbb{E}[X_t \cdot 1]\big| \;\leq\; \mathbb{E}\big[X_t^2\big]^{\frac{1}{2}}\,\mathbb{E}\big[1^2\big]^{\frac{1}{2}} \;=\; \mathbb{E}\big[X_t^2\big]^{\frac{1}{2}} < \infty\,. \tag{1.31}$$

The inequality used is simply Cauchy-Schwarz inequality ($|\langle f, g\rangle| \leq \|f\| \cdot \|g\|$) on $L^2(\Omega, \mathbb{P})$, with the scalar product

$$\langle f, g\rangle = \int_\Omega f(\omega)\, g(\omega)\, d\mathbb{P}(\omega), \qquad f, g \in L^2(\Omega, \mathbb{P}). \tag{1.32}$$

$L^2(\Omega, \mathbb{P})$ is the space of real functions on Ω, whose square is integrable with respect to the measure $\mathbb{P}$ (the reader may refer to Jacod and Protter [2000, Chapter 9] for background on integration with respect to a probability measure). Given a stochastic process X on $\mathcal{I}$ with finite second-order moments, we can encode the information about its mean at any point $t \in \mathcal{I}$ and about the covariance between any pair of random variables X_s and X_t in two functions, which we present in the next subsection.

1.4.1. Definitions and Results

Definition 1.4.1. Let $\mathcal{I}$ be a set, and X a second-order stochastic process with index set $\mathcal{I}$. The function $m\colon \mathcal{I} \to \mathbb{R}$, defined by

$$m(t) := \mathbb{E}[X_t], \quad t \in \mathcal{I}, \tag{1.33}$$

is called the mean function of X. The function $C\colon \mathcal{I} \times \mathcal{I} \to \mathbb{R}$, defined by

$$C(s,t) := \operatorname{Cov}(X_s, X_t), \quad s,t \in \mathcal{I}, \tag{1.34}$$

is called the covariance function of X.

Remark 1.3. The mean and covariance functions are not sufficient, in general, to identify the law of the process. However, if the process is Gaussian, they uniquely identify its law. This follows immediately from the fact that the distribution of a k-dimensional Gaussian random vector is uniquely determined by its mean (in $\mathbb{R}^k$) and covariance matrix (in $\mathbb{R}^{k\times k}$), and from Kolmogorov's Theorem 1.3.4, which ensures that the finite-dimensional distributions uniquely specify the law of the process.

Let us now recall that the covariance matrix of any multidimensional random vector is always positive semi-definite, and obviously symmetric. This justifies the following definition.

Definition 1.4.2. Let $\mathcal{I}$ be a set. We say that a function $C\colon \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ is a valid covariance function, if the following two conditions hold:

1. for any $s,t \in \mathcal{I}$, $C(s,t) = C(t,s)$;

2. for any $k \in \mathbb{N}$ and any different $t_1, \dots, t_k \in \mathcal{I}$, the matrix

$$\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{k \times k}, \quad a_{ij} = C(t_i, t_j)\,, \tag{1.35}$$

is positive semi-definite.

The covariance function of a stochastic process clearly satisfies the two conditions above. Conversely given any valid covariance function $C\colon \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ and any function $m\colon \mathcal{I} \to \mathbb{R}$, one may wonder whether there exists a stochastic process having mean function $m(\cdot)$ and covariance function $C(\cdot, \cdot)$. The answer is positive. In fact, it is not difficult to show something even stronger: a *Gaussian* process with the specified mean and covariance functions exists. The proof of this statement only needs to check that the family of finite-dimensional Gaussian measures with mean vectors and covariance matrices derived from $m(\cdot)$ and $C(\cdot, \cdot)$ is consistent, according to the definition of consistency in Kolmogorov's extension theorem (recall Remark 1.2 on page 22); the result then follows by Kolmogorov's theorem itself. We skip the proof, but retain this important and classical result.

Theorem 1.4.3. *Let $\mathcal{I}$ be a set, $m\colon \mathcal{I} \to \mathbb{R}$ any function, and $C\colon \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ a valid covariance function. Then, there exists a Gaussian process X with mean function $m(\cdot)$ and covariance function $C(\cdot, \cdot)$.*

Remark 1.4. Consider the setting of the above theorem, guaranteeing the existence of a Gaussian process X with prescribed mean and covariance functions. If the index set $\mathcal{I}$ is a measurable space (for example, $\mathcal{I}$ is an interval of the real line with associated Borel σ-algebra $\mathcal{B}(\mathcal{I})$), then no statement can be made on the measurability of the process' paths as functions of $t \in \mathcal{I}$. Notice, indeed, that the Definition 1.3.2 of stochastic process does not ask for measurability in $\mathcal{I}$. This is the classical definition of stochastic process, for which Kolmogorov's extension theorem (Shiryaev [1996, Chap II.9, Thm 1]) holds true.

1.4.2. Connection to Mean-square and Pathwise Continuity

The covariance function encodes important information about the process: in particular, its regularity affects the regularity of the process. To make this rigorous, we need

a notion of distance between elements of the index set $\mathcal{I}$, as well as a notion of distance between square-integrable random variables. As to the first point, we suppose that the index set is an open set of $\mathbb{R}^p$, with the Euclidean distance. To remark both via notation and terminology that this is not anymore an unstructured set, we denote it by $\mathcal{P}$ rather than $\mathcal{I}$ and refer to $\mathcal{P}$ as to the *parameter space* or *input space* of the process. As to the second point, we consider the distance on $L^2(\Omega, \mathbb{P})$ induced by the scalar product (1.32), and define continuity in this metric.

Terminology: Random Processes and Random Fields

In the case where $\mathcal{P} \subseteq \mathbb{R}^p$ with $p \geq 2$, it is common to refer to the process as to a "random field". Accordingly, in the stochastic literature, Gaussian processes with multi-dimensional inputs are usually referred to as Gaussian random fields. The expression Gaussian process, however, has always been used within the GP emulation literature, independently of the input dimensionality. In this work, we do not alter what has become a well-established nomenclature convention in the field. We therefore use the term "Gaussian process" (or stochastic process) also in the case where the process' input space is multi-dimensional.

Definition 1.4.4. Let X be a stochastic process with parameter space $\mathcal{P} \subseteq \mathbb{R}^p$, and let $\boldsymbol{u}^* \in \mathcal{P}$. We say that X is continuous in mean square at $\boldsymbol{u}^*$, or mean-square (MS) continuous at $\boldsymbol{u}^*$, if for any sequence $\big(\boldsymbol{u_n}\big)_{n\in\mathbb{N}} \subseteq \mathcal{P}$ the following holds:

$$\lim_{n\to\infty} \boldsymbol{u_n} = \boldsymbol{u}^* \quad \Longrightarrow \quad \lim_{n\to\infty} \mathbb{E}\big[\, |X_{\boldsymbol{u}^*} - X_{\boldsymbol{u_n}}|^2 \,\big] = 0\,. \tag{1.36}$$

In the previous definition, notice that $\mathbb{E}\big[\, |X_{\boldsymbol{u}^*} - X_{\boldsymbol{u_n}}|^2 \,\big] = \|X_{\boldsymbol{u}^*} - X_{\boldsymbol{u_n}}\|^2$, where $\|\cdot\|$ denotes the norm induced by the scalar product (1.32). The following result links the MS continuity of a process to the continuity of its covariance function.

Proposition 1.4.5. *Let X be a second-order process with parameter space $\mathcal{P} \subseteq \mathbb{R}^p$ and continuous mean function $m(\cdot)$. Let $C(\cdot,\cdot)$ be its covariance function. Then, the process X is MS continuous at $\boldsymbol{u}^* \in \mathcal{P}$ if and only if $C(\cdot,\cdot)$ is continuous at $(\boldsymbol{u}^*, \boldsymbol{u}^*)$.*

Proof. First, observe that we can assume the mean $m(\cdot)$ to be constantly zero. Indeed,

since $m(\cdot)$ is continuous, the MS continuity of the original process is equivalent to the MS continuity of the centred process $X_\cdot - m(\cdot)$. In the following, we therefore assume $m(\cdot) \equiv 0$, hence $C(\boldsymbol{u}, \boldsymbol{u}') = \mathbb{E}\big[X_{\boldsymbol{u}} X_{\boldsymbol{u}'}\big]$ for any $\boldsymbol{u}, \boldsymbol{u}' \in \mathcal{P}$.

"If" part

Let $(\boldsymbol{u_n}) \subseteq \mathcal{P}$ be a sequence. For each $n \in \mathbb{N}$, the following holds:

$$\begin{aligned}\mathbb{E}\big[\,|X_{\boldsymbol{u}^*} - X_{\boldsymbol{u_n}}|^2\,\big] &= \mathbb{E}\left[X_{\boldsymbol{u}^*}^2\right] + \mathbb{E}\big[X_{\boldsymbol{u_n}}^2\big] - 2\mathbb{E}\big[X_{\boldsymbol{u}^*} X_{\boldsymbol{u_n}}\big] \\ &= C(\boldsymbol{u}^*, \boldsymbol{u}^*) + C(\boldsymbol{u_n}, \boldsymbol{u_n}) - 2C(\boldsymbol{u}^*, \boldsymbol{u_n})\,. \end{aligned} \tag{1.37}$$

If $\boldsymbol{u_n}$ converges to $\boldsymbol{u}^*$, and $C(\cdot, \cdot)$ is continuous at $(\boldsymbol{u}^*, \boldsymbol{u}^*)$, from (1.37) we obtain:

$$\lim_{n\to\infty} \mathbb{E}\big[\,|X_{\boldsymbol{u}^*} - X_{\boldsymbol{u_n}}|^2\,\big] = C(\boldsymbol{u}^*, \boldsymbol{u}^*) + C(\boldsymbol{u}^*, \boldsymbol{u}^*) - 2C(\boldsymbol{u}^*, \boldsymbol{u}^*) = 0\,. \tag{1.38}$$

Hence, the process is MS continuous at $\boldsymbol{u}^* \in \mathcal{P}$.

"Only if" part

Consider two sequences $\big(\boldsymbol{u_n}\big) \subseteq \mathcal{P}$ and $(\boldsymbol{v_n}) \subseteq \mathcal{P}$ both converging to $\boldsymbol{u}^* \in \mathcal{P}$. We would like to prove the following:

$$\lim_{n\to\infty} C\big(\boldsymbol{u_n}, \boldsymbol{v_n}\big) = C(\boldsymbol{u}^*, \boldsymbol{u}^*)\,, \tag{1.39}$$

under the assumption that the process X is MS continuous at $\boldsymbol{u}^*$. It is convenient to rephrase this last hypothesis in terms of the corresponding L^2 norm, and use basic norm properties. The fact that X is MS continuous at $\boldsymbol{u}^*$ can be restated as follows:

$$\|X_{\boldsymbol{u_n}} - X_{\boldsymbol{u}^*}\|,\ \|X_{\boldsymbol{v_n}} - X_{\boldsymbol{u}^*}\| \overset{n\to\infty}{\longrightarrow} 0\,. \tag{1.40}$$

Consequently, we also have:

$$\|X_{\boldsymbol{u_n}} - X_{\boldsymbol{v_n}}\| \leq \|X_{\boldsymbol{u_n}} - X_{\boldsymbol{u}^*}\| + \|X_{\boldsymbol{u}^*} - X_{\boldsymbol{v_n}}\| \overset{n\to\infty}{\longrightarrow} 0\,. \tag{1.41}$$

Moreover, from (1.40) and the reverse triangle inequality (that is, $\big|\|X_{\boldsymbol{u_n}}\| - \|X_{\boldsymbol{u}^*}\|\big| \leq \|X_{\boldsymbol{u_n}} - X_{\boldsymbol{u}^*}\|$), we get:

$$\|X_{\boldsymbol{u_n}}\|,\ \|X_{\boldsymbol{v_n}}\| \overset{n\to\infty}{\longrightarrow} \|X_{\boldsymbol{u}^*}\|\,. \tag{1.42}$$

Hence, the limit as n tends to infinity of the following identity:

$$\mathbb{E}\big[\,|X_{\boldsymbol{u_n}} - X_{\boldsymbol{v_n}}|^2\,\big] \;=\; \mathbb{E}\big[X^2_{\boldsymbol{u_n}}\big] + \mathbb{E}\big[X^2_{\boldsymbol{v_n}}\big] - 2\mathbb{E}\big[X_{\boldsymbol{u_n}} X_{\boldsymbol{v_n}}\big]\,, \tag{1.43}$$

becomes as follows:

$$0 \;=\; \mathbb{E}\big[X^2_{\boldsymbol{u}^*}\big] + \mathbb{E}\big[X^2_{\boldsymbol{u}^*}\big] - 2\lim_{n\to\infty} C\big(\boldsymbol{u_n}, \boldsymbol{v_n}\big)\,. \tag{1.44}$$

Recalling that $\mathbb{E}[X^2_{\boldsymbol{u}^*}] = C(\boldsymbol{u}^*, \boldsymbol{u}^*)$, we have shown that (1.39) holds, as it was to be proved. □

An important class of covariance functions are stationary covariance functions, widely used in GP emulation and more generally in Machine Learning. We say that a covariance function $C(\cdot,\cdot)$ on $\mathcal{P}\times\mathcal{P}$ is stationary if there exists a function $k\colon \mathbb{R}^p \to \mathbb{R}$ such that

$$C(\boldsymbol{u},\boldsymbol{v}) = k(\boldsymbol{u}-\boldsymbol{v}) \quad \forall\, \boldsymbol{u},\boldsymbol{v}\in\mathcal{P}. \tag{1.45}$$

The function $k(\cdot)$ is often called the kernel function. We will commonly denote its argument by $\boldsymbol{h}\in\mathbb{R}^p$. Notice that, by first centering the process X and by then considering the trivial inequality $\mathrm{Var}[X_{\boldsymbol{u}} - X_{\boldsymbol{v}}] \geq 0$, one immediately verifies that

$$k(\boldsymbol{h}) \leq k(\boldsymbol{0}) \quad \forall\, \boldsymbol{h}\in\mathbb{R}^p. \tag{1.46}$$

In the case of a stationary covariance function, Proposition 1.4.5 can be immediately rephrased as follows.

Corollary 1.4.6 (of Proposition 1.4.5)**.** *Let X be a second-order process with continuous mean function, and stationary covariance function with kernel $k(\cdot)$. The process is MS continuous at any $\boldsymbol{u}\in\mathcal{P}$ if and only if $k(\cdot)$ is continuous at $\boldsymbol{h}=\boldsymbol{0}$.*

Notice that MS continuity is not equivalent to continuity of the process paths, as functions of $\boldsymbol{u}\in\mathcal{P}$ for fixed $\omega\in\Omega$. In a general setting, none of the two types of continuity implies the other. In the stochastic literature, results are available that guarantee pathwise continuity under appropriate conditions on the covariance function. For stationary processes, these conditions are often formulated in terms of polynomial

and logarithmic bounds on the stationary kernel $k(\cdot)$ of equation (1.45). For completeness, we provide below two examples of such conditions. The first result is due to John Kent and holds for generic stationary process. Under such general assumptions, the author shows that the condition can hardly be weakened.

Theorem 1.4.7 (Kent [1989]). *Let X be a zero-mean stationary stochastic process, with parameter space $\mathcal{P} \subseteq \mathbb{R}^p$. Suppose the kernel $k(\cdot)$ of the covariance function of X is p times continuously differentiable at $\boldsymbol{h} = \mathbf{0}$. Denote by $T_p(\boldsymbol{h})$ the multivariate Taylor polynomial of order p of $k(\cdot)$ around $\boldsymbol{h} = \mathbf{0}$, and by $\sigma_p(\cdot)$ the remainder:*

$$\sigma_p(\boldsymbol{h}) = k(\boldsymbol{h}) - T_p(\boldsymbol{h})\,. \tag{1.47}$$

If there exists $\gamma > 0$ such that

$$\big|\sigma_p(\boldsymbol{h})\big| = O\left(\frac{r^p}{|\log r|^{3+\gamma}}\right) \qquad \text{as} \quad r = \|\boldsymbol{h}\| \to 0\,, \tag{1.48}$$

then the process has almost surely (i.e., with probability one) continuous realisations.

For completeness, let us briefly recall the meaning of the "big O" notation. Given two real function $f(\cdot)$ and $g(\cdot)$ both defined in a neighbourhood of $\boldsymbol{x}^* \in \mathbb{R}^p$, we say that f is "big O" of g as $\boldsymbol{x}$ tends to $\boldsymbol{x}^*$ if the following holds:

$$\lim_{\boldsymbol{x}\to\boldsymbol{x}^*} \left| \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \right| < \infty\,. \tag{1.49}$$

Condition (1.49) assumes the limit exists. Otherwise, replace lim with lim sup.

The second result that we present only holds for Gaussian processes. The Gaussianity assumption allows to provide a weaker condition than (1.48) under which path continuity is guaranteed.

Theorem 1.4.8 (Adler [1981], Theorem 3.4.1). *Let X be a zero-mean Gaussian process with parameter space $\mathcal{P} \subseteq \mathbb{R}^p$ and continuous stationary covariance function $C(\boldsymbol{u}, \boldsymbol{v}) = k(\boldsymbol{u} - \boldsymbol{v})$. If there exists $C > 0$ and $\varepsilon > 0$ such that*

$$k(\mathbf{0}) - k(\boldsymbol{h}) \leq \frac{C}{\big|\log \|\boldsymbol{h}\|\big|^{1+\varepsilon}} \qquad \forall\, \boldsymbol{h} \in \mathcal{P}, \tag{1.50}$$

then the process has almost surely continuous paths in $\mathcal{P}$.

Further results relating the regularity of the covariance function to the one of the process can be found in the literature (see, for example, Theorem 3.3.2 and equations (3.4.1), (3.4.2) in Adler [1981]). Extensions to the non-stationary case are also available. However, as Theorem 1.4.7 and Theorem 1.4.8 show, sufficient conditions for pathwise continuity are generally more complex to check than sufficient conditions for MS continuity; moreover, they are not as general as the last ones. In the following, we only state results guaranteeing higher order regularity of the process in the MS metric, and provide appropriate references for the pathwise analogous results.

1.4.3. Connection to Mean-Square Differentiability of Any Order

In this section, we extend the results of Subsection 1.4.2 to derivatives of first and higher order. To this aim, we first define the concept of MS differentiability.

Definition 1.4.9. Let X be a stochastic process with open parameter space $\mathcal{P} \subseteq \mathbb{R}^p$, and fix $i \in \{1, \dots, p\}$. We say that a random variable Y is the MS derivative of X at $\boldsymbol{u}^* \in \mathcal{P}$ in the i^{th} direction, if the following holds:

$$\lim_{h \to 0} \mathbb{E}\left[\left| \frac{X_{\boldsymbol{u}^* + h\boldsymbol{e_i}} - X_{\boldsymbol{u}^*}}{h} - Y \right|^2 \right] = 0\,, \tag{1.51}$$

where the j^{th} component of $\boldsymbol{e_i} \in \mathbb{R}^p$ equals δ_{ij}, for $j = 1, \dots, p$. In this case, we write $Y = \partial X / \partial u_i(\boldsymbol{u}^*)$.

The previous definition can be extended to higher order derivatives, see for example Adler and Taylor [2007, § 1.4.2]. An analogue of Proposition 1.4.5 for derivatives of first or higher order can be found in Rasmussen and Williams [2006, § 4.4.1]. For simplicity, we formulate this result only in the case of stationary covariance functions, this being the case of main interest in the present work.

Theorem 1.4.10. *Let X be a zero-mean, second-order process with parameter space $\mathcal{P} \subseteq \mathbb{R}^p$. Suppose X has stationary covariance function with kernel $k(\cdot)$, and fix $q \in \mathbb{N}$*

and any q *coordinates* $(u_{i_1}, \dots, u_{i_q}) \in \{u_1, \dots, u_p\}^q$. *Then, the following MS partial derivative of* X *at any* $\boldsymbol{u} \in \mathcal{P}$ *exists:*

$$\frac{\partial^q X}{\partial u_{i_1} \cdots \partial u_{i_q}}(\boldsymbol{u}) \,, \tag{1.52}$$

if and only if the corresponding partial derivative of order $2q$ *of* $k(\cdot)$ *at* $\boldsymbol{h} = \boldsymbol{0}$ *exists:*

$$\frac{\partial^{2q} k}{\partial h_{i_1}^2 \cdots \partial h_{i_q}^2}(\boldsymbol{0}) \tag{1.53}$$

For example, if we take $q = 1$, we get that the process is MS differentiable in all directions, if and only if the kernel $k(\cdot)$ possesses all second-order derivatives with respect to the single variables: $\partial^2 k / \partial h_i^2$. We refer the reader to Adler and Taylor [2007, Thm 1.4.2] for additional hypotheses under which, in the Gaussian case, MS q^{th}-order differentiability implies q^{th}-order differentiability of the sample paths.

To sum up, Corollary 1.4.6 and Theorem 1.4.10 state that, for stationary processes, the regularity of the covariance function in $\boldsymbol{0}$ directly affects the MS regularity of the process. As we have observed in Subsection 1.4.2, the former also affects the regularity of the process' paths. In Figure 1.5, we provide an example of sample trajectories of zero-mean Gaussian processes with different covariance functions. These are introduced in the following section: they are characterised by different levels of regularity, which is reflected into the regularity of the resulting sample trajectories shown in the figure.

1.4.4. Important Families of Covariance Functions

In this section we present families of covariance functions commonly employed in applications, especially within GP emulation. In Part II of this work, we discuss their use in two different climate reconstruction problems.

We present families of covariance functions whose value on a pair $(\boldsymbol{u}, \boldsymbol{v})$ only depends on the length of the vector $\boldsymbol{u} - \boldsymbol{v}$. That is, we have:

$$C(\boldsymbol{u}, \boldsymbol{v}) = k(\|\boldsymbol{u} - \boldsymbol{v}\|) \quad \forall\, \boldsymbol{u}, \boldsymbol{v} \in \mathcal{P} \,, \tag{1.54}$$

for some function $k\colon [0,\infty) \to \mathbb{R}$. Although, technically, this function is different from the one of equation (1.45) (their domains are different), we still call it kernel function: the assumption that $C(\boldsymbol{u},\boldsymbol{v})$ only depends on $\|\boldsymbol{u}-\boldsymbol{v}\|$ naturally makes $k(\cdot)$ a function of positive real numbers, rather than p-dimensional vectors. The argument of $k(\cdot)$, representing the norm of a vector, will generally be denoted by the letter r.

The norm $\|\cdot\|$ can be imagined to be the Euclidean norm in $\mathbb{R}^p$. Nothing, however, prevents other norms from being used. In the following, we suppose that the map $\boldsymbol{h} \mapsto \|\boldsymbol{h}\|^2$ is C^∞ in $\boldsymbol{h}=\boldsymbol{0}$, in order for the comments on regularity which will follow to hold; the Euclidean norm, and more generally the norms introduced in Section 1.5, clearly satisfy this property. We relate the choice of the norm in (1.54) to the concept of correlation lengths, in Section 1.5.

Squared Exponential

The squared exponential (or Gaussian) kernel is defined as follows:

$$k(r) = \exp\left(-\frac{r^2}{2}\right), \quad r \geq 0\,. \tag{1.55}$$

This is one of the most commonly employed kernels. The function in (1.55) is infinitely many times differentiable at $r=0$, also as a function of $\boldsymbol{h} \in \mathbb{R}^p$ when $r=\|\boldsymbol{h}\|$. Hence, a zero-mean stochastic process with this covariance function will be infinitely many times MS differentiable (provided that $\boldsymbol{h} \mapsto \|\boldsymbol{h}\|^2$ is C^∞ in $\boldsymbol{h}=\boldsymbol{0}$).

Matérn Family

The Matérn kernel depends on a positive parameter ν. It is defined as follows:

$$k_\nu(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}r\right)^\nu K_\nu\left(\sqrt{2\nu}r\right), \tag{1.56}$$

where $\Gamma(\cdot)$ is the Gamma function, and $K_\nu(\cdot)$ is the modified Bessel function of second kind (Abramowitz and Stegun [1970, Section 9.6.1]). Its properties are not immediately evident from the expression above. However, it is known that a Gaussian process with Matérn covariance with parameter ν, is q times differentiable both in mean square and pathwise if and only if $\nu > q$. See Santner et al. [2003, Section 2.3.4, Example 2.5], and Rasmussen and Williams [2006, Section 4.2]).

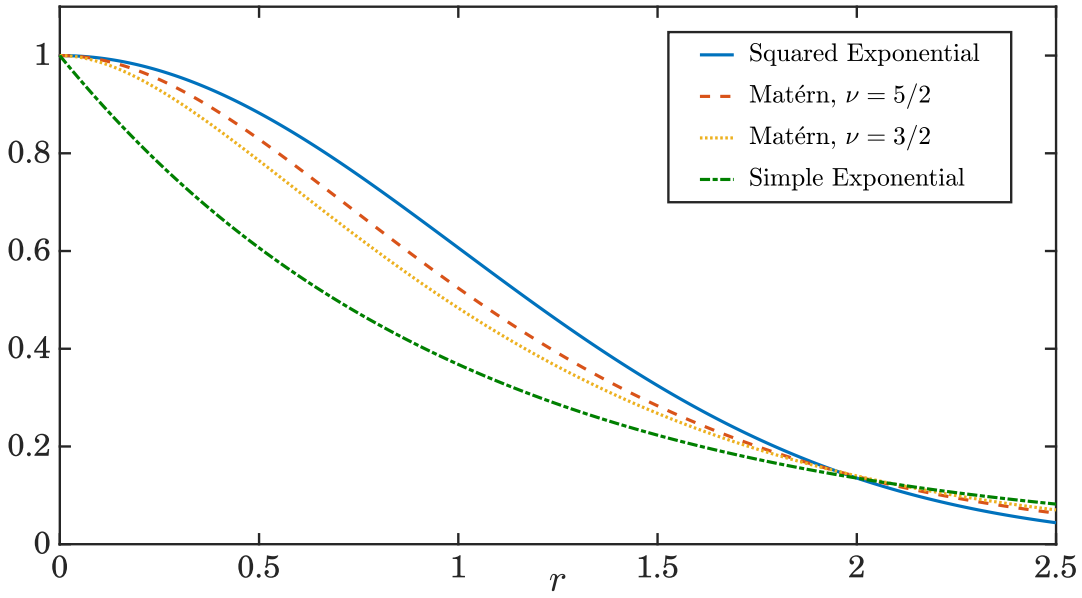


Figure 1.4: Plot of the covariance kernels presented in Subsection 1.4.4. The first three (squared exponential and Matérn, for $\nu > 1$) have zero derivative in $r = 0$, hence they are differentiable in 0 also as a function of $\boldsymbol{h}$ when $r = |\boldsymbol{h}|$; the simple exponential is not.

Expression (1.56) considerably simplifies when the parameter ν is half-integer, $\nu = n + 1/2$, $n \in \mathbb{N}$. In particular, for $n = 1$ and $n = 2$, we get:

$$k_{3/2}(r) = \left(1 + \sqrt{3}r\right) \exp\left(-\sqrt{3}r\right), \tag{1.57}$$

$$k_{5/2}(r) = \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp\left(-\sqrt{5}r\right). \tag{1.58}$$

According to the previous result, a zero-mean Gaussian process with covariance function (1.57) or (1.58) is differentiable, respectively, once or twice, both pathwise and in mean square. As a side note, let us observe that the MS differentiability can also be derived from Theorem 1.4.10, by checking the existence of

$$\frac{\partial^2 g}{\partial u_i^2}(\mathbf{0}) \quad \text{and} \quad \frac{\partial^4 g}{\partial u_i^2 \partial u_j^2}(\mathbf{0}),$$

for the function $g(\boldsymbol{h}) = k(\|\boldsymbol{h}\|)$, $\boldsymbol{h} \in \mathbb{R}^p$.

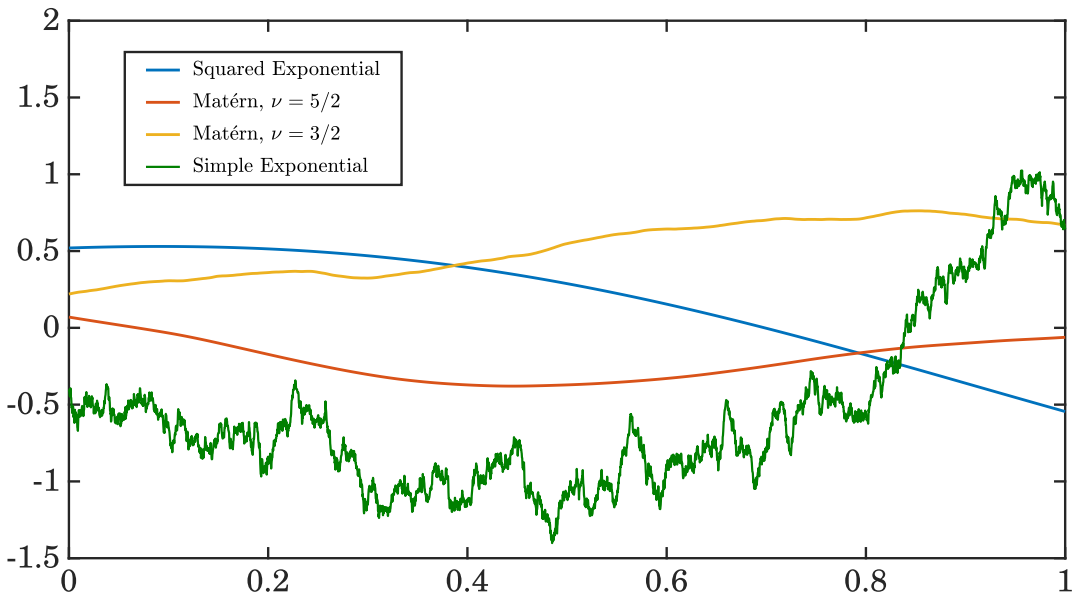


Figure 1.5: Sample trajectories of zero-mean GPs with different covariance functions. The absolute exponential yields continuous but not differentiable paths (green line). The other three covariance functions yield higher, but different, levels of path (and mean-square) regularity. In particular: C^1 for Matérn $\nu = 3/2$, yellow line; C^2 for Matérn $\nu = 5/2$, red line; C^∞ for squared exponential, blue line.

Simple Exponential

The simple (or absolute) exponential kernel is defined as follows:

$$k(r) = \exp(-r). \tag{1.59}$$

It is clear that the function $k(\|\boldsymbol{h}\|)$ for $\boldsymbol{h} \in \mathbb{R}^p$ is continuous but not differentiable in $\boldsymbol{h} = \mathbf{0}$. Hence, a process with this covariance function is MS continuous, but not MS differentiable. This also holds pathwise: it can be seen, in fact, that the simple exponential kernel is the Matérn kernel corresponding to $\nu = 1/2$. However, we have presented it separately due to its importance and simple expression. This is, for example, the covariance function of the Ornstein-Uhlenbeck process, a famous Gaussian process obtained as solution to a linear stochastic differential equation.

Figure 1.4 shows the plot of the different kernels presented in this section. One-dimensional sample trajectories of GPs with these covariance functions are instead shown in Figure 1.5: it can be appreciated that the trajectories have different

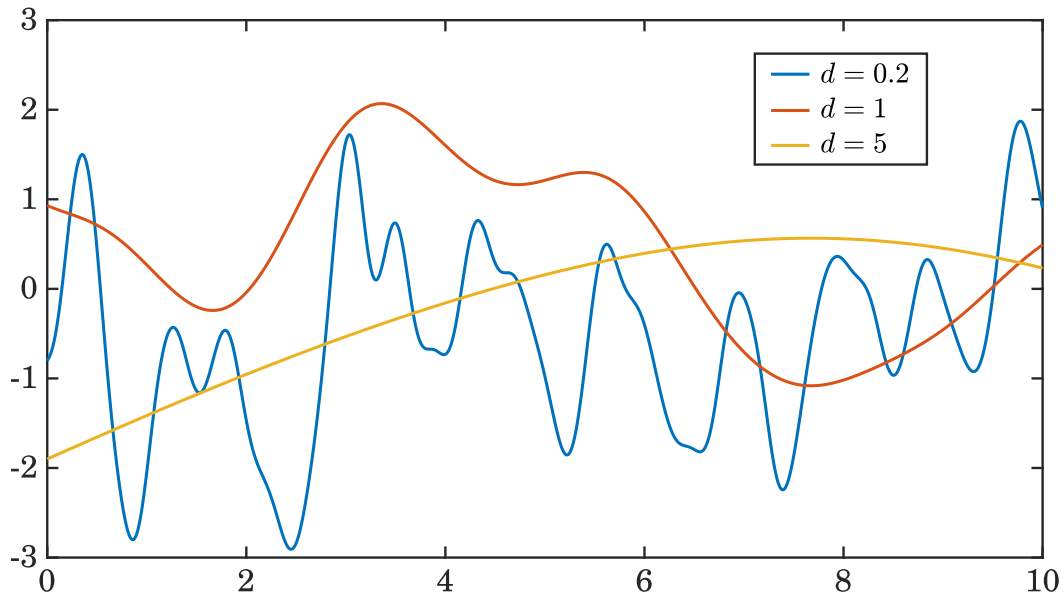


Figure 1.6: GP trajectories corresponding to different correlation lengths. The squared exponential kernel has been used. Within an input interval of length 10, a trajectory with correlation length $d = 0.2$ usually displays numerous fluctuations. These decrease consistently for $d = 1$ and even more for $d = 5$. In the last case, the correlation between outputs at the endpoints of the interval is still non-negligible.

regularity, according to what the regularity of the function $k(\|\boldsymbol{h}\|)$ in $\boldsymbol{h} = \boldsymbol{0}$ is.

1.5. Correlation Lengths

If the norm used in equation (1.54) is the Euclidean norm, the resulting covariance function is invariant under rotations. That is, it satisfies:

$$C(\boldsymbol{R}\boldsymbol{u}, \boldsymbol{R}\boldsymbol{v}) = C(\boldsymbol{u}, \boldsymbol{v})\,, \tag{1.60}$$

for any rotation $\boldsymbol{R}$ of $\mathbb{R}^p$ and any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^p$. This immediately follows from the property $\|\boldsymbol{R}\boldsymbol{h}\| = \|\boldsymbol{h}\|$, valid for any orthogonal transformation when the employed norm is the Euclidean one. Covariance functions satisfying (1.60) are therefore convenient in applications where no special role is attached to the different axes. For this reason, they are often referred to as isotropic.

In many applications, however, the input parameters of the stochastic process of

interest are characterised by different scales along the different dimensions. If a covariance function of the form (1.54) is used, it is then convenient to consider a norm that treats distances in the different dimensions accordingly. Here, for positive $d_1, \ldots, d_p$, we consider the following norm:

$$\|\boldsymbol{h}\|_{\boldsymbol{d}} = \sqrt{\sum_{j=1}^{p} \left(\frac{h_i}{d_j}\right)^2}, \quad \boldsymbol{h} \in \mathbb{R}^p. \tag{1.61}$$

The vector $\boldsymbol{d}$ is the vector with components d_j. Hence, for a given kernel $k(\cdot)$, the associated covariance function reads as follows:

$$C(\boldsymbol{u}, \boldsymbol{v}) = k\big(\|\boldsymbol{u} - \boldsymbol{v}\|_{\boldsymbol{d}}\big), \quad \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^p. \tag{1.62}$$

The quantity d_j is a measure of how far apart from each other two inputs $\boldsymbol{u}$ and $\boldsymbol{v}$ need to be along the j^{th} dimension, in order for the covariance function (1.62) to decrease significantly. We refer to the d_j as correlation lengths: the higher they are, the smaller the quantity $r = \|\boldsymbol{u} - \boldsymbol{v}\|_{\boldsymbol{d}}$ is, and therefore the more correlated the outputs corresponding to the inputs $\boldsymbol{u}$ and $\boldsymbol{v}$ will be.

In Figure 1.6 trajectories of a zero-mean GP with squared exponential covariance function are shown in one dimension, for different values of the only correlation length $d = d_1$. It can be appreciated that trajectories with higher correlation length tend to "remember" more about their past, while trajectories with small correlation length show an oscillatory behaviour even within small input intervals. In practice, choosing appropriate values of the correlation lengths is not an easy task. We show methods to estimate these in Chapter 4 and Chapter 5.

2. Gaussian Process Emulation

Abstract: This chapter provides a detailed presentation of univariate Gaussian process emulation. The formulas commonly used in the relevant literature, which is reviewed at the beginning of the chapter, are derived within a formal probabilistic setting. Our presentation requires the introduction of ad-hoc and sometimes technical results, which are stated and proved. In parallel, different diagrams illustrate the various steps dealt with. The choice classically made in the literature is also discussed. In the final section, the setting is extended to the one where the observed simulator outputs come with associated uncertainty and the potential for an identifiability issue is discussed.

2.1. Introduction

In Section 1.1, we have provided an overview of the context and motivations leading to the birth of Gaussian process (GP) emulation. This chapter is now devoted to a detailed presentation of its theory. The framework and all relevant results are derived in Sections 2.2–2.5: more details on the content of these sections are provided at the end of Section 2.2, once the relevant notation has been introduced. Section 2.6 illustrates the classical choice adopted in applications. After schematically summarising the overall GP framework and formulas in Section 2.7, we consider in Section 2.8 the case of chaotic simulators and a correction term in the emulator that is relevant to deal with such simulators.

Before presenting the theory, in Subsection 2.1.1 we provide a review of the most relevant work in the field. The content of the subsequent sections is obviously inspired by the pioneering work of some of the authors mentioned therein. However, the mathematical formulation used and all proofs have been entirely developed by the author of this work. As such, some ideas differ from the ones found within the classical works. It has been the author's aim to systematically present the theory and to justify all the steps yielding the formulas commonly used in applications.

2.1.1. Literature Review

As remarked in Section 1.1, the field of GP emulation was born to face the problem of the analysis of complex simulators. We establish here some basic notation, helpful in the following review and more generally needed throughout the chapter. We view the simulator as a function:

$$\begin{aligned} f\colon \mathcal{P} &\longrightarrow \mathbb{R} \\ \boldsymbol{x} &\mapsto f(\boldsymbol{x})\,, \end{aligned} \tag{2.1}$$

where $\mathcal{P} \subseteq \mathbb{R}^p$ is the set of valid inputs. An emulator can then be described as a stochastic interpolator of $f(\cdot)$, modelled as a GP and whose prior distribution is updated on the basis of the observed simulator outputs at a sequence of inputs in $\mathcal{P}$. Notice that the output of the simulator is here assumed one-dimensional only for convenience; the case of multi-dimensional output will be treated in Chapter 3.

Identifying a single, specific work as the one marking the birth of the GP emulation field is not an easy task. One of the most acclaimed papers in the context of fitting a cheaper predictor to the outputs of computer codes is the one of Sacks et al. [1989]. The paper is however mostly based on a frequentist framework, which uses Gaussian processes to describe the behaviour of the simulator. In fact, the work also makes a comparison with the so-called kriging approach, where a linear combination of the outputs y_i is used to predict the value $f(\boldsymbol{x})$ for unknown $\boldsymbol{x} \in \mathcal{P}$. The kriging method is especially used in geostatistics, where the input space $\mathcal{P}$ is usually two- or three-dimensional. See the book Cressie [1993]. The Bayesian approach is instead emphasised in a subsequent paper of Currin et al. [1991], which also compares the setting to the numerical analysis one of linear and cubic splines. Connections between the use of Gaussian processes in interpolation and classical numerical analysis problems such as quadrature and optimisation are also discussed in O'Hagan [1992].

Tony O'Hagan is generally considered (one of) the scientific father(s) of GP emulation. The work O'Hagan [1978] may be recognised an early precursor of the later development of GP emulation: in the paper, the author uses a Bayesian approach based on GPs to make inference on the regression function of a statistical model. The works mentioned in the previous paragraph, alongside others, may be seen as an extension of this setting to the one where the regression function is replaced by a general mathematical function, or a computer code. Curiously enough, however, none of these works explicitly referred to the statistical model developed as to an emulator.

Between the end of the 1990's and the first years of the 2000's, numerous pieces of work were produced that focused on the uncertainties associated with the inference on computer codes (Haylock and O'Hagan [1996], Oakley and O'Hagan [2002], Oakley and O'Hagan [2004]). The works gave rise to the Uncertainty Quantification (UQ) field and the word "emulator" became a classical term. It is worth mentioning that not all the works on emulators are based on GPs. A parallel approach is the Bayes Linear one: as opposed to the more classical full Bayesian approach, developed by O'Hagan, Oakley etc, the Bayes linear approach only relies on the specification of first and second order moments of prior distributions. Papers such as Craig et al. [2001], Goldstein and Rougier [2006] or Cumming and Goldstein [2010] are eminent examples of the works in the area of Bayes linear emulators. For a broader introduction to the

principles and applications of the Bayes linear approach, we refer the reader to the book Goldstein and Wooff [2007].

With the birth of the UQ field, and especially within the context of computer codes, the name *uncertainty analysis* started to be used with a precise meaning: specifically, to denote the study of the propagation of uncertainty from the inputs to the outputs of a computer simulator, when the "true/best" values of the input parameters are unknown (Haylock and O'Hagan [1996], O'Hagan et al. [1998]). In this context, the name uncertainty distribution refers to the distribution induced by the simulator on the outputs, given the uncertainty in the inputs: see Oakley and O'Hagan [2002].

The study of the different uncertainties associated with the analysis of computer simulators is a topic of key practical importance, especially when decisions have to be made on the basis of the analysis. The topic has in fact attracted the attention of numerous influential authors. In their highly popular and cited paper, Kennedy and O'Hagan [2001] make the effort to systematically classify the different sources of uncertainty that affect a computer simulator analysis. Besides the so-called code uncertainty, already introduced in Subsection 1.1.2, the authors identify sources of uncertainty such as the parameter uncertainty (reflecting the lack of knowledge of best inputs), the residual variability (variability due to factors not explicitly included in the input configurations), and others. In particular, their work studies the uncertainties related to what they call "model inadequacy", or model discrepancy: that is, the difference between the value predicted by a simulator and the value that the real physical process would take under the conditions specified as inputs to the simulator.

Model discrepancy is not an easy issue to tackle, and it has been target, indeed, of extensive investigation; in the end, any analysis aims to make robust inference on reality, rather than on the sheer simulator dynamic. Model discrepancy is usually accounted for via an additional term within the emulator model. This is the case of the already mentioned paper Kennedy and O'Hagan [2001], but also of other works such as Goldstein and Rougier [2004]. Here, the authors construct a probabilistic framework which has the aim to link the physical system of interest to the results obtained via one or more simulators.

We mention here that the idea of using more than one simulator to study the same system has also been developed in a slightly different context, and with a slightly

different aim. The context is the one where either a "coarse" (Cumming and Goldstein [2009]) and cheaper version of a given simulator is available, or separate simulators characterised by different levels of complexity are available, to study the same system. See for example Kennedy and O'Hagan [2000]. In either case, different emulators are built and the information coming from the simpler and faster-to-run models is "passed on" to the emulators of the more complex simulators. This way, a hierarchy of emulators is built, known as multi-level emulators. An example of application to the climate system is provided in Tran et al. [2016].

Coming back to the central idea of linking the simulator to reality, we briefly touch upon a key idea, that we will later employ in Chapter 5. This is the one of history-matching. The idea consists in using real-world observations to identify input configurations of a computer simulator whose corresponding outputs best match the observed data. In such a case, it may be useful to incorporate model-discrepancy in the emulator of the simulator of interest. The term history-matching originally comes from the oil industry (see Mattax and Dalton [1990] or Craig et al. [1997]), but is now used to refer to what we have just described, independently of the field of application. See for example Vernon et al. [2010] for an application within the fascinating context of Galaxy formation.

The standard approach of GP emulation has been developed and mostly used for computer simulators depending on a number of continuous inputs, and one or more outputs. In 2009, Conti et al. [2009] extended this framework to the one of dynamic simulators: that is, simulators whose outputs are time series, provided at a sequence of discrete times. The authors tackle the problem of emulating such simulators in an iterative way, by developing a framework and an algorithm that does not need to know a priori up to which time the time series should be emulated. This represents a clear improvement with respect to the case where a fixed-length time series is more classically treated as a multi-dimensional output.

We point out here another aspect of the classical emulation approach: it constructs a statistical model, the emulator, that perfectly interpolates the observed simulator outputs at specific inputs $\boldsymbol{x_i} \in \mathcal{P}$ called design points. An extension to stochastic simulators is proposed in Johnson et al. [2011]. In both deterministic and stochastic cases, it is relatively common to include a so-called nugget term within an emulator,

which allows to build a model not interpolating the observed data. The effect of the nugget term on GP models is investigated in detail in Andrianakis and Challenor [2012]. In their work, the authors also examine the improved numerical stability that its use yields. We will investigate the use of a nugget term in Section 2.8, specifically proposing its use within the context of chaotic simulators, and by deriving the relevant formulas via continuity arguments.

We conclude this review with an important remark. After building an emulator, a crucial step before this can be used as "surrogate" of the simulator consists in its validation: that is, in checking that the emulator predictions are reasonable approximations of the simulator outputs. Bastos and O'Hagan [2009] is the seminal article on the topic. It provides a comprehensive list of diagnostics, of both numerical and graphical nature, to identify potential problems with different aspects of an emulator.

2.2. Two-Level Hierarchical Model

As stated in equation (2.1), in this chapter we consider a simulator with one-dimensional outputs:

$$\begin{aligned} f\colon \mathcal{P} &\longrightarrow \mathbb{R} \\ \boldsymbol{x} &\mapsto f(\boldsymbol{x})\,. \end{aligned} \tag{2.2}$$

The set $\mathcal{P} \subseteq \mathbb{R}^p$ is the set of valid inputs to the simulator. As the previous equation implicitly reveals, we assume – at least in this chapter – that the simulator is deterministic: running the code twice on the same input $\boldsymbol{x} \in \mathcal{P}$ will return both times the same output. Throughout the chapter, we also assume that the simulator has been run on a sequence of "design points" $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \in \mathcal{P}$, and denote with $y_i = f(\boldsymbol{x_i})$ the corresponding outputs. Examples of how to choose the design points will be discussed in Part II of this thesis. For a more comprehensive treatment, see Chapters 5 and 6 of Santner et al. [2003].

Following the classical literature (Currin et al. [1991], O'Hagan [1992]), we model the simulator $f(\cdot)$ as a GP $\eta(\cdot)$, with the same input space $\mathcal{P}$ of the simulator itself. In the author's view, the choice of modelling the simulator as a stochastic process represents

a way to express the uncertainty that is inherently attached to the simulator dynamic. Indeed, whilst we assume the simulator to be deterministic, it is also the case that this represents most often a black box to the modeller/experimenter. The value $f(\boldsymbol{x})$ will be completely unknown, till the simulator is actually run on $\boldsymbol{x}$. Within a Bayesian terminology, we can say that modelling the simulator as a stochastic process allows us to express (not-too-strong) "beliefs" on the simulator, which can then be "updated" in the light of the observed outputs y_i, $i = 1, \ldots, n$.

Using the notation of Chapter 1, we consider an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and represent the GP $\eta(\cdot)$ with input space $\mathcal{P}$ as follows:

$$\eta \colon \big(\Omega, \mathcal{F}, \mathbb{P}\big) \times \mathcal{P} \;\longrightarrow\; \big(\mathbb{R}, \mathcal{B}(\mathbb{R})\big)\,. \tag{2.3}$$

Let us recall that $\mathcal{B}(\mathbb{R})$ denotes the Borel σ-algebra on $\mathbb{R}$. As in the classical literature, we assume that the mean function of $\eta(\cdot)$ is an unknown linear combination of q known functions of the inputs. That is, we consider the mean function:

$$m_\beta(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T \boldsymbol{\beta}\,, \quad \boldsymbol{x} \in \mathcal{P}, \tag{2.4}$$

where $\boldsymbol{\beta} \in \mathbb{R}^q$ is the unknown vector of coefficients, and the set of q regression functions $\boldsymbol{h} \colon \mathcal{P} \to \mathbb{R}^q$ is specified according to the problem. Moreover, we model the covariance function of $\eta(\cdot)$ as follows:

$$v_{\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2\, c(\boldsymbol{x}, \boldsymbol{x}')\,, \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}\,, \tag{2.5}$$

where $c \colon \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ is any valid covariance function, and σ^2 is a scaling factor. Within the emulation literature, the function $c(\cdot, \cdot)$ is usually specified as a correlation function – *i.e.*, a valid covariance function satisfying $c(\boldsymbol{x}, \boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in \mathcal{P}$. In such a case, equation (2.5) implies that the process is being modelled as homoscedastic (same variance at all $\boldsymbol{x} \in \mathcal{P}$) with variance σ^2. However, as we show in the following, the theory of GP emulation holds true as long as $c(\cdot, \cdot)$ is any valid covariance function. This way, we can also model heteroscedastic processes via equation (2.5).

For fixed $\boldsymbol{\beta} \in \mathbb{R}^q$ and $\sigma^2 > 0$, we write that $\eta(\cdot)$ is a GP with mean function $m_\beta(\cdot)$

and covariance function $v_{\sigma^2}(\cdot,\cdot)$ via the following compact notation:

$$\eta(\cdot) \sim \mathcal{GP}\big(\, m_\beta(\cdot),\, v_{\sigma^2}(\cdot,\cdot)\,\big). \tag{2.6}$$

Within a standard frequentist approach, $\boldsymbol{\beta}$ and σ^2 would be interpreted as unknown constants. Their values would be usually estimated to maximise the likelihood of the data $\big\{\big(\boldsymbol{x_i}, y_i\big)\big\}_{i=1,\dots,n}$. The Bayesian approach at the basis of GP emulation instead assigns a prior distribution to $\boldsymbol{\beta}$ and σ^2, and conditions this to having observed the simulator outputs $y_i = f(\boldsymbol{x_i})$ for $i = 1, \dots, n$. This way, a model with a hierarchical structure is built: $\boldsymbol{\beta}$ and σ^2 are random quantities, commonly referred to as hyperparameters of the model; conditioned on their values, $\eta(\cdot)$ is a random process distributed as in (2.6). Hence, in the coming sections we proceed as follows:

i) In Section 2.3, we specify the prior distribution of $\boldsymbol{\beta}$ and σ^2, and formulate the problem in terms of a single (infinite-dimensional) random quantity.

ii) In Section 2.4, we condition the latter to the observations $y_i = f(\boldsymbol{x_i})$, $i = 1, \dots, n$.

iii) In Section 2.5 we uncondition the model on $\boldsymbol{\beta}$ and σ^2, and derive the marginal posterior for $\eta(\cdot)$.

Diagrams will illustrate the completion of each of these steps.

2.3. Prior Distribution of the Model

We start this section by a recap of univariate and multivariate probability distributions. All of these are continuous and admit a probability density. Particularly the Normal-Inverse-Gamma, introduced last, will be used throughout the chapter.

2.3.1. Recap of Useful Distributions

The following three distributions have support in $[0, \infty)$. After the expression of their densities, we review some of their properties of interest for us.

Chi-squared: A random variable V follows a chi-squared distribution with d degrees of freedom if it is the sum of the squares of d independent standard normal random variables. We write $V \sim \chi^2(d)$. Its density f_V reads as follows:

$$f_V(x) = \frac{1}{2^{d/2}\,\Gamma(d/2)}\, x^{\frac{d}{2}-1} e^{-x/2}, \quad x \geq 0\,. \tag{2.7}$$

Gamma: A random variable X follows a Gamma distribution with shape parameter $a > 0$ and rate parameter $r > 0$, if its density f_X is as follows:

$$f_X(x) = \frac{r^{\,a}}{\Gamma(a)}\, x^{a-1}\, e^{-rx}, \quad x \geq 0\,. \tag{2.8}$$

We write $X \sim \Gamma(a, r)$.

Inverse-Gamma: A random variable Y follows an Inverse-Gamma distribution with shape parameter $a > 0$ and scale parameter $s > 0$, if the random variable $1/Y$ is distributed as $\Gamma(a, s)$. We write $Y \sim IG(a, s)$, and have:

$$f_Y(x) = \frac{s^{\,a}}{\Gamma(a)}\, \frac{1}{x^{a+1}}\, e^{-s/x}, \quad x > 0\,. \tag{2.9}$$

The previous expression can be easily derived through the transformation formula for probability densities, applied to the function $g(x) = 1/x$ and the density (2.8). The formula is recalled for convenience in Lemma A.1, in Appendix.

The following are useful properties of the previous distributions. From the density transformation formula, it is immediate to see that a rescaled Gamma random variable is still Gamma distributed. More specifically:

$$X \sim \Gamma(a, r) \quad \Longleftrightarrow \quad \frac{1}{c}\, X \sim \Gamma(a, cr) \quad \forall\, c > 0\,. \tag{2.10}$$

This property justifies the name "rate" for r. Similarly, the name "scale" is appropriate for the parameter s of an Inverse-Gamma distribution, since from (2.10) we obtain:

$$Y \sim IG(a, s) \quad \Longleftrightarrow \quad c\,Y \sim IG(a, cs) \quad \forall\, c > 0\,. \tag{2.11}$$

Moreover, by comparing (2.7) and (2.8), we see that the Gamma distribution is a generalisation of the chi-squared distribution. Indeed, for integer d, we have:

$$V \sim \chi^2(d) \quad \Longleftrightarrow \quad V \sim \Gamma\left(\frac{d}{2}, \frac{1}{2}\right). \tag{2.12}$$

Finally, by combining properties (2.10) and (2.12), we see that any Gamma random variable can be written as a multiple of a chi-squared random variable:

$$X \sim \Gamma(a, r) \quad \Longleftrightarrow \quad X = \frac{V}{2r}, \text{ where } V \sim \chi^2(2a). \tag{2.13}$$

Equivalently:

$$Y \sim IG(a, s) \quad \Longleftrightarrow \quad Y = \frac{2s}{V}, \text{ where } V \sim \chi^2(2a). \tag{2.14}$$

We now introduce a new distribution, of particular relevance in Subsection 2.3.2. As opposed to the previous ones, this is multivariate.

Normal-Inverse-Gamma: Let $\boldsymbol{b} \in \mathbb{R}^q$, $\boldsymbol{B} \in \mathbb{R}^{q \times q}$ a symmetric, positive definite matrix, and $a, s > 0$ positive constants. A random vector $(\boldsymbol{\beta}, \sigma^2)$ with $\boldsymbol{\beta} \in \mathbb{R}^q$ and $\sigma^2 > 0$ follows a Normal-Inverse-Gamma (NIG) distribution with parameters $\boldsymbol{b}, \boldsymbol{B}, a, s$, if

$$\sigma^2 \sim IG(a, s) \quad \text{and} \quad \boldsymbol{\beta}|\sigma^2 \sim N(\boldsymbol{b}, \sigma^2 \boldsymbol{B}^{-1}). \tag{2.15}$$

In this case, we write $(\boldsymbol{\beta}, \sigma^2) \sim NIG(\boldsymbol{b}, \boldsymbol{B}, a, s)$.

The joint density of a NIG distribution can be worked out through the standard rule for conditional densities (see for example Jacod and Protter [2000, Chapter 12]):

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2) &= p(\boldsymbol{\beta}|\sigma^2)\, p(\sigma^2) \\ &= \frac{C}{(\sigma^2)^{a+1+q/2}} \exp\left[-\frac{1}{2\sigma^2}\left(2s + (\boldsymbol{\beta} - \boldsymbol{b})^T \boldsymbol{B} (\boldsymbol{\beta} - \boldsymbol{b})\right)\right], \end{aligned} \tag{2.16}$$

where C is the normalisation constant.

2.3.2. Prior Choice for the Hyperparameters β and σ^2

In this section we specify a prior distribution for the pair of hyperparameters $\boldsymbol{\beta} \in \mathbb{R}^q$ and $\sigma^2 > 0$. Conditioned on their value, the process $\eta(\cdot)$ is modelled as a GP (equation (2.6)).

As the notation used in Section 2.2 suggests, the choice we make for the prior of the pair $(\boldsymbol{\beta}, \sigma^2)$ is the one of a NIG. This follows the idea developed in O'Hagan [1992], underlying most of the applications of GP emulation where a hierarchical model as the one is (2.17) is used[1]. We notice that the choice of a NIG prior has the main advantage of making the subsequent inference analytically tractable. In particular, as we detail in Section 2.4 (specifically, Subsection 2.4.2), it allows to carry out a conjugate analysis, *i.e.*, a Bayesian analysis where the prior and the posterior distributions belong to the same family. We also notice that the choice of a NIG distribution allows, in principle, to cover a variety of density shapes, given the relatively large number of parameters that this involves (*i.e.*, $\boldsymbol{b}, \boldsymbol{B}, a$ and s). However, an estimation of these parameters is challenging in most applications, hence a "default" choice is usually considered. We discuss this in Section 2.6.

At present, let us consider the parameters $\boldsymbol{b}, \boldsymbol{B}, a$ and s as fixed. To remark that they refer to the prior, we add the subscript 0. Thus, from (2.6), we have:

$$\eta(\cdot) \,|\, \boldsymbol{\beta}, \sigma^2 \sim \mathcal{GP}\big(\, m_\beta(\cdot),\, v_{\sigma^2}(\cdot,\cdot)\,\big), \tag{2.17.a}$$

$$(\boldsymbol{\beta}, \sigma^2) \sim NIG(\boldsymbol{b_0}, \boldsymbol{B_0}, a_0, s_0). \tag{2.17.b}$$

The expression of the functions m_β and v_{σ^2} is given in (2.4) and (2.5).

Let us now denote by $\mathbb{R}^{\mathcal{P}}$ the space of functions from $\mathcal{P}$ to $\mathbb{R}$, to which all trajectories of the process $\eta(\cdot)$ belong. Equations (2.17) can be seen as specifying the prior distribution of a random "variable" leaving in a product space, that is:

$$\big(\eta(\cdot), \boldsymbol{\beta}, \sigma^2\big) \colon (\Omega, \mathcal{F}, \mathbb{P}) \longrightarrow \big(\mathbb{R}^{\mathcal{P}} \times \mathbb{R}^q \times \mathbb{R}^+, \mathcal{B}\big), \tag{2.18}$$

[1] The NIG choice is rarely mentioned explicitly. Most works however refer to a "non-informative" choice, proportional to σ^{-2}: this follows from the present one, as we shall discuss in Section 2.6.

where $\mathcal{B}$ is the Borel σ-algebra on $\mathbb{R}^{\mathcal{P}} \times \mathbb{R}^q \times \mathbb{R}^+$. The main aim of this chapter is to condition the prior distribution of (2.18) to the observed simulator outputs, and to extract the marginal posterior for $\eta(\cdot)$. This is done, respectively, in Section 2.4 and Section 2.5. Those two steps, alongside the one that this section has carried out, are schematically represented in Figure 2.1. A similar diagram will be updated at the end of each of the following two sections, to provide a visual summary of the key step that each of these sections is concerned with.

2.3.3. Shorthand Notation Used in the Chapter

We conclude this Section 2.3 by setting some notation, used in the remainder of the chapter. We denote by Y_i the random variable obtained upon evaluation of the GP $\eta(\cdot)$ at the design point $\boldsymbol{x_i}$, and by $\boldsymbol{Y} \in \mathbb{R}^n$ the random vector with components Y_i:

$$\boldsymbol{Y} = (Y_1, \ldots, Y_n) \in \mathbb{R}^n, \qquad Y_i = \eta(\boldsymbol{x_i}) \;\text{ for }\; i = 1, \ldots, n. \tag{2.19}$$

Similarly, for generic $\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} \in \mathcal{P}$, we set:

$$\widetilde{\boldsymbol{Y}} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_k) \in \mathbb{R}^k, \qquad \widetilde{Y}_j = \eta(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \;\text{ for }\; j = 1, \ldots, k. \tag{2.20}$$

Moreover, we denote by $\boldsymbol{y} \in \mathbb{R}^n$ the vector of observed simulator outputs:

$$\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathbb{R}^n, \qquad y_i = f(\boldsymbol{x_i}) \;\text{ for }\; i = 1, \ldots, n. \tag{2.21}$$

Finally, throughout the chapter, we use the index $i \in \{1, \ldots, n\}$ to refer to quantities associated with the n design points $\boldsymbol{x_i} \in \mathcal{P}$, and the index $j \in \{1, \ldots, k\}$ to refer to quantities associated with the generic points $\widetilde{\boldsymbol{x}}_{\boldsymbol{j}} \in \mathcal{P}$.

In terms of the above notation, given the generality of $\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} \in \mathcal{P}$, the aim of finding the marginal posterior distribution of $\eta(\cdot)$ given the simulator outputs at the design points can be restated as finding the following conditional law:

$$\mathcal{L}\big(\widetilde{\boldsymbol{Y}} \,|\, \boldsymbol{Y} = \boldsymbol{y}\big)\,. \tag{2.22}$$

This is achieved in two steps, detailed in Section 2.4 and Section 2.5.

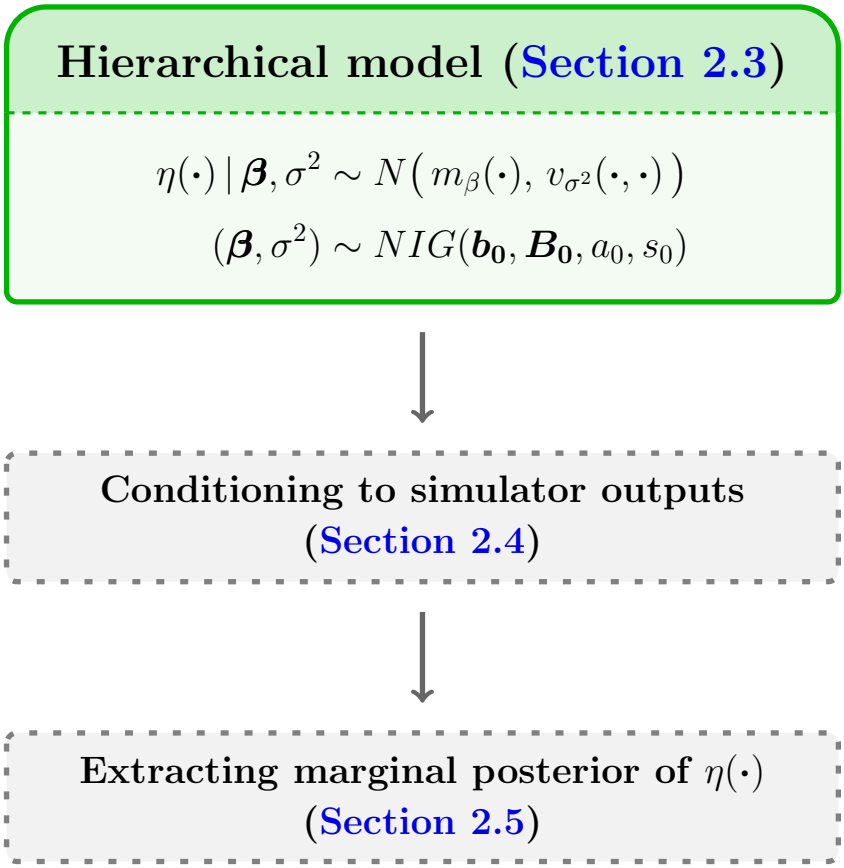


Figure 2.1: Schematic representation of the steps behind the construction of a GP emulator. In the green box, what Section 2.3 has just dealt with.

2.4. Conditioning the Model to Observations

This section is concerned with conditioning the prior model (2.17) to the observed simulator outputs $\boldsymbol{y} \in \mathbb{R}^n$. In terms of the notation introduced in Subsection 2.3.3, we aim to find $\mathcal{L}\big(\widetilde{\boldsymbol{Y}}, \boldsymbol{\beta}, \sigma^2 \,|\, \boldsymbol{Y} = \boldsymbol{y}\big)$, which we more simply write as follows:

$$\mathcal{L}\big(\widetilde{\boldsymbol{Y}}, \boldsymbol{\beta}, \sigma^2 | \, \boldsymbol{y}\big)\,. \tag{2.23}$$

We compute the density of this distribution in two steps, by deriving the densities associated with $\mathcal{L}(\widetilde{\boldsymbol{Y}} \,|\, \boldsymbol{\beta}, \sigma^2, \boldsymbol{y})$ in Subsection 2.4.1, and with $\mathcal{L}(\boldsymbol{\beta}, \sigma^2 \,|\, \boldsymbol{y})$ in Subsection 2.4.2. Their product returns the density associated with (2.23).

2.4.1. Conditioning a Gaussian Vector

Equation (2.17.a) specifies the conditional distribution of the process $\eta(\cdot)$ given $\boldsymbol{\beta}, \sigma^2$:

$$\eta(\cdot) \,|\, \boldsymbol{\beta}, \sigma^2 \sim \mathcal{GP}\big(m_\beta(\cdot), \sigma^2 c(\cdot,\cdot)\big)\,. \tag{2.24}$$

In terms of the notation introduced in (2.20), and given the form of $m_\beta(\cdot)$ in (2.4), we can therefore write the following, for any choice of $\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} \in \mathcal{P}$:

$$\widetilde{\boldsymbol{Y}} \,|\, \boldsymbol{\beta}, \sigma^2 \sim N\big(\widetilde{\boldsymbol{H}}\boldsymbol{\beta},\, \sigma^2 \widetilde{\boldsymbol{A}}\big)\,, \tag{2.25}$$

where

$$\widetilde{\boldsymbol{H}} = \begin{pmatrix} \boldsymbol{h}(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}})^T \\ \vdots \\ \boldsymbol{h}(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}})^T \end{pmatrix} \in \mathbb{R}^{k\times q}\,, \quad \widetilde{\boldsymbol{A}} = \begin{pmatrix} c(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{1}}) & \cdots & c(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}) \\ \vdots & \ddots & \vdots \\ c(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{1}}) & \cdots & c(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}) \end{pmatrix} \in \mathbb{R}^{k\times k}\,. \tag{2.26}$$

We now have to further condition (2.25) on the observation $\boldsymbol{Y} = \boldsymbol{y}$. Since $\eta(\cdot)$ is a GP, the vector $(\widetilde{\boldsymbol{Y}}, \boldsymbol{Y})$ is jointly Gaussian distributed, hence the conditioning becomes a simple application of a well-known result about conditioning of Gaussian random vectors. For completeness, Lemma A.2 in Appendix reports the result.

Before stating here the result of interest for us (Proposition 2.4.1), let us introduce the following notation, of particular importance throughout the whole chapter. We denote by $\boldsymbol{H}$ and $\boldsymbol{A}$ the following matrices:

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{h}(\boldsymbol{x}_{\boldsymbol{1}})^T \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x}_{\boldsymbol{n}})^T \end{pmatrix} \in \mathbb{R}^{n\times q}\,, \quad \boldsymbol{A} = \begin{pmatrix} c(\boldsymbol{x}_{\boldsymbol{1}}, \boldsymbol{x}_{\boldsymbol{1}}) & \ldots & c(\boldsymbol{x}_{\boldsymbol{1}}, \boldsymbol{x}_{\boldsymbol{n}}) \\ \vdots & \ddots & \vdots \\ c(\boldsymbol{x}_{\boldsymbol{n}}, \boldsymbol{x}_{\boldsymbol{1}}) & \ldots & c(\boldsymbol{x}_{\boldsymbol{n}}, \boldsymbol{x}_{\boldsymbol{n}}) \end{pmatrix} \in \mathbb{R}^{n\times n}\,. \tag{2.27}$$

Moreover, we denote by $\boldsymbol{t}(\boldsymbol{x}) \in \mathbb{R}^n$ the vector of prior correlations between any point $\boldsymbol{x}$ of the input space and the n design points:

$$\boldsymbol{t}(\boldsymbol{x}) = \big(c(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{1}}), \ldots, c(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{n}})\big)^T \in \mathbb{R}^n\,, \quad \boldsymbol{x} \in \mathcal{P}\,. \tag{2.28}$$

Proposition 2.4.1. *Let $\boldsymbol{\beta} \in \mathbb{R}^q$, $\sigma^2 > 0$, and $\boldsymbol{y} \in \mathbb{R}^n$ be fixed. Suppose that the distribution of the process $\eta(\cdot)$, given $\boldsymbol{\beta}$ and σ^2, be as in (2.24). Then, the process $\eta(\cdot)$ further conditioned on the event $\big(\eta(\boldsymbol{x_1}), \dots, \eta(\boldsymbol{x_n})\big)^T = \boldsymbol{y}$ is still a Gaussian process. Its distribution is as follows:*

$$\eta(\cdot) \,|\, \boldsymbol{\beta}, \sigma^2, \boldsymbol{y} \;\sim\; GP\big(m^*_\beta(\cdot),\, v^*_{\sigma^2}(\cdot,\cdot)\big)\,, \tag{2.29}$$

where, for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}$, $v^*_{\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 c(\boldsymbol{x}, \boldsymbol{x}')$,

$$m^*_\beta(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{\beta} + \boldsymbol{t}(\boldsymbol{x})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta}), \tag{2.30.a}$$

$$c^*(\boldsymbol{x}, \boldsymbol{x}') = c(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{t}(\boldsymbol{x})^T\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}'). \tag{2.30.b}$$

The expressions of $\boldsymbol{H} \in \mathbb{R}^{n\times q}$, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{t}(\boldsymbol{x}) \in \mathbb{R}^n$ are provided in (2.27), (2.28).

Proof. Let us consider any k points $\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \dots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} \in \mathcal{P}$. We show that the finite-dimensional distribution of $\eta(\cdot)$ at these points, conditioned on $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{y}$, is multivariate Gaussian with the correct mean vector and variance matrix. To the aim, we use the notation introduced in (2.19) and (2.20) and apply Lemma A.2.

Given the fixed values of $\boldsymbol{\beta}$ and σ^2, the vector $(\widetilde{\boldsymbol{Y}}, \boldsymbol{Y}) \in \mathbb{R}^{k+n}$ is jointly Gaussian by assumption (2.24). Hence, by the first part of Lemma A.2, the distribution of $\widetilde{\boldsymbol{Y}}$ given $\boldsymbol{Y} = \boldsymbol{y}$ is as well Gaussian. To find the conditioned mean and variance of $\widetilde{\boldsymbol{Y}}$, we apply formulas (A.4). The mean and variance of $\boldsymbol{Y}$ and $\widetilde{\boldsymbol{Y}}$ are respectively as follows:

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{Y}} &= \boldsymbol{H}\boldsymbol{\beta}, \quad & \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Y}} &= \sigma^2\boldsymbol{A}, \\ \boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}} &= \widetilde{\boldsymbol{H}}\boldsymbol{\beta}, \quad & \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{Y}}} &= \sigma^2\widetilde{\boldsymbol{A}}, \end{aligned}$$

with $\widetilde{\boldsymbol{H}}$ and $\widetilde{\boldsymbol{A}}$ as defined in (2.26). Moreover, by (2.24), the covariance matrix $\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\boldsymbol{Y}}$ between $\widetilde{\boldsymbol{Y}}$ and $\boldsymbol{Y}$ is $\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\boldsymbol{Y}} = \sigma^2\widetilde{\boldsymbol{T}}$, where

$$\widetilde{\boldsymbol{T}} = \begin{pmatrix} c(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \boldsymbol{x_1}) & \cdots & c(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \boldsymbol{x_n}) \\ \vdots & \ddots & \vdots \\ c(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}, \boldsymbol{x_1}) & \cdots & c(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}, \boldsymbol{x_n}) \end{pmatrix} \in \mathbb{R}^{k\times n}\,.$$

Hence, applying formula (A.4.a) in Appendix, we find:

$$\begin{aligned}\boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}}^{\text{cond}} &= \boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}} + \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\boldsymbol{Y}}\boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Y}}{}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_{\boldsymbol{Y}}) \\ &= \widetilde{\boldsymbol{H}}\boldsymbol{\beta} + (\sigma^2\widetilde{\boldsymbol{T}})(\sigma^2\boldsymbol{A})^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta}) \\ &= \widetilde{\boldsymbol{H}}\boldsymbol{\beta} + \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})\,. \end{aligned} \tag{2.31}$$

This is an equality in $\mathbb{R}^k$. If we consider the j^{th} component of both sides, and simply call $\boldsymbol{x}$ the generic point $\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}$, then formula (2.30.a) is recovered. As for the variance of $\widetilde{\boldsymbol{Y}}$ given $\boldsymbol{Y}=\boldsymbol{y}$, equation (A.4.b) of Lemma A.2 gives us:

$$\begin{aligned}\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{Y}}}^{\text{cond}} &= \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{Y}}} - \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\boldsymbol{Y}}\,\boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Y}}{}^{-1}\,\boldsymbol{\Sigma}_{\boldsymbol{Y}\widetilde{\boldsymbol{Y}}} \\ &= \sigma^2\widetilde{\boldsymbol{A}} - (\sigma^2\widetilde{\boldsymbol{T}})(\sigma^2\boldsymbol{A})^{-1}(\sigma^2\widetilde{\boldsymbol{T}})^T \\ &= \sigma^2\Big(\widetilde{\boldsymbol{A}} \;-\; \widetilde{\boldsymbol{T}}\,\boldsymbol{A}^{-1}\,\widetilde{\boldsymbol{T}}^T\Big)\,. \end{aligned} \tag{2.32}$$

This is an equality in $\mathbb{R}^{k\times k}$. By considering the element (j_1, j_2) of both sides of last equation, and renaming $\widetilde{\boldsymbol{x}}_{\boldsymbol{j_1}}$ as $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}_{\boldsymbol{j_2}}$ as $\boldsymbol{x}'$, we immediately get formula (2.30.b). This completes the proof. □

Proposition 2.4.1 provides the law of $\eta(\cdot)$ conditioned on $\boldsymbol{\beta}$, σ^2 and the observations $\boldsymbol{y}$, that is $\mathcal{L}(\widetilde{\boldsymbol{Y}}\,|\,\boldsymbol{\beta},\sigma^2,\boldsymbol{y})$. In the next subsection, we derive $\mathcal{L}(\boldsymbol{\beta},\sigma^2\,|\,\boldsymbol{y})$.

2.4.2. Bayesian Conjugate Analysis on Hyperparameters

In equation (2.17.b) we have modelled the marginal prior distribution of $(\boldsymbol{\beta},\sigma^2)$ as Normal-Inverse-Gamma. In this subsection we condition this to the n observations $\boldsymbol{Y}=\boldsymbol{y}$, and derive the posterior of the pair $(\boldsymbol{\beta},\sigma^2)$. Proposition 2.4.2 shows the result.

The proof is an application of Bayes' rule, equation (1.10). None of the steps is therefore conceptually involved, although the algebra often requires some care. We provide below the details of all the steps in a hopefully clear and easy-to-follow way.

Proposition 2.4.2. *Consider model (2.17). In particular, assume:*

$$(\boldsymbol{\beta},\sigma^2) \sim NIG(\boldsymbol{b_0},\boldsymbol{B_0},a_0,s_0)\,, \tag{2.33}$$

for some $\boldsymbol{b_0} \in \mathbb{R}^q$, $\boldsymbol{B_0} \in \mathbb{R}^{q\times q}$ *symmetric positive semi-definite,* $a_0, s_0 > 0$. *Then, for any* $\boldsymbol{y} \in \mathbb{R}^n$, *the distribution of* $(\boldsymbol{\beta}, \sigma^2)$ *conditioned on* $\boldsymbol{Y} = \boldsymbol{y}$ *is still NIG. Specifically, we have:*

$$(\boldsymbol{\beta}, \sigma^2) \,|\, \boldsymbol{y} \;\sim\; NIG(\boldsymbol{b}, \boldsymbol{B}, a, s)\,, \tag{2.34}$$

where:

$$\boldsymbol{B} = \boldsymbol{B_0} + \boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H} \;\in \mathbb{R}^{q\times q}\,, \tag{2.35.a}$$

$$\boldsymbol{b} = \boldsymbol{B}^{-1}\left(\boldsymbol{B_0}\boldsymbol{b_0} + \boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y}\right) \;\in \mathbb{R}^q, \tag{2.35.b}$$

$$a = a_0 + \frac{n}{2}\,, \tag{2.35.c}$$

$$s = s_0 + \frac{(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b_0})^T\boldsymbol{F}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b_0})}{2}\,, \tag{2.35.d}$$

and $\boldsymbol{F} = \boldsymbol{A} + \boldsymbol{H}\boldsymbol{B_0}^{-1}\boldsymbol{H}^T \in \mathbb{R}^{n\times n}$. *The matrices* $\boldsymbol{H}$ *and* $\boldsymbol{A}$ *are defined in (2.27).*

Proof. As in (2.19), set $\boldsymbol{Y} = \big(\eta(\boldsymbol{x_1}), \ldots, \eta(\boldsymbol{x_n})\big)^T$. From equations (2.17), the likelihood of $\boldsymbol{Y} = \boldsymbol{y}$ given $\boldsymbol{\beta}$ and σ^2 is Normal, and the prior of the pair $(\boldsymbol{\beta}, \sigma^2)$ is NIG. Recalling from (2.16) the form of the NIG density, we can write the following:

$$p_{_{Y|\beta,\sigma^2}}(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) \propto \frac{1}{(\sigma^2)^{n/2}}\; \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta})\right]$$

$$p_{_{\beta,\sigma^2}}(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{(\sigma^2)^{a_0+1+q/2}}\; \exp\left[-\frac{1}{2\sigma^2}\Big(2s_0 + (\boldsymbol{\beta} - \boldsymbol{b_0})^T\boldsymbol{B_0}(\boldsymbol{\beta} - \boldsymbol{b_0})\Big)\right]$$

All factors not involving $\boldsymbol{y}, \boldsymbol{\beta}, \sigma^2$ have been ignored on the RHS of the previous equations. Through Bayes's rule (equation (1.10)) we obtain the following expression for the posterior of $(\boldsymbol{\beta}, \sigma^2)$:

$$p_{_{\beta,\sigma^2|Y}}(\boldsymbol{\beta}, \sigma^2 \,|\, \boldsymbol{y}) \;\propto\; p_{_{Y|\beta,\sigma^2}}(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2) \times p_{_{\beta,\sigma^2}}(\boldsymbol{\beta}, \sigma^2)$$

$$\propto \frac{\exp\left[-\frac{1}{2\sigma^2}\Big(2s_0 + (\boldsymbol{\beta} - \boldsymbol{b_0})^T\boldsymbol{B_0}(\boldsymbol{\beta} - \boldsymbol{b_0}) + (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta})\Big)\right]}{(\sigma^2)^{a_0+\frac{n}{2}+1+\frac{q}{2}}}\,. \tag{2.36}$$

We now need to rearrange the argument of the exponential, to see that this is indeed the density of a NIG distribution for the pair $(\boldsymbol{\beta}, \sigma^2)$. This will prove the statement.

Step 1: Write the argument of the exponential as a quadratic form in $\boldsymbol{\beta}$, plus remainder. To this aim, we have:

$$\begin{aligned}
&(\boldsymbol{\beta}-\boldsymbol{b_0})^T\boldsymbol{B_0}(\boldsymbol{\beta}-\boldsymbol{b_0}) + (\boldsymbol{y}-\boldsymbol{H\beta})^T\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H\beta}) \\
&\quad = \boldsymbol{\beta}^T(\boldsymbol{B_0}+\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H})\boldsymbol{\beta} - 2\boldsymbol{\beta}^T(\boldsymbol{B_0 b_0}+\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y}) + \boldsymbol{b_0}^T\boldsymbol{B b_0} + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} \\
&\quad = \boldsymbol{\beta}^T\boldsymbol{B}\,\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\boldsymbol{B b} + \boldsymbol{b_0}^T\boldsymbol{B_0 b_0} + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} \\
&\quad = (\boldsymbol{\beta}-\boldsymbol{b})^T\boldsymbol{B}\,(\boldsymbol{\beta}-\boldsymbol{b}) - \boldsymbol{b}^T\boldsymbol{B b} + \boldsymbol{b_0}^T\boldsymbol{B_0 b_0} + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y}\,. \qquad (2.37)
\end{aligned}$$

The second equality immediately follows from the definitions of $\boldsymbol{B}$ and $\boldsymbol{b}$ in (2.35); the last one is a simple completion of the square (quadratic form in $\boldsymbol{\beta}-\boldsymbol{b}$). Let us now expand the term $\boldsymbol{b}^T\boldsymbol{B b}$ in (2.37). From the definition of $\boldsymbol{B}$ and $\boldsymbol{b}$, and the symmetry of $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{B_0}$, we obtain the following:

$$\begin{aligned}
\boldsymbol{b}^T\boldsymbol{B b} &= \left(\boldsymbol{B_0 b_0}+\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y}\right)^T\boldsymbol{B}^{-1}\left(\boldsymbol{B_0 b_0}+\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y}\right) \\
&= \boldsymbol{b_0}^T\boldsymbol{B_0}\boldsymbol{B}^{-1}\boldsymbol{B_0 b_0} + 2\boldsymbol{b_0}^T\boldsymbol{B_0}\boldsymbol{B}^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{B}^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y}. \qquad (2.38)
\end{aligned}$$

Step 2: Simplify the three addends in (2.38), to get new expression for $\boldsymbol{b}^T\boldsymbol{B b}$.
To accomplish the aim, we use a linear algebra lemma, Lemma B.1 in Appendix, to explicitly invert $\boldsymbol{B} = \boldsymbol{B_0}+\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}$. Formula (B.1) applied to such $\boldsymbol{B}$ yields:

$$\boldsymbol{B}^{-1} = \boldsymbol{B_0}^{-1} - \boldsymbol{B_0}^{-1}\boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{H}\boldsymbol{B_0}^{-1}, \qquad (2.39)$$

where $\boldsymbol{F} = \boldsymbol{A}+\boldsymbol{H}\boldsymbol{B_0}^{-1}\boldsymbol{H}^T$, as defined in the statement. From this, we deduce:

$$\boldsymbol{B_0}\boldsymbol{B}^{-1}\boldsymbol{B_0} = \boldsymbol{B_0} - \boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{H}\,; \qquad (2.40)$$

$$\begin{aligned}
\boldsymbol{B_0}\boldsymbol{B}^{-1}\boldsymbol{H}^T &= \boldsymbol{H}^T - \boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{H}\boldsymbol{B_0}^{-1}\boldsymbol{H}^T \\
&\overset{(\text{def. of } \boldsymbol{F})}{=} \boldsymbol{H}^T - \boldsymbol{H}^T\boldsymbol{F}^{-1}(\boldsymbol{F}-\boldsymbol{A}) \\
&= \boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{A}\,; \qquad (2.41)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{A}^{-1}\boldsymbol{H}\left(\boldsymbol{B}^{-1}\boldsymbol{H}^T\right)\boldsymbol{A}^{-1} &\overset{(2.41)}{=} \boldsymbol{A}^{-1}\boldsymbol{H}\left(\boldsymbol{B_0}^{-1}\boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{A}\right)\boldsymbol{A}^{-1} \\
&= \boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{B_0}^{-1}\boldsymbol{H}^T\boldsymbol{F}^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{F}-\boldsymbol{A})\boldsymbol{F}^{-1} \\
&= \boldsymbol{A}^{-1} - \boldsymbol{F}^{-1}\,. \qquad (2.42)
\end{aligned}$$

Given the above identities, the three terms of (2.38) become as follows:

1. $\boldsymbol{b_0}^T\left(\boldsymbol{B_0}\boldsymbol{B}^{-1}\boldsymbol{B_0}\right)\boldsymbol{b_0} \overset{(2.40)}{=} \boldsymbol{b_0}^T\boldsymbol{B_0}\boldsymbol{b_0} - \boldsymbol{b_0}^T\boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{H}\boldsymbol{b_0}$
2. $2\boldsymbol{b_0}^T\left(\boldsymbol{B_0}\boldsymbol{B}^{-1}\boldsymbol{H}^T\right)\boldsymbol{A}^{-1}\boldsymbol{y} \overset{(2.41)}{=} 2\boldsymbol{b_0}^T\left(\boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{A}\right)\boldsymbol{A}^{-1}y = 2\boldsymbol{b_0}^T\boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{y}$
3. $\boldsymbol{y}^T\left(\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{B}^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\right)\boldsymbol{y} \overset{(2.42)}{=} \boldsymbol{y}^T\left(\boldsymbol{A}^{-1}-\boldsymbol{F}^{-1}\right)\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{F}^{-1}\boldsymbol{y}.$

Step 3: Rewrite the expression for $\boldsymbol{b}^T\boldsymbol{B}\boldsymbol{b}$ and plug this back in (2.37).
Recalling (2.38), by simply adding the three terms above we obtain:

$$\begin{aligned}\boldsymbol{b}^t\boldsymbol{B}\boldsymbol{b} &= \boldsymbol{b_0}^T\boldsymbol{B_0}\boldsymbol{b_0} + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} - \boldsymbol{b_0}^T\boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{H}\boldsymbol{b_0} + 2\boldsymbol{b_0}^T\boldsymbol{H}^T\boldsymbol{F}^{-1}\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{F}^{-1}\boldsymbol{y} \\ &= \boldsymbol{b_0}^T\boldsymbol{B_0}\boldsymbol{b_0} + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} - (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})^T\boldsymbol{F}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0}),\end{aligned}$$

Equivalently (just rearrange the order):

$$-\boldsymbol{b}^T\boldsymbol{B}\boldsymbol{b} + \boldsymbol{b_0}^T\boldsymbol{B_0}\boldsymbol{b_0} + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} = (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})^T\boldsymbol{F}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})\,. \tag{2.43}$$

Substituting (2.43) back into (2.37) immediately yields the following:

$$\begin{aligned}&(\boldsymbol{\beta}-\boldsymbol{b_0})^T\boldsymbol{B_0}(\boldsymbol{\beta}-\boldsymbol{b_0}) + (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})^T\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta}) \\ &\qquad = (\boldsymbol{\beta}-\boldsymbol{b})^T\boldsymbol{B}\,(\boldsymbol{\beta}-\boldsymbol{b}) + (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})^T\boldsymbol{F}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})\,.\end{aligned} \tag{2.44}$$

Step 4: Plug (2.44) back into the posterior density (2.36). This yields:

$$p(\boldsymbol{\beta},\sigma^2|\boldsymbol{y}) \propto \frac{\exp\left[-\frac{1}{2\sigma^2}\left(2s_0 + (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})^T\boldsymbol{F}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0}) + (\boldsymbol{\beta}-\boldsymbol{b})^T\boldsymbol{B}\,(\boldsymbol{\beta}-\boldsymbol{b})\right)\right]}{(\sigma^2)^{\left(a_0+\frac{n}{2}\right)+1+\frac{q}{2}}}\,.$$

Given the definitions of a and s in (2.35), we can rewrite this as

$$p(\boldsymbol{\beta},\sigma^2|\boldsymbol{y}) \propto \frac{\exp\left[-\frac{1}{2\sigma^2}\left(2s + (\boldsymbol{\beta}-\boldsymbol{b})^T\boldsymbol{B}\,(\boldsymbol{\beta}-\boldsymbol{b})\right)\right]}{(\sigma^2)^{a+1+\frac{q}{2}}}\,. \tag{2.45}$$

Comparing (2.45) with (2.16), we see that this is indeed the density of a $NIG(\boldsymbol{b},\boldsymbol{B},a,s)$ distribution. Hence, the proof is complete. □

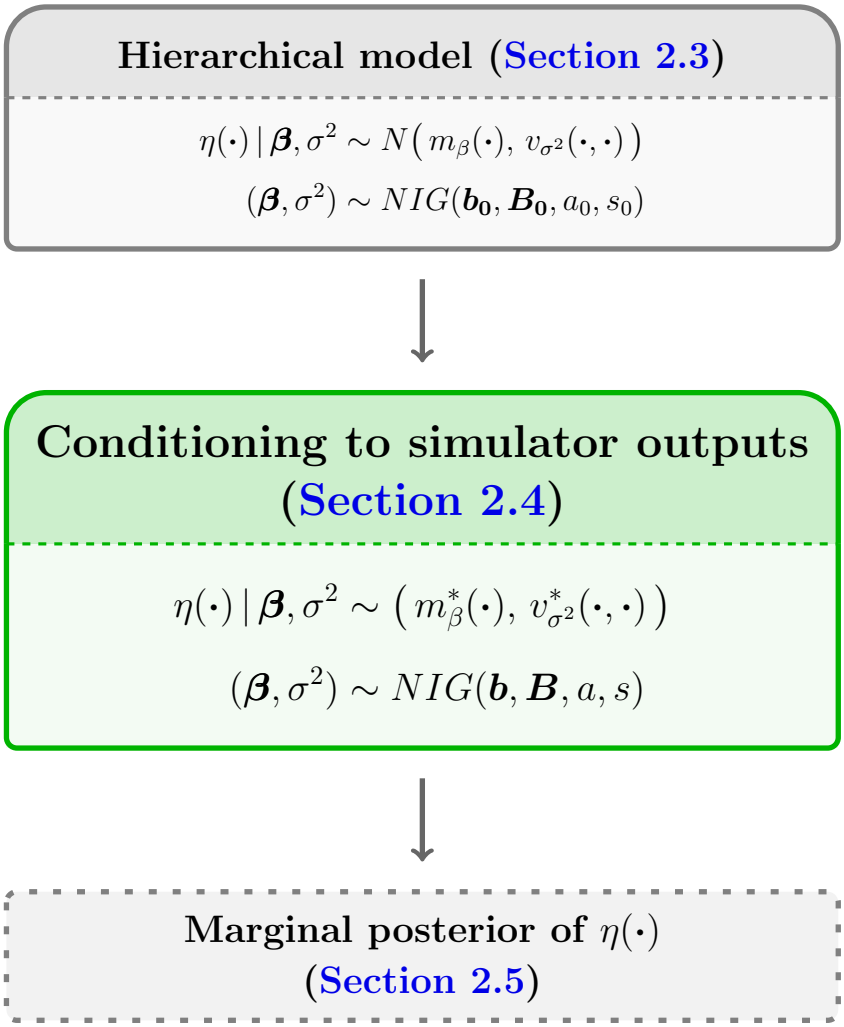


Figure 2.2: Continued from Figure 2.1, schematic representation of the steps behind the construction of a GP emulator. The middle green box highlights the results of Section 2.4, specifically associated with Propositions 2.4.1 and 2.4.2.

Proposition 2.4.1 and Proposition 2.4.2 together provide the distribution of the triple $\big(\eta(\cdot),\boldsymbol{\beta},\sigma^2\big)$, conditioned on having observed the simulator outputs $y_1,\ldots,y_n$:

$$\eta(\cdot)\,|\,\boldsymbol{\beta},\sigma^2,\boldsymbol{y} \sim \mathcal{GP}\big(m^*_\beta(\cdot),\, \sigma^2 c^*(\cdot,\cdot)\big)\,, \tag{2.46.a}$$

$$(\boldsymbol{\beta},\sigma^2)\,|\,\boldsymbol{y} \sim NIG(\boldsymbol{b},\boldsymbol{B},a,s)\,. \tag{2.46.b}$$

In Section 2.5, we uncondition expression (2.46.a) from $\boldsymbol{\beta}$ and σ^2, essentially by integrating over their posterior distribution (2.46.b). This provides the law of the final emulator, $\mathcal{L}\big(\eta(\cdot)\,|\,\boldsymbol{Y}=\boldsymbol{y}\big)$.

2.5. Marginal Posterior Distribution of the Model

In order to work out the distribution of $\eta(\cdot)$ conditioned on $\boldsymbol{Y} = \boldsymbol{y}$ only, two parallel paths can be followed. One is to integrate out $\boldsymbol{\beta}$ and σ^2 from the conditional density of $(\eta(\cdot), \boldsymbol{\beta}, \sigma^2)$ given $\boldsymbol{y}$. Using the notation introduced in (2.20), we can write:

$$\begin{aligned} p_{\widetilde{Y}|Y}(\widetilde{\boldsymbol{y}}|\boldsymbol{y}) &= \int p_{\widetilde{Y},\beta,\sigma^2|Y}(\widetilde{\boldsymbol{y}}, \boldsymbol{\beta}, \sigma^2 \,|\, \boldsymbol{y})\; d\boldsymbol{\beta}\, d\sigma^2 \\ &= \int p_{\widetilde{Y}|\beta,\sigma^2,Y}(\widetilde{\boldsymbol{y}} \,|\, \boldsymbol{\beta}, \sigma^2, \boldsymbol{y}) \times p_{\beta,\sigma^2\,|\,Y}(\boldsymbol{\beta}, \sigma^2 \,|\, \boldsymbol{y})\; d\boldsymbol{\beta}\, d\sigma^2\,. \end{aligned} \tag{2.47}$$

The distributions in the last line are provided in (2.46.a), (2.46.b). Alternatively, we can write down the expression of a random variable whose distribution is the one of $\widetilde{\boldsymbol{Y}}$ given the values of $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{y}$, and replace the fixed constants $\boldsymbol{\beta}$ and σ^2 by random variables which follow the distribution (2.46.b), recognising the distribution that the random variable obtained this way follows.

Departing from the standard literature, we follow the second approach here and provide the result in Theorem 2.5.5. The approach needs however some technical results to be rigorously justified, which the author of this work has developed and proved. Moreover, we need to introduce ad-hoc definitions, among which the key one of Student-t process. This is done in Subsection 2.5.1.

2.5.1. Some Definitions and Technical Results

Definition 2.5.1 (Student-t random vector)**.** Let $\nu > 0$, $\boldsymbol{\mu} \in \mathbb{R}^k$, and let $\boldsymbol{\Sigma} \in \mathbb{R}^{k\times k}$ be a symmetric, positive semi-definite matrix. A random vector $\boldsymbol{Y} \in \mathbb{R}^k$ is distributed according to a Student-t distribution with ν degrees of freedom, mean $\boldsymbol{\mu}$, and kernel matrix $\boldsymbol{\Sigma}$, if it can be written as follows:

$$\boldsymbol{Y} = \boldsymbol{\mu} + \sqrt{\frac{\nu}{V}}\, \boldsymbol{X} \;\in \mathbb{R}^k, \tag{2.48}$$

where $\boldsymbol{X} \sim N(\boldsymbol{0_k}, \boldsymbol{\Sigma})$, $V \sim \chi^2(\nu)$, and $\boldsymbol{X}$ and V are independent of each other. We write $\boldsymbol{Y} \sim t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Notice that the one-dimensional case with $\mu = 0$ and $\Sigma = I_1 = 1$ recovers the classical t-distribution. Moreover, not difficult calculations show the following:

$$\boldsymbol{Y} \sim t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \Longrightarrow \quad \begin{cases} \mathbb{E}\left[\boldsymbol{Y}\right] = \boldsymbol{\mu}\,, & \text{(2.49.a)} \\ \mathrm{Var}\left[\boldsymbol{Y}\right] = \dfrac{\nu}{\nu - 2}\,\boldsymbol{\Sigma}\,, \quad \text{if } \nu > 2\,. & \text{(2.49.b)} \end{cases}$$

Once the Student-t distribution has been defined in the multivariate case, it is immediate to extend the concept to random processes, in exactly the same way in which a GP represents the infinite-dimensional analogue of Gaussian vectors.

Definition 2.5.2 (Student-t process)**.** A real-valued stochastic process η with input space $\mathcal{P}$ is a Student-t process (or t-process) with $\nu > 0$ degrees of freedom, mean function $m\colon \mathcal{P} \to \mathbb{R}$, and kernel function $S\colon \mathcal{P} \times \mathcal{P} \to \mathbb{R}$, if for any $\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}} \in \mathcal{P}$ the following holds:

$$\big(\eta(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}), \ldots, \eta(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}})\big)^T \sim t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.50}$$

where

$$\boldsymbol{\mu} = \begin{pmatrix} m(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}) \\ \vdots \\ m(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}) \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} S(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{1}}) & \ldots & S(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}) \\ \vdots & \ddots & \vdots \\ S(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{1}}) & \ldots & S(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}) \end{pmatrix}. \tag{2.51}$$

Let us now provide a formal definition of the concepts of location and scale parameters for a family of distributions. These will be useful to state Lemma 2.5.4. Intuitively, they are used to parameterise distributions sharing the same density "shape": the location parameter provides a measure of the displacement of the distribution; the scale parameter provides a measure of how spread this is.

Definition 2.5.3. Let $\big(\mathbb{P}_{\mu,\sigma}\big)$ be a family of probability distributions over $\big(\mathbb{R}^k, \mathcal{B}(\mathbb{R}^k)\big)$, indexed by $\boldsymbol{\mu} \in \mathbb{R}^k$, $\sigma > 0$. Denote by $F_{\mu,\sigma}\colon \mathbb{R}^k \to [0,1]$ the joint cumulative distribution function of $\mathbb{P}_{\mu,\sigma}$:

$$F_{\mu,\sigma}\big(\,(a_1, \ldots, a_k)\,\big) = \mathbb{P}_{\mu,\sigma}\big(X_1 \leq a_1, \ldots, X_k \leq a_k\big) \quad \forall\, \boldsymbol{a} \in \mathbb{R}^k\,, \tag{2.52}$$

where $(X_1, \ldots, X_k) \sim \mathbb{P}_{\mu,\sigma}$. We call $\boldsymbol{\mu}$ a *location parameter* and σ a *scale parameter*

for the family, if there exists $F\colon \mathbb{R}^k \to [0,1]$ such that

$$F_{\mu,\sigma}(\boldsymbol{x}) = F\left(\frac{\boldsymbol{x}-\boldsymbol{\mu}}{\sigma}\right) \qquad \forall\, \boldsymbol{\mu}, \boldsymbol{x} \in \mathbb{R}^k,\ \sigma > 0\,. \tag{2.53}$$

Notice that, if such F exists, then $F = F_{0,1}$.

Example 2.1. If $\mu \in \mathbb{R}$, $\mathbb{P}_{\mu,\sigma} \sim N(\mu, \sigma^2)$ is an example of a family with location parameter μ and scale parameter σ. More broadly, $\boldsymbol{\mu} \in \mathbb{R}^k$ and $\sigma > 0$ are location and scale parameters for a family of multivariate normal distributions $\mathbb{P}_{\mu,\sigma} \sim N(\boldsymbol{\mu}, \sigma^2 \boldsymbol{A})$, where $\boldsymbol{A} \in \mathbb{R}^{k\times k}$ is a fixed, symmetric and positive semi-definite matrix.

Example 2.2. Consider $\mathbb{P}_s \sim IG(a,s)$ for any fixed $a > 0$. The parameter s is a scale parameter for the family, see property (2.11). There is no location parameter as such. However, one may easily consider a larger family $(\mathbb{P}_{\mu,s})$, with $\mathbb{P}_{\mu,s}$ being the distribution of $\mu + X$, $X \sim IG(a,s)$. For this family, μ and s represent a location and a scale parameter, respectively.

Remark 2.3. With the notation of Definition 2.5.3, if we call ν the probability measure associated with $F_{0,1} = F$, we have

$$\mathbb{P}_{\mu,\sigma}(\boldsymbol{\mu} + \sigma \boldsymbol{A}) = \nu(\boldsymbol{A}) \quad \forall\, \boldsymbol{A} \in \mathcal{B}(\mathbb{R}^k)\,.^{2} \tag{2.54}$$

Albeit intuitive, we provide a proof of the claim. Since the sets of the form $\boldsymbol{B} = \prod_{j=1}^k (-\infty, b_j]$ generate the Borel σ-algebra on $\mathbb{R}^k$, we need to check (2.54) on these sets only. For such a $\boldsymbol{B}$, call $\boldsymbol{b} = (b_1, \dots, b_k)$. Then we have:

$$\begin{aligned}
\mathbb{P}_{\mu,\sigma}(\boldsymbol{\mu} + \sigma \boldsymbol{B}) &= \mathbb{P}_{\boldsymbol{\mu},\sigma}\left(\prod_{j=1}^{k} (-\infty, \mu_j + \sigma b_j]\right) \\
&\overset{(2.52)}{=} F_{\mu,\sigma}\big(\,(\mu_1 + \sigma b_1,\ \dots\ , \mu_k + \sigma b_k)\,\big) \\
&= F_{\mu,\sigma}\left(\boldsymbol{\mu} + \sigma \boldsymbol{b}\right) \\
&\overset{(2.53)}{=} F(\boldsymbol{b}) = \nu(\boldsymbol{B})\,.
\end{aligned}$$

[2] As per standard notation, for $\boldsymbol{x} \in \mathbb{R}^k$ and $\boldsymbol{A} \subseteq \mathbb{R}^k$, we define $\boldsymbol{x} + \boldsymbol{A} := \{\boldsymbol{x} + \boldsymbol{a} \,|\, \boldsymbol{a} \in \boldsymbol{A}\}$.

The last equality follows from the fact that F is the cumulative distribution function of ν and by the definition of $\boldsymbol{B}$.

We can now state and prove the following slightly technical result. This will allow us to deal properly with independence of different random variables in the proof of Theorem 2.5.5, when we replace the posterior expressions $\boldsymbol{\beta}$ and σ^2 within the expression of the process $\eta(\cdot)$.

Lemma 2.5.4. *Let $\left(\mathbb{P}_{b,s}\right)$ be a family of distributions over $\mathbb{R}^k$ with location parameter $\boldsymbol{b} \in \mathbb{R}^k$ and scale parameter $s > 0$. Consider random $\boldsymbol{Y} \in \mathbb{R}^k$, $\boldsymbol{\beta} \in \mathbb{R}^k$ and $\sigma > 0$, and suppose that*

$$\boldsymbol{Y} \,|\, (\boldsymbol{\beta} = \boldsymbol{b}, \sigma = s) \sim \mathbb{P}_{\boldsymbol{f}(\boldsymbol{b}),\, g(s)} \qquad \forall\, \boldsymbol{b} \in \mathbb{R}^k,\ s > 0\,, \tag{2.55}$$

for some suitable functions $\boldsymbol{f}\colon \mathbb{R}^k \to \mathbb{R}^k$ and $g\colon \mathbb{R}^+ \to \mathbb{R}^+$. Then:

$$\frac{\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{\beta})}{g(\sigma)} \;\perp\!\!\!\perp\; (\boldsymbol{\beta}, \sigma)\,, \tag{2.56}$$

where the symbol $\perp\!\!\!\perp$ denotes independence of random variables, or vectors.

Proof. For fixed $\boldsymbol{b} \in \mathbb{R}^k$ and $s > 0$, consider the Borel-measurable, bijective map

$$\begin{aligned} \boldsymbol{\varphi}_{\boldsymbol{b},s}\colon\ \mathbb{R}^k &\longrightarrow \mathbb{R}^k \\ \boldsymbol{y} &\longmapsto \frac{\boldsymbol{y} - \boldsymbol{b}}{s}\,, \end{aligned} \tag{2.57}$$

and define the random vector (function of $\boldsymbol{Y}$)

$$\boldsymbol{Z}_{\boldsymbol{b},\boldsymbol{s}} := \boldsymbol{\varphi}_{\boldsymbol{b},\boldsymbol{s}}(\boldsymbol{Y})\,. \tag{2.58}$$

Ultimately, we would like to prove that the random vector $\boldsymbol{Z}_{\boldsymbol{f}(\boldsymbol{\beta}),\, \boldsymbol{g}(\boldsymbol{\sigma})}$ is independent of the pair $(\boldsymbol{\beta}, \sigma^2)$. To simplify the notation, let us call:

$$\widetilde{\boldsymbol{b}} := \boldsymbol{f}(\boldsymbol{b}) \in \mathbb{R}^k, \quad \widetilde{s} := g(s) > 0 \tag{2.59.a}$$

$$\widetilde{\boldsymbol{\beta}} := \boldsymbol{f}(\boldsymbol{\beta}) \in \mathbb{R}^k, \quad \widetilde{\sigma} := g(\sigma) > 0\,. \tag{2.59.b}$$

Notice that $\widetilde{\boldsymbol{b}}$ and $\widetilde{s}$ in (2.59.a) are a real vector and number, respectively; $\widetilde{\boldsymbol{\beta}}$ and $\widetilde{\sigma}$ in

(2.59.b) are instead random quantities. Let us now denote by ν the probability measure $\mathbb{P}_{0,1}$, so that equation (2.54) holds. Then, for $\boldsymbol{A} \in \mathcal{B}(\mathbb{R}^k)$, we have the following:

$$\begin{aligned}
&\mathbb{P}\left(\boldsymbol{Z}_{\widetilde{\boldsymbol{b}},\widetilde{s}} \in \boldsymbol{A} \mid (\boldsymbol{\beta},\sigma) = (\boldsymbol{b},s)\right) \\
&\quad = \mathbb{P}\left(\boldsymbol{Y} \in {\boldsymbol{\varphi}_{\widetilde{\boldsymbol{b}},\widetilde{s}}}^{-1}(\boldsymbol{A}) \mid (\boldsymbol{\beta},\sigma) = (\boldsymbol{b},s)\right) \\
&\quad \overset{(2.55)}{=} \mathbb{P}_{\widetilde{\boldsymbol{b}},\widetilde{s}}({\boldsymbol{\varphi}_{\widetilde{\boldsymbol{b}},\widetilde{s}}}^{-1}(\boldsymbol{A})) = \mathbb{P}_{\widetilde{\boldsymbol{b}},\widetilde{s}}(\widetilde{\boldsymbol{b}} + \widetilde{s}\boldsymbol{A}) \\
&\quad \overset{(2.54)}{=} \nu(\boldsymbol{A})\,.
\end{aligned} \tag{2.60}$$

Since the set $\boldsymbol{A} \in \mathcal{B}(\mathbb{R}^k)$ is arbitrary, and $\boldsymbol{b}$ and s are as well, we can reformulate (2.60) as follows:

$$\boldsymbol{Z}_{\widetilde{\boldsymbol{b}},\widetilde{s}} \,|\, (\boldsymbol{\beta} = \boldsymbol{b}, \sigma = s) \;\sim \nu \quad \forall\, \boldsymbol{b} \in \mathbb{R}^k,\; s > 0\,. \tag{2.61}$$

Moreover, clearly, once conditioned on the event $\{\boldsymbol{\beta} = \boldsymbol{b}, \sigma = s\}$, the distribution of $\boldsymbol{Z}_{\boldsymbol{f(\beta)},\boldsymbol{g(\sigma)}}$ becomes the same as the one of $\boldsymbol{Z}_{\boldsymbol{f(b)},\boldsymbol{g(s)}}$. Hence, from (2.61), it immediately follows that:

$$\boldsymbol{Z}_{\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\sigma}}} \,|\, (\boldsymbol{\beta} = \boldsymbol{b}, \sigma = s) \;\sim\; \nu \quad \forall\, \boldsymbol{b} \in \mathbb{R}^k,\; s > 0\,. \tag{2.62}$$

The proof is almost complete. Since the probability measure ν does not depend on $\boldsymbol{\beta}$ and σ, it is intuitive from (2.62) that $\boldsymbol{Z}_{\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\sigma}}}$ is independent of the pair $(\boldsymbol{\beta},\sigma)$. We show this rigorously.

For simplicity, let us denote by λ the joint distribution of the pair $(\boldsymbol{\beta},\sigma)$ on $\mathbb{R}^k \times \mathbb{R}^+$. Simply by definition of conditional distribution, for all $\boldsymbol{A_1} \in \mathcal{B}(\mathbb{R}^k)$ and $\boldsymbol{A_2} \in \mathcal{B}(\mathbb{R}^k \times \mathbb{R}^+)$, we can write the following:

$$\begin{aligned}
\mathbb{P}\left[\boldsymbol{Z}_{\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\sigma}}} \in \boldsymbol{A_1},\, (\boldsymbol{\beta},\sigma) \in \boldsymbol{A_2}\right] &= \int_{A_2} \mathbb{P}\Big[\boldsymbol{Z}_{\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\sigma}}} \in \boldsymbol{A_1} \,|\, (\boldsymbol{\beta},\sigma) = (\boldsymbol{b},s)\Big]\, d\lambda(\boldsymbol{b},s) \\
&\overset{(2.62)}{=} \int_{A_2} \nu(\boldsymbol{A_1})\, d\lambda(\boldsymbol{b},s) \\
&= \nu(\boldsymbol{A_1})\, \lambda(\boldsymbol{A_2})\,.
\end{aligned} \tag{2.63}$$

Given the product form in which the LHS of (2.63) has factorised, together with the generality of $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$, we can conclude that $\boldsymbol{Z}_{\widetilde{\boldsymbol{\beta}},\widetilde{\sigma}}$ and $(\boldsymbol{\beta}, \sigma)$ are independent. This completes the proof. □

2.5.2. Distribution of the Emulator

We can now use the previous result to find out the marginal distribution of the process $\eta(\cdot)$ conditioned on $\boldsymbol{Y} = \boldsymbol{y}$ (notation introduced in Subsection 2.3.3). Before stating Theorem 2.5.5, let us briefly recall the setting.

We consider the random variable:

$$(\eta(\cdot), \boldsymbol{\beta}, \sigma^2)\colon\ (\Omega, \mathcal{F}, \mathbb{P}) \longrightarrow (\mathbb{R}^{\mathcal{P}} \times \mathbb{R}^q \times \mathbb{R}^+,\ \mathcal{B})\,. \tag{2.64}$$

Its conditional distribution given $\boldsymbol{Y} = \boldsymbol{y}$ is as follows (Propositions 2.4.1 and 2.4.2):

$$\eta(\cdot)\,|\,\boldsymbol{\beta}, \sigma^2, \boldsymbol{y} \sim \mathcal{GP}\big(m^*_\beta(\cdot),\ \sigma^2 c^*(\cdot,\cdot)\big)\,, \tag{2.65.a}$$

$$(\boldsymbol{\beta}, \sigma^2)\,|\,\boldsymbol{y} \sim NIG(\boldsymbol{b}, \boldsymbol{B}, a, s)\,. \tag{2.65.b}$$

The expressions of $m^*_\beta(\cdot)$ and $c^*(\cdot,\cdot)$ are as follows:

$$m^*_\beta(x) = \boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{\beta} + \boldsymbol{t}(\boldsymbol{x})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta})\,, \tag{2.66.a}$$

$$c^*(\boldsymbol{x}, \boldsymbol{x}') = c(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{t}(\boldsymbol{x})^T\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}')\,. \tag{2.66.b}$$

The expression of the hypeparamenters $\boldsymbol{b}, \boldsymbol{B}, a, s$ in terms of $\boldsymbol{b_0}, \boldsymbol{B_0}, a_0, s_0$ is given in Proposition 2.4.2.

Theorem 2.5.5. *Under the above notation, the process $\eta(\cdot)$ conditioned on $\boldsymbol{Y} = \boldsymbol{y}$ is a Student-t process with $2a$ degrees of freedom, mean function $m\colon \mathcal{P} \to \mathbb{R}$ given by*

$$m(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{b} + \boldsymbol{t}(\boldsymbol{x})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})\,, \tag{2.67}$$

and kernel function $S\colon \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ given by

$$S(\boldsymbol{x}, \boldsymbol{x}') = \frac{s}{a}\Big[c^*(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{p}(\boldsymbol{x})^T\boldsymbol{B}^{-1}\boldsymbol{p}(\boldsymbol{x}')\Big]\,. \tag{2.68}$$

For $\boldsymbol{x} \in \mathcal{P}$, the definition of the vector $\boldsymbol{p}(\boldsymbol{x}) \in \mathbb{R}^q$ is as follows:

$$\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{t}(\boldsymbol{x}) \ \in \mathbb{R}^q. \tag{2.69}$$

Remark 2.4. By expanding $c^*(\cdot, \cdot)$ in the expression of the kernel function (2.68), and given property (2.49.b), we see that the covariance function of η given $\boldsymbol{Y}$ reads as follows:

$$v(\boldsymbol{x}, \boldsymbol{x}') = \frac{s}{a-1} \left[c(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{t}(x)^T \boldsymbol{A}^{-1} \boldsymbol{t}(\boldsymbol{x}') + \boldsymbol{p}(\boldsymbol{x})^T \boldsymbol{B}^{-1} \boldsymbol{p}(\boldsymbol{x}') \right]. \tag{2.70}$$

The quantity $s/(a-1)$ is the mean of an $IG(a, s)$ random variable, hence the posterior mean of σ^2.

Proof. As usual, let us denote by $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \in \mathcal{P}$ the design points and consider any k inputs $\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_k \in \mathcal{P}$. We will study the joint distribution, at the inputs $\widetilde{\boldsymbol{x}}_j$, of the conditioned process $\eta(\cdot)$. As in equation (2.26), we define:

$$\widetilde{\boldsymbol{H}} = \begin{pmatrix} \boldsymbol{h}(\widetilde{\boldsymbol{x}}_1)^T \\ \vdots \\ \boldsymbol{h}(\widetilde{\boldsymbol{x}}_k)^T \end{pmatrix} \in \mathbb{R}^{k \times q}, \qquad \widetilde{\boldsymbol{T}} = \begin{pmatrix} \boldsymbol{t}(\widetilde{\boldsymbol{x}}_1)^T \\ \vdots \\ \boldsymbol{t}(\widetilde{\boldsymbol{x}}_k)^T \end{pmatrix} \in \mathbb{R}^{k \times n},$$

and $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{k \times k}$ the matrix with elements $\widetilde{A}_{ij} = c(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{x}}_j)$. We use the notation introduced in Subsection 2.3.3. Hence, from (2.65.a), we have:

$$\widetilde{\boldsymbol{Y}} \,|\, \boldsymbol{\beta}, \sigma^2, \boldsymbol{y} \ \sim \ N(\boldsymbol{f}(\boldsymbol{\beta}), \sigma^2 \boldsymbol{\Sigma}), \tag{2.71.a}$$

$$\boldsymbol{f}(\boldsymbol{\beta}) \ = \ \widetilde{\boldsymbol{H}} \boldsymbol{\beta} + \widetilde{\boldsymbol{T}} \boldsymbol{A}^{-1} (\boldsymbol{y} - \boldsymbol{H} \boldsymbol{\beta}) \in \mathbb{R}^k, \tag{2.71.b}$$

$$\boldsymbol{\Sigma} \ = \ \widetilde{\boldsymbol{A}} - \widetilde{\boldsymbol{T}} \boldsymbol{A}^{-1} \widetilde{\boldsymbol{T}}^T \in \mathbb{R}^{k \times k}. \tag{2.71.c}$$

We can also think of this as the distribution of the vector $\widetilde{\boldsymbol{Y}}$ given $\boldsymbol{\beta}$ and σ^2, where the mean of this distribution depends on a fixed vector $\boldsymbol{y}$.

Let us now apply Lemma 2.5.4 to the family of distributions $\mathbb{P}_{\boldsymbol{b},s} = N(\boldsymbol{b}, s^2 \boldsymbol{\Sigma})$, with location parameter $\boldsymbol{b} \in \mathbb{R}^q$ and scale parameter $s > 0$. The vector $\boldsymbol{Y}$ in the statement of the lemma is here $\widetilde{\boldsymbol{Y}}$, the function $\boldsymbol{f}(\cdot)$ is as above, and $g(\cdot)$ is the identity of $\mathbb{R}^+$.

Given the equations in (2.71), the hypothesis of the lemma is fulfilled. Hence, the random vector:

$$\boldsymbol{X_1} = \frac{\widetilde{\boldsymbol{Y}} - \boldsymbol{f}(\boldsymbol{\beta})}{\sigma} \in \mathbb{R}^k \tag{2.72}$$

is independent of both $\boldsymbol{\beta}$ and σ. The distribution of $\boldsymbol{X_1}$ is then the same as its conditional distribution on $\boldsymbol{\beta}$ and σ^2. This is normal for $\widetilde{\boldsymbol{Y}}$, hence $\boldsymbol{X_1} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$.

We rewrite (2.72) as:

$$\widetilde{\boldsymbol{Y}} = \boldsymbol{f}(\boldsymbol{\beta}) + \sigma \boldsymbol{X_1}, \quad \boldsymbol{X_1} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}) \text{ independent of } \boldsymbol{\beta}, \sigma. \tag{2.73}$$

We can iterate the process, this time on $\boldsymbol{\beta}$. By assumption (2.65.b), we have $\boldsymbol{\beta} \,|\, \sigma^2 \sim N(\boldsymbol{b}, \sigma^2 \boldsymbol{B}^{-1})$. Let us again apply Lemma 2.5.4, in the simpler case where the "location" function $\boldsymbol{f}(\cdot)$ is constant and equal to the fixed vector $\boldsymbol{b}$, and $g(\cdot)$ is again the identity of $\mathbb{R}^+$. We get:

$$\boldsymbol{\beta} = \boldsymbol{b} + \sigma \boldsymbol{X_2}, \quad \boldsymbol{X_2} \sim N(\boldsymbol{0}, \boldsymbol{B}^{-1}) \text{ independent of } \sigma. \tag{2.74}$$

Finally, since $\sigma^2 \sim IG(a, s)$, we can write (cf. (2.14)):

$$\sigma^2 = \frac{2s}{V}, \quad V \sim \chi^2(2a). \tag{2.75}$$

Independence of σ then translates into independence of V.

Summary so far: Equations (2.73)–(2.75) give us the following:

$$\widetilde{\boldsymbol{Y}} = \boldsymbol{f}(\boldsymbol{\beta}) + \sigma \boldsymbol{X_1}, \quad \boldsymbol{\beta} = \boldsymbol{b} + \sigma \boldsymbol{X_2}, \quad \sigma^2 = \frac{s}{a} \frac{2a}{V}, \tag{2.76}$$

with

$$\boldsymbol{X_1} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}), \quad \boldsymbol{X_2} \sim N(\boldsymbol{0}, \boldsymbol{B}^{-1}), \quad V \sim \chi^2(2a) \tag{2.77}$$

all independent of one another.

We can now plug the second and third identities of (2.76) into the first identity of the same equation, in order to have a more explicit expression for $\widetilde{\boldsymbol{Y}} \in \mathbb{R}^k$. Starting from the expression of $\boldsymbol{f}(\boldsymbol{\beta}) \in \mathbb{R}^k$ in (2.71.b), we get:

$$\widetilde{\boldsymbol{Y}} = \widetilde{\boldsymbol{H}}\boldsymbol{\beta} + \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta}) + \sigma \boldsymbol{X_1}$$

$$\begin{aligned} &\overset{(\boldsymbol{\beta}=\boldsymbol{b}+\sigma\boldsymbol{X_2})}{=} \widetilde{\boldsymbol{H}}\boldsymbol{b} + \sigma\widetilde{\boldsymbol{H}}\boldsymbol{X_2} + \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}\boldsymbol{y} - \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}\boldsymbol{H}(\boldsymbol{b}+\sigma\boldsymbol{X_2}) \;+\; \sigma\boldsymbol{X_1} \\ &= \quad \widetilde{\boldsymbol{H}}\boldsymbol{b} + \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b}) + \sigma\Big[(\widetilde{\boldsymbol{H}} - \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}\boldsymbol{H})\boldsymbol{X_2} + \boldsymbol{X_1}\Big] \\ &= \quad \widetilde{\boldsymbol{H}}\boldsymbol{b} + \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b}) + \sigma\boldsymbol{X_3}\,, \end{aligned} \tag{2.78}$$

where therefore $\boldsymbol{X_3}$ is independent of V, since $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are independent of V:

$$\boldsymbol{X_3} \sim N(\boldsymbol{0}, \boldsymbol{W})\,, \quad \boldsymbol{W} = \boldsymbol{\Sigma} + (\widetilde{\boldsymbol{H}} - \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}\boldsymbol{H})\boldsymbol{B}^{-1}(\widetilde{\boldsymbol{H}} - \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}\boldsymbol{H})^T\,. \tag{2.79}$$

Plugging the expression of σ from (2.76) into (2.78), we get:

$$\widetilde{\boldsymbol{Y}} = \widetilde{\boldsymbol{H}}\boldsymbol{b} + \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b}) + \sqrt{\frac{2a}{V}}\,\boldsymbol{X}\,, \tag{2.80}$$

where $\boldsymbol{X} = \sqrt{s/a}\,\boldsymbol{X_3} \sim N\left(\boldsymbol{0},\, \frac{s}{a}\,\boldsymbol{W}\right)$ is independent of $V \sim \chi^2(2a)$.

Equation (2.80) proves that the distribution of $\widetilde{\boldsymbol{Y}}$ given $\boldsymbol{y}$, now unconditioned on $\boldsymbol{\beta}$ and σ^2, is indeed multivariate Student-t, with:

1. $2a$ degrees of freedom;
2. mean $\widetilde{\boldsymbol{H}}\boldsymbol{b} + \widetilde{\boldsymbol{T}}\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b}) \in \mathbb{R}^k$;
3. kernel matrix $\frac{s}{a}\,\boldsymbol{W} \in \mathbb{R}^{k\times k}$.

Given the generality of $\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}$, this proves that $\eta(\cdot)$ conditioned on $\boldsymbol{Y} = \boldsymbol{y}$ is a t-process with $2a$ degrees of freedom. The j^{th} component of the mean, when read for a general $\boldsymbol{x}$ rather than $\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}$, coincides with the expression in (2.67). Similarly, the component (j_1, j_2) of $\frac{s}{a}\,\boldsymbol{W} \in \mathbb{R}^{k\times k}$, for general $\boldsymbol{x}$ and $\boldsymbol{x}'$, reads as in (2.68). This completes the proof. □

The previous result is of capital importance, since it provides the distribution of the emulator, *i.e.*, the posterior distribution of the process $\eta(\cdot)$, given the observed simulator outputs $f(\boldsymbol{x_1}), \ldots, f(\boldsymbol{x_n})$. Notice that the transition from a prior Gaussian distribution to a posterior Student-t distribution is due to having unconditioned the law of the process $\eta(\cdot)$ from the pair $(\boldsymbol{\beta}, \sigma^2)$, rather than to having conditioned $\eta(\cdot)$ on the observed simulator outputs $f(\boldsymbol{x_i})$.

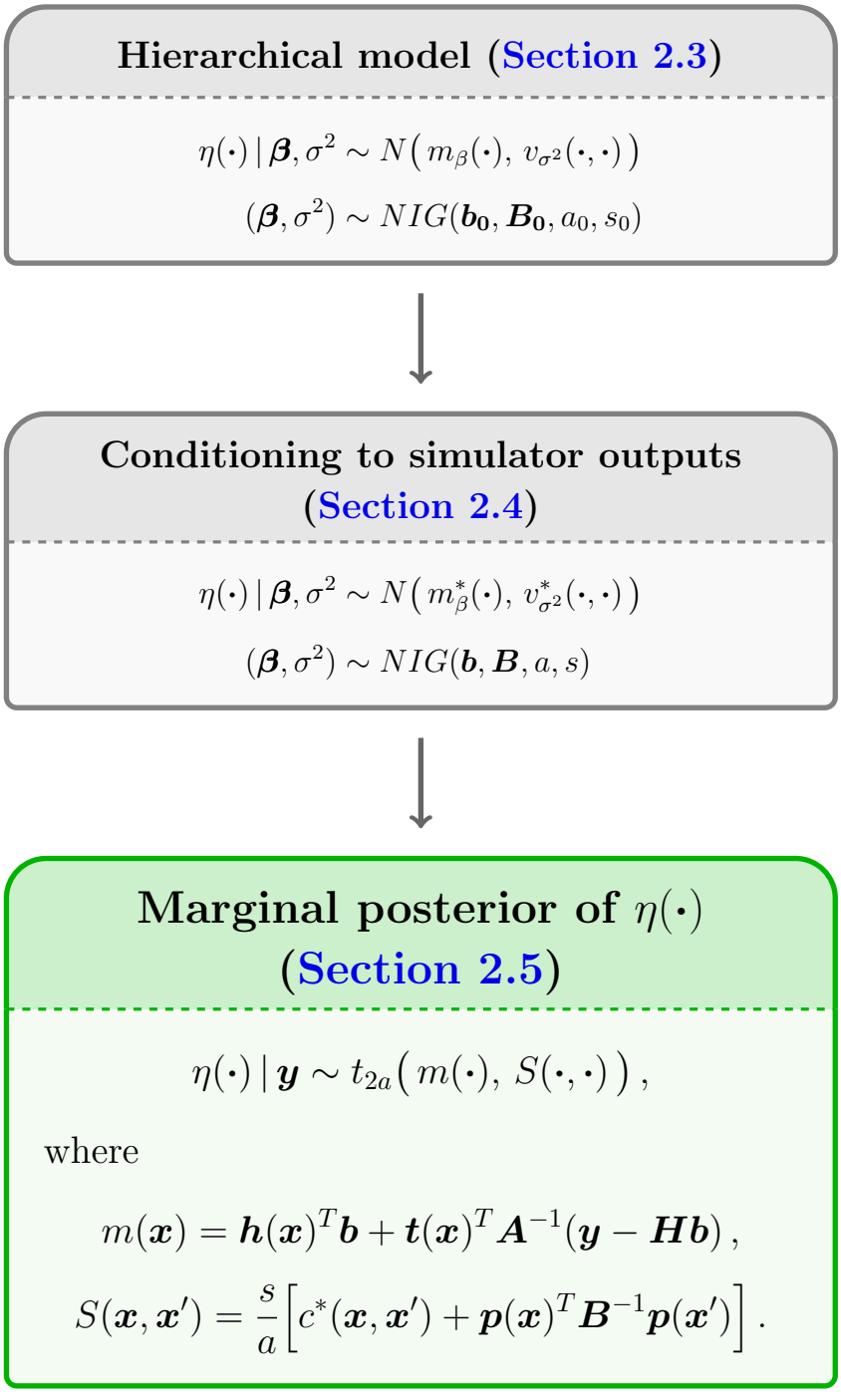


Figure 2.3: Continued from Figure 2.2, the last step behind the construction of a GP emulator is depicted in the green box. Details are in Theorem 2.5.5. The distribution is the one of a t-process, with the displayed mean function $m(\cdot)$ and kernel function $S(\cdot,\cdot)$.

2.6. Classical Prior Choice

In order to build an emulator, the hyperparameters $\boldsymbol{b_0}, \boldsymbol{B_0}, a_0, s_0$ used to define the marginal prior distribution of $\boldsymbol{\beta} \in \mathbb{R}^q$ and $\sigma^2 > 0$ need to be specified. Where available, information on $m_\beta(\cdot)$ and $v_{\sigma^2}(\cdot,\cdot)$ from the problem can be translated into information about the parameters $\boldsymbol{\beta}$ and σ^2, so as to make informed decisions about the values of $\boldsymbol{b_0}$, $\boldsymbol{B_0}$, a_0, and s_0 to choose. These may also be estimated via simpler regression models, such as linear regression. In most applications, however, the choice of a *non-informative* prior for the pair $(\boldsymbol{\beta}, \sigma^2)$ is made (O'Hagan [1992], Bonceur et al. [2015]). The term non-informative refers to a distribution which takes values over large real intervals, with approximately uniform probability. In the case of

$$(\boldsymbol{\beta}, \sigma^2) \;\sim\; NIG(\boldsymbol{b_0}, \boldsymbol{B_0}, a_0, s_0)\,,$$

the hyperparameters can formally be chosen to yield a "flat" marginal density for $\boldsymbol{\beta}$, and an "as flat as possible" density for σ^2. In order to give this a more precise meaning, let us recall from (2.16) the form of the prior density:

$$\pi(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{(\sigma^2)^{a_0+1+q/2}} \exp\left[-\frac{1}{2\sigma^2}\left(2s_0 + (\boldsymbol{\beta}-\boldsymbol{b_0})^T \boldsymbol{B_0}(\boldsymbol{\beta}-\boldsymbol{b_0})\right)\right]. \tag{2.81}$$

If we formally set $\boldsymbol{B_0} = \boldsymbol{0_{q\times q}}$, the dependence on $\boldsymbol{\beta}$ vanishes. This mathematically improper choice is equivalent to attributing infinite conditional variance to $\boldsymbol{\beta}$, since indeed $\boldsymbol{\beta}\,|\,\sigma^2 \sim N(\boldsymbol{b_0}, \sigma^2 \boldsymbol{B_0}^{-1})$. Moreover, setting $s_0 = 0$ in (2.81) will lead an improper polynomial density for σ^2: improper, in that the integral

$$\int_0^{+\infty} \frac{1}{(\sigma^2)^\gamma}\, d\sigma^2, \qquad \gamma = a_0 + 1 + \frac{q}{2},$$

diverges for any real value of γ. However, $\gamma = 1$ is the only value for which the integral diverges in any neighbourhood of both zero and infinity. Therefore, in order to have an "as flat as possible" prior for σ^2, the choice $a_0 = -q/2$ corresponding to $\gamma = 1$ can be made. Such a choice yields:

$$\pi(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^2}\,, \tag{2.82}$$

a notation often used in the literature to denote the common non-informative choice made for the prior of $(\boldsymbol{\beta}, \sigma^2)$. However, as pointed out, setting

$$\boldsymbol{B_0} = \mathbf{0}_{\boldsymbol{q}\times\boldsymbol{q}}, \quad s_0 = 0, \quad a_0 = -\frac{q}{2} \tag{2.83}$$

does not yield a proper prior density in (2.81). In the following, we propose a limit argument.

Note of the author

The proposed argument needs a remark, concerning the choice of the prior parameter a_0. Specifically, while a proper limit argument can be carried out for the other parameters, this is not the case of a_0, whose negative value lies outside the closure of the set of allowed prior values. Formally, its use leads nonetheless a proper density as posterior, which we derive below. It should be noticed, however, that this can only be given a formal "posterior" meaning, at least within our presentation, since it is not directly associated with any prior.

If $\boldsymbol{B_0}$ is any strictly positive definite matrix, $s_0 > 0$, and $a_0 > -q/2$, then the expression in (2.81) has the following properties: it decays exponentially in $\boldsymbol{\beta} \in \mathbb{R}^q$ as $\|\boldsymbol{\beta}\|$ tends to infinity, it is bounded in a neighbourhood of $\sigma^2 = 0$, and it is integrable when σ^2 tends to infinity, since $\gamma > 1$. While these conditions are sufficient to make (2.81) integrable in $\boldsymbol{\beta}$ for fixed σ^2, and in σ^2 for fixed $\boldsymbol{\beta}$, they are not sufficient to guarantee integrability over $\mathbb{R}^q \times \mathbb{R}^+$. Formally, however, we can compute the posterior values, equations (2.35.a)–(2.35.d), in the case where $\boldsymbol{b_0} = \mathbf{0}$ and in the limit:

$$\boldsymbol{B_0} \overset{\text{PD}}{\searrow} 0\,, \quad s_0 \searrow 0\,, \quad a_0 \searrow -\frac{q}{2}\,. \tag{2.84}$$

The first limit is taken over any sequence of positive definite matrices tending to the zero matrix, for example $\boldsymbol{B_0} = \varepsilon\, \boldsymbol{I_q}$ when $\varepsilon \searrow 0$: the limit will not depend on the specific sequence. From (2.35.a), we obtain:

$$\lim_{\boldsymbol{B_0}\to\mathbf{0}} \boldsymbol{B} = \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{H} \in \mathbb{R}^{q\times q}. \tag{2.85}$$

Hence, from (2.35.b) we obtain the posterior expression of $\boldsymbol{b}$, and from (2.35.c) the one of a:

$$\lim_{\boldsymbol{B_0}\to\boldsymbol{0}} \boldsymbol{b} = \left(\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y} \in \mathbb{R}^q, \tag{2.86}$$

$$\lim_{a_0\to\frac{-q}{2}} a = \frac{n-q}{2}\,. \tag{2.87}$$

To work out the limit expression of s from (2.35.d), we need to expand $\boldsymbol{F}^{-1}$. Since $\boldsymbol{F} = \boldsymbol{A}+\boldsymbol{H}\boldsymbol{B_0}^{-1}\boldsymbol{H}^T \in \mathbb{R}^{n\times n}$, we can again take advantage of Lemma B.1 in Appendix. This yields the following expression for $\boldsymbol{F}^{-1}$:

$$\boldsymbol{F}^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{H}\left(\boldsymbol{B_0} + \boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1} \ \in \mathbb{R}^{n\times n}. \tag{2.88}$$

Hence, the following is the posterior limit expression of s:

$$\begin{aligned}\lim_{\substack{\boldsymbol{B_0}\to\boldsymbol{0}\\ s_0\to 0}} s &= \lim_{\substack{\boldsymbol{B_0}\to\boldsymbol{0}\\ s_0\to 0}} \left[s_0 + \frac{(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})^T\boldsymbol{F}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})}{2}\right] \\ &= \frac{1}{2}\,(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})^T\left[\boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}H\left(\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\right](\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b_0})\,. \end{aligned} \tag{2.89}$$

This expression can be significantly simplified. First, notice that multiplying the matrix

$$\boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{H}\left(\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}$$

by $\boldsymbol{H}$ on the right-hand side or by $\boldsymbol{H}^T$ on the left-hand side, returns in both cases the null matrix. Therefore, we can simplify (2.89) into the following:

$$\lim_{\substack{\boldsymbol{B_0}\to\boldsymbol{0}\\ s_0\to 0}} s = \frac{1}{2}\,\boldsymbol{y}^T\left[\boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{H}\left(\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\right]\boldsymbol{y}\,. \tag{2.90}$$

Moreover, given the limit expression of $\boldsymbol{b}$ in (2.86), we have:

$$\begin{aligned}\lim_{\substack{\boldsymbol{B_0}\to\boldsymbol{0}\\ s_0\to 0}} s &= \frac{1}{2}\,\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} \;-\; \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{b} \\ &= \frac{1}{2}\,\boldsymbol{y}^T\boldsymbol{A}^{-1}\left(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b}\right)\end{aligned}$$

$$\overset{(\star)}{=} \frac{1}{2} (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})^T \boldsymbol{A}^{-1} (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}) . \tag{2.91}$$

We have made a slight abuse of notation, by simply denoting with $\boldsymbol{b}$ the limit expression in (2.86). Notice that the equality $(\star)$ provides a more symmetric expression for s: it simply follows from $(\boldsymbol{H}\boldsymbol{b})^T \boldsymbol{A}^{-1} (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}) = 0$, which is trivial to check using the expression (2.86) of $\boldsymbol{b}$.

Hence, in the limit (2.84), all the posterior hyperparameters converge to well defined quantities, which identify a proper posterior density for $(\boldsymbol{\beta}, \sigma^2)$.

2.7. Summary of Emulation Setting and Formulas

Even if not conceptually advanced, some of the results in the previous sections have been reasonably technical. In the following, we concisely summarise the main steps and assumptions used to build a GP emulator: the aim is both of providing a unifying overview of its Bayesian setting and final formulas, and to produce a compact reference for future use within this work.

The starting point are the outputs of a complex simulator $f(\cdot)$ on n design points $\boldsymbol{x_1}, \dots, \boldsymbol{x_n} \in \mathcal{P}$. $\mathcal{P} \subseteq \mathbb{R}^p$ is the input space of the simulator. The observed outputs are denoted by $y_i = f(\boldsymbol{x_i})$. We want to build a stochastic process $\eta(\cdot)$ providing predictions of the simulator output corresponding to any input $\boldsymbol{x} \in \mathcal{P}$.

1. The process $\eta(\cdot)$ is modelled as a Gaussian process. The prior mean function $m_\beta(\cdot)$ and covariance function $v_{\sigma^2}(\cdot, \cdot)$ of η are specified as follows:

$$m_\beta(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T \boldsymbol{\beta}, \quad \boldsymbol{x} \in \mathcal{P}, \tag{2.92}$$

$$v_{\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \, c(\boldsymbol{x}, \boldsymbol{x}'), \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P} , \tag{2.93}$$

   where

   - $\boldsymbol{h}(\cdot)$ is a vector of q real functions of the inputs $\boldsymbol{x}$, $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^q$;
   - $c(\cdot, \cdot)$ is a valid covariance function;
   - $\boldsymbol{\beta} \in \mathbb{R}^q$ and $\sigma^2 > 0$ are coefficients.

2. The pair $(\boldsymbol{\beta}, \sigma^2)$ is endowed with the following prior distribution:

$$(\boldsymbol{\beta}, \sigma^2) \sim NIG(\boldsymbol{b_0}, \boldsymbol{B_0}, a_0, s_0)\,. \tag{2.94}$$

   The joint model $(\eta(\cdot), \boldsymbol{\beta}, \sigma^2)$ is conditioned on $\{\eta(\boldsymbol{x_i}) = y_i\}_{i=1,\dots,n}$. The posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$ is $NIG(\boldsymbol{b}, \boldsymbol{B}, a, s)$, Proposition 2.4.2.

3. The pair $(\boldsymbol{\beta}, \sigma^2)$ is integrated out of the posterior model. The marginal distribution of $\eta(\cdot)$, conditioned on $\{\eta(\boldsymbol{x_i}) = y_i\}_i$ only, is the one of a Student-t process, Theorem 2.5.5.

In the literature, the choice of a non-informative prior for the pair $(\boldsymbol{\beta}, \sigma^2)$ is often made, see Section 2.6. In this case, the posterior hyperparameters $\boldsymbol{b}, \boldsymbol{B}, a, s$ read as follows:

$$\boldsymbol{B} = \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{H} \in \mathbb{R}^{q \times q}, \tag{2.95.a}$$

$$\boldsymbol{b} = \boldsymbol{B}^{-1} \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{y} \in \mathbb{R}^{q}, \tag{2.95.b}$$

$$a = \frac{n-q}{2}\,, \tag{2.95.c}$$

$$s = \frac{1}{2}\,(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})^T \boldsymbol{A}^{-1} (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})\,. \tag{2.95.d}$$

Under this choice, the resulting emulator is Student-t process with $n - q$ degrees of freedom, mean function

$$m(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T \boldsymbol{b} + \boldsymbol{t}(\boldsymbol{x})^T \boldsymbol{A}^{-1} (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})\,, \tag{2.96}$$

and covariance function

$$v(\boldsymbol{x}, \boldsymbol{x}') = \widehat{\sigma}^2 \left[c(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{t}(\boldsymbol{x})^T \boldsymbol{A}^{-1} \boldsymbol{t}(\boldsymbol{x}') + \boldsymbol{p}(\boldsymbol{x})^T \boldsymbol{B}^{-1} \boldsymbol{p}(\boldsymbol{x}') \right]. \tag{2.97}$$

The positive quantity

$$\widehat{\sigma}^2 = \frac{(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})^T \boldsymbol{A}^{-1} (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})}{n - q - 2} \tag{2.98}$$

represents the posterior mean of σ^2.

For the convenience of the reader and for future reference, we recall below the expres-

sion of the matrices $\boldsymbol{A}$ and $\boldsymbol{H}$ and of the functions $\boldsymbol{t}(\cdot)$ and $\boldsymbol{p}(\cdot)$:

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{h}(\boldsymbol{x_1})^T \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x_n})^T \end{pmatrix} \in \mathbb{R}^{n\times q}, \quad \boldsymbol{A} = \begin{pmatrix} c(\boldsymbol{x_1},\boldsymbol{x_1}) & \dots & c(\boldsymbol{x_1},\boldsymbol{x_n}) \\ \vdots & \ddots & \vdots \\ c(\boldsymbol{x_n},\boldsymbol{x_1}) & \dots & c(\boldsymbol{x_n},\boldsymbol{x_n}) \end{pmatrix} \in \mathbb{R}^{n\times n}, \quad (2.99)$$

$$\boldsymbol{t}(\boldsymbol{x}) = \big(c(\boldsymbol{x},\boldsymbol{x_1}),\dots,c(\boldsymbol{x},\boldsymbol{x_n})\big)^T \in \mathbb{R}^n, \quad \boldsymbol{x}\in\mathcal{P}, \quad (2.100)$$

$$\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}) \ \in \mathbb{R}^q, \quad \boldsymbol{x}\in\mathcal{P}. \quad (2.101)$$

2.8. The Case of Chaotic and Stochastic Simulators

So far, we have considered the case of a deterministic simulator. In this section we introduce a tool that is useful if the simulator is stochastic, but also if the simulator is deterministic but chaotic: that is, if the simulator response $f(\boldsymbol{x})$ is always the same across different runs at the same input $\boldsymbol{x}\in\mathcal{P}$, but $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ are generally very different for almost identical inputs $\boldsymbol{x}\neq\boldsymbol{x}'\in\mathcal{P}$. This case will be of particular interest in Part II of this work.

To motivate the rest of this section, we start with a simple observation. We state it as a lemma due to the importance of the result, although this is an immediate consequence of the emulation setting itself.

Lemma 2.8.1. *Let $m(\cdot)$ and $v(\cdot,\cdot)$ be the mean and covariance function of an emulator built with set of design points $\mathcal{D}=\{\boldsymbol{x_i}\}_{i=1,\dots,n}\subset\mathcal{P}$ and corresponding outputs $\{y_i\}_i$. Then the emulator perfectly interpolates the outputs. That is:*

$$m(\boldsymbol{x_i}) = y_i, \quad v(\boldsymbol{x_i},\boldsymbol{x_i}) = 0. \quad (2.102)$$

Proof. The result is a consequence of the emulation setting, particularly of the step where the process $\eta(\cdot)$ has been conditioned on the event $\{\eta(\boldsymbol{x_i})=y_i\}_{i=1,\dots,n}$. □

Remark 2.5. Even without any probabilistic interpretation, the claim of Lemma 2.8.1 can be proven via a simple algebraic check. Let us denote by $\boldsymbol{e_i}\in\mathbb{R}^n$ the i^{th} vector

of the canonical basis of $\mathbb{R}^n$. Then the i^{th} column of the matrix identity $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I_n}$ reads $\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x_i}) = \boldsymbol{e_i}$. Hence:

$$\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x_i}) = \boldsymbol{H}^T\boldsymbol{e_i} = \boldsymbol{h}(\boldsymbol{x_i})\,. \tag{2.103}$$

Thus, from (2.101):

$$\boldsymbol{p}(\boldsymbol{x_i}) = \boldsymbol{h}(\boldsymbol{x_i}) - \boldsymbol{h}(\boldsymbol{x_i}) = \boldsymbol{0_{q\times 1}}\,. \tag{2.104}$$

Similarly, from $\boldsymbol{t}(\boldsymbol{x_i})^T\boldsymbol{A}^{-1} = \boldsymbol{e_i}^T$, we get:

$$\boldsymbol{t}(\boldsymbol{x_i})^T\boldsymbol{A}^{-1}\boldsymbol{H} = \boldsymbol{h}(\boldsymbol{x_i})^T\,. \tag{2.105}$$

This yields (see equations (2.96) and (2.97)):

$$m(\boldsymbol{x_i}) = \boldsymbol{h}(\boldsymbol{x_i})^T\boldsymbol{b} + \boldsymbol{e_i}^T\boldsymbol{y} - \boldsymbol{h}(\boldsymbol{x_i})^T\boldsymbol{b} = y_i\,, \tag{2.106}$$

$$\begin{aligned} v(\boldsymbol{x_i},\boldsymbol{x_i}) &= \widehat{\sigma}^2\Big[c(\boldsymbol{x_i},\boldsymbol{x_i}) - \boldsymbol{e_i}^T\boldsymbol{t}(\boldsymbol{x_i}) + 0\Big] \\ &= \widehat{\sigma}^2\Big[c(\boldsymbol{x_i},\boldsymbol{x_i}) - c(\boldsymbol{x_i},\boldsymbol{x_i})\Big] = 0\,, \end{aligned} \tag{2.107}$$

which completes the purely algebraic check of (2.102).

2.8.1. Adding Observational Variance (Nugget Term)

If the simulator is stochastic or chaotic, property (2.102) may not be desirable to reproduce its nature. In such a case, it may be appropriate to interpret the simulator output y_i as one instance among different outcomes corresponding to the input $\boldsymbol{x_i}$ – or to a very small neighbourhood of the latter in the chaotic case – and hence to build an emulator whose predictions are truly stochastic even at the design points. This way, the emulator may be able to identify a smooth mean curve $m(\cdot)$ not deterministically interpolating the pairs $(\boldsymbol{x_i}, y_i)$, with the shifts $y_i - m(\boldsymbol{x_i})$ at the design points being explained by the additional variance due to the simulator nature.

To implement this approach, we use the so-called "nugget term", investigated in detail in Andrianakis and Challenor [2012]. In the paper, this is as well investigated under a numerical stability point of view. In the following, we present the use of the nugget and

derive the formulas of practical interest via a different approach than the classical one, *i.e.*, by using continuity arguments to identify two different components of a relevant emulator.

The nugget correction consists in adding a Kronecker-δ term, with continuous inputs, to the prior covariance function $c(\cdot, \cdot)$ of an emulator. That is, we replace $c(\boldsymbol{x}, \boldsymbol{x}')$ with the following:

$$c_\nu(\boldsymbol{x}, \boldsymbol{x}') = c_s(\boldsymbol{x}, \boldsymbol{x}') + \nu\, \delta_{\boldsymbol{x},\boldsymbol{x}'}, \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}, \tag{2.108}$$

where $\nu > 0$ and

$$\delta_{\boldsymbol{x},\boldsymbol{x}'} = \begin{cases} 1 & \text{if } \boldsymbol{x} = \boldsymbol{x}' \\ 0 & \text{otherwise} \end{cases}, \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}. \tag{2.109}$$

Before investigating the effects of using a covariance function of the form (2.108) in emulation, we make an important remark and discuss the associated terminology.

Remark and Terminology

The function $\delta_{\boldsymbol{x},\boldsymbol{x}'}$, as a function of the two inputs $\boldsymbol{x}$ and $\boldsymbol{x}' \in \mathcal{P}$, is a valid covariance function (Definition 1.4.2): it is clearly symmetric and as well positive definite. Due to Theorem 1.4.3, a Gaussian process with input space $\mathcal{P} \in \mathbb{R}^p$, constantly zero mean, and covariance function equal to $\delta_{\boldsymbol{x},\boldsymbol{x}'}$ exists. However, the paths of such a process are extremely irregular: in fact, they have zero probability of being measurable as functions of $\boldsymbol{x} \in \mathcal{P}$ (in this regard, recall Remark 1.4 on page 26). Nonetheless, throughout the emulation literature, such a GP is frequently used, especially in conjunction with the "nugget term". The reason is that, while being formally defined on a continuous space, in any emulation application the process is only evaluated at a finite number of sites in $\mathcal{P}$. We'll see an example of this in Subsection 4.9.2.

The way the above process is named is not uniform across the emulation literature: it is sometimes referred to as white noise[a] (*e.g.*, Craig et al. [2001]), or it is itself called the "nugget term" (*e.g.*, Vernon et al. [2010]), or it is just defined in terms of its mean and covariance functions and given no specific name (*e.g.*, Goldstein and Rougier [2004]). In all these cases, the process is used within an emulator to account for the so-called "residual" variability, the one in the simulator output

which cannot be explained in terms of the input $\boldsymbol{x}$ alone (for example, because of an intrinsic stochastic behaviour of the simulator, or because of the role of factors which have not been included in $\boldsymbol{x}$). In this work, for ease of future reference, we refer to a GP with zero mean and covariance function equal to $\delta_{\boldsymbol{x},\boldsymbol{x}'}$ as to Gaussian noise. For such a process $\varepsilon(\cdot)$, we use the notation $\varepsilon(\cdot) \sim \mathcal{GN}(1)$. More generally, we write:

$$\varepsilon(\cdot) \sim \mathcal{GN}(\sigma^2)$$

if $\varepsilon(\cdot)/\sigma \sim \mathcal{GN}(1)$ for $\sigma > 0$. As mentioned above, values of this process will only be used simultaneously at a finite number of sites only.

[a] Note that, in stochastic analysis, the term *white noise* refers to a different mathematical object, *i.e.*, to a generalised random process. Within an appropriate framework (analogous to the one of generalised functions, and which is not the aim of this work to introduce), white noise can be viewed as the "derivative" of Brownian motion. It is therefore clear that such a process cannot be defined in the classical sense and it is not real-valued, since the paths of Brownian Motion are almost surely not differentiable at any point (they are not even α-Hölder continuous for $\alpha \geq 1/2$).

In (2.108), we suppose that the function $c_s(\cdot,\cdot)$ is a valid covariance function, which therefore encodes the main prior covariance structure of the emulator: typical examples are presented in Subsection 1.4.4. In the following, we assume, that $c_s(\cdot,\cdot)$ it is at least continuous, although in practical applications it is often more regular than that. In this regard, the subscript "s" may be thought of as standing for the word "smooth" (in contrast to the highly irregular Kronecker-δ term), although we simply assume continuity and not infinite differentiability of $c_s(\cdot,\cdot)$.

In equation (2.108), the Kronecker-δ term adds independent variance to $c_s(\cdot,\cdot)$. Since $c_s(\cdot,\cdot)$ is a valid covariance function and $\nu > 0$, it is straightforward to see that the function $c_\nu(\cdot,\cdot)$ is as well a valid covariance function. However, $c_\nu(\cdot,\cdot)$ is discontinuous under the assumed regularity of $c_s(\cdot,\cdot)$. This induces an emulator with discontinuous paths, in accordance with (although, strictly speaking, not being necessarily implied by) the results of Subsection 1.4.2, but which still predicts the deterministic value y_i at the point $\boldsymbol{x_i}$. Such an emulator, $\eta_\nu(\cdot)$, does not seem to achieve the goals set out at the beginning of the section: smooth non-interpolating mean and truly stochastic predictions. However, we would like to recognise that it can be decomposed

as $\eta_\nu(\cdot) = \eta(\cdot) + \varepsilon(\cdot)$ almost everywhere in $\mathcal{P}$, where:

- ▷ the process $\eta(\cdot)$ is continuous, and it does not deterministically interpolate the simulated outputs y_i;
- ▷ the residual $\varepsilon(\cdot)$ can be identified as Gaussian noise.

We formalise the previous claim in Theorem 2.8.3. In the rest of this section, it will be convenient to differentiate between quantities computed with respect to the prior covariance $c_s(\cdot,\cdot)$, or with respect to $c_\nu(\cdot,\cdot)$. We refer in particular to the quantities $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{b}$, $\boldsymbol{t}(\cdot)$, $\widehat{\sigma}^2$, $m(\cdot)$, $v(\cdot,\cdot)$ appearing in equations (2.95)–(2.101). To this aim, we add the subscript "s" or "ν", according to the case.

We start by a proposition, which examines the continuity properties of an emulator built with prior covariance $c_\nu(\cdot,\cdot)$. Theorem 2.8.3 will follow. The regression function $h\colon \mathcal{P} \to \mathbb{R}^q$ is assumed continuous in both coming results, even if not explicitly stated.

Proposition 2.8.2. *Let $c_s(\cdot,\cdot)$ be a continuous covariance function, $c_\nu(\cdot,\cdot)$ as in (2.108), and $\eta_\nu(\cdot)$ the emulator built with prior covariance $c_\nu(\cdot,\cdot)$. Further denote by $\mathcal{D}$ the set design points of $\eta_\nu(\cdot)$, by $m_\nu\colon \mathcal{P} \to \mathbb{R}$ its mean function and by $v_\nu\colon \mathcal{P}\times\mathcal{P} \to \mathbb{R}$ its covariance function (equations (2.96), (2.97)). Then:*

1. *$m_\nu(\cdot)$ is discontinuous in $\boldsymbol{x}$ if and only if $\boldsymbol{x} \in \mathcal{D}$;*
2. *$v_\nu(\cdot,\cdot)$ is discontinuous in $(\boldsymbol{x},\boldsymbol{x}')$ if and only if $\boldsymbol{x} \in \mathcal{D}$, or $\boldsymbol{x}' \in \mathcal{D}$, or $\boldsymbol{x} = \boldsymbol{x}'$;*
3. *The unique continuous functions $m_c(\cdot)$ and $v_c(\cdot,\cdot)$ that extend $m_\nu(\cdot)$ and $v_\nu(\cdot,\cdot)$ outside their respective discontinuity regions read as follows:*

$$m_c(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{b} + \boldsymbol{t_s}(\boldsymbol{x})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}) \tag{2.110}$$

$$v_c(\boldsymbol{x},\boldsymbol{x}') = \widehat{\sigma}^2\left[\boldsymbol{c_s}(\boldsymbol{x},\boldsymbol{x}') - \boldsymbol{t_s}(\boldsymbol{x})^T\boldsymbol{A}^{-1}\boldsymbol{t_s}(\boldsymbol{x}') + \boldsymbol{p_s}(\boldsymbol{x})^T\boldsymbol{B}^{-1}\boldsymbol{p_s}(\boldsymbol{x}')\right], \tag{2.111}$$

where $\boldsymbol{A} := \boldsymbol{A_\nu}$, $\boldsymbol{B} := \boldsymbol{B_\nu}$, $\boldsymbol{b} := \boldsymbol{b_\nu}$, $\widehat{\sigma} := \widehat{\sigma}_\nu$. Notice that the constant matrix $\boldsymbol{A}$ appearing in the definition of $\boldsymbol{p_s}(\cdot)$, equation (2.101), is as well $\boldsymbol{A} = \boldsymbol{A_\nu}$.

Proof. Recall the expression of $m_\nu(\boldsymbol{x})$:

$$m_\nu(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{b} + \boldsymbol{t_\nu}(\boldsymbol{x})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b}). \tag{2.112}$$

Since $\boldsymbol{h}(\cdot)$ is continuous, and the vectors $\boldsymbol{b}$ and $\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{b})$ are constant, the discontinuity points of $m_\nu(\cdot)$ coincide with the ones of $\boldsymbol{t}_{\boldsymbol{\nu}}(\cdot)$. The i^{th} component of $\boldsymbol{t}_{\boldsymbol{\nu}}(\boldsymbol{x})$ reads:

$$c_\nu(\boldsymbol{x}, \boldsymbol{x_i}) = c_s(\boldsymbol{x}, \boldsymbol{x_i}) + \nu\delta_{\boldsymbol{x},\boldsymbol{x_i}}\,, \tag{2.113}$$

which is discontinuous if and only if $\boldsymbol{x} = \boldsymbol{x_i}$ ($c_s(\cdot,\cdot)$ is continuous by hypothesis). Hence, the vector $\boldsymbol{t}_{\boldsymbol{\nu}}(\cdot)$ is discontinuous in $\boldsymbol{x}$ if and only if $\boldsymbol{x}$ belongs to the set of design points $\mathcal{D}$: this proves claim 1. Moreover, being the set $\mathcal{D} \subseteq \mathcal{P}$ discrete, the function $m_\nu(\cdot)$, continuous on $\mathcal{P} \setminus \mathcal{D}$, can be continuously extended to $\mathcal{D}$ in a unique way. The extension is the one claimed in (2.110). Indeed, we have:

$$\lim_{\boldsymbol{x}\to\boldsymbol{x_i}} \delta_{\boldsymbol{x},\boldsymbol{x_i}} = 0 \quad \forall\, \boldsymbol{x_i} \in \mathcal{D}\,,$$

and therefore, considering (2.113) for all $i = 1, \dots, n$, we obtain:

$$\lim_{\boldsymbol{x}\to\boldsymbol{x_i}} \boldsymbol{t}_{\boldsymbol{\nu}}(\boldsymbol{x}) = \boldsymbol{t}_{\boldsymbol{s}}(\boldsymbol{x}) \quad \forall\, \boldsymbol{x_i} \in \mathcal{D}\,.$$

This shows that (2.112) continuously extends to (2.110) outside $\mathcal{D}$.

A similar reasoning can be carried out for the function $v_\nu(\cdot,\cdot)$:

$$v_\nu(\boldsymbol{x}, \boldsymbol{x}') = \widehat{\sigma}^2 \left[c_\nu(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{t}_{\boldsymbol{\nu}}(\boldsymbol{x})^T \boldsymbol{A}^{-1} \boldsymbol{t}_{\boldsymbol{\nu}}(\boldsymbol{x}') + \boldsymbol{p}_{\boldsymbol{\nu}}(\boldsymbol{x})^T \boldsymbol{B}^{-1} \boldsymbol{p}_{\boldsymbol{\nu}}(\boldsymbol{x}')\right]. \tag{2.114}$$

First, notice that the function $c_\nu(\cdot,\cdot)$ is discontinuous only along the diagonal

$$\mathcal{H} = \{(\boldsymbol{x}, \boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{P}\} \subseteq \mathcal{P} \times \mathcal{P} \tag{2.115}$$

of the set $\mathcal{P} \times \mathcal{P}$, and its continuous extension is clearly the function $c_s(\boldsymbol{x}, \boldsymbol{x}')$. Moreover, from (2.101), we see that the set of discontinuity of the function $\boldsymbol{p}_{\boldsymbol{\nu}}(\cdot)$ coincides with the one of the function $\boldsymbol{t}_{\boldsymbol{\nu}}(\cdot)$: *i.e.*, the set $\mathcal{D}$. This means that both the second and the third addend in (2.114) are discontinuous on $(\mathcal{D}\times\mathcal{P}) \cup (\mathcal{P}\times\mathcal{D})$. Therefore, outside the set

$$\mathcal{V} = \mathcal{H} \cup (\mathcal{D}\times\mathcal{P}) \cup (\mathcal{P}\times\mathcal{D})\,, \tag{2.116}$$

the function $v_\nu(\cdot,\cdot)$ is continuous. Furthermore, it is as well straightforward that,

inside $\mathcal{V}$, $v_\nu(\cdot,\cdot)$ is discontinuous. Hence, this proves claim 2.

Finally, the set $\mathcal{V}$ has zero Lebesgue measure in $\mathcal{P}\times\mathcal{P}\subseteq\mathbb{R}^p\times\mathbb{R}^p$, thus the continuous extension of $\boldsymbol{v_\nu}(\cdot,\cdot)$ outside $\mathcal{V}$ is unique. The claim that the extension is the one in (2.111) is, again, a straightforward consequence of the fact that, outside $\mathcal{D}$, $\boldsymbol{t_\nu}(\cdot)$ extends continuously to $\boldsymbol{t_s}(\cdot)$ and $\boldsymbol{p_\nu}(\cdot)$ extends continuously to what we have defined as $\boldsymbol{p_s}(\cdot)$ in the statement. The proof is thus complete.

□

The following result allows us to recognise both a regular and a random-noise component in an emulator $\eta_\nu(\cdot)$ built through $c_\nu(\cdot,\cdot)$. This is equation (2.117) below. We state the theorem and discuss its meaning and relevance within the context of this section, deferring its proof to immediately after the short discussion.

Theorem 2.8.3. *Let $m_\nu(\cdot)$ and $v_\nu(\cdot,\cdot)$ be the mean and covariance function of an emulator $\eta_\nu(\cdot)$ built with prior covariance $c_\nu(\cdot,\cdot)$ and set of design points $\mathcal{D}$. Consider a t-process $\eta(\cdot)$ on $\mathcal{P}$ with the same degrees of freedom as $\eta_\nu(\cdot)$, but continuous mean and covariance functions, $m_c(\cdot)$ and $v_c(\cdot,\cdot)$, as in (2.110), (2.111). Further define, for any two $\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}\in\mathcal{P}\setminus\mathcal{D}$, $j=1,2$, the two following random variables:*

$$\varphi_j=\eta(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})+\varepsilon_j\,,\quad \varepsilon_j\sim N\big(0,\nu\,\widehat{\sigma}_\nu^2\big), \tag{2.117}$$

with $\widehat{\sigma}_\nu$ defined as in (2.98), and ε_1 and ε_2 independent of each other and of $\eta(\cdot)$. Then, it holds:

$$\mathbb{E}\,(\varphi_j)=m_\nu(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})\,,\qquad j=1,2\,, \tag{2.118.a}$$

$$\mathrm{Cov}(\varphi_j,\varphi_h)=v_\nu(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}},\widetilde{\boldsymbol{x}}_{\boldsymbol{h}})\,,\qquad j,h\in\{1,2\}\,, \tag{2.118.b}$$

In the light of the theorem, we can sum up the results of this overall section as follows. If the prior covariance $c_\nu(\cdot,\cdot)$ is used in the emulation formulas (2.96)–(2.101), then the resulting emulator $\eta_\nu(\cdot)$ has discontinuous mean and covariance, and yet deterministically interpolates the observed simulator outputs. Theorem 2.8.3, however, ensures the following: outside of the design points, any finite-dimensional distribution of this emulator is the same as the one of the stochastic process $\varphi(\cdot)$ obtained by summing:

a) the continuous version $\eta(\cdot)$ of the emulator $\eta_\nu(\cdot)$, and

b) independent Gaussian noise $\varepsilon(\cdot)$ of constant variance $\nu\,\widehat{\sigma}^2_\nu$.

At the design points, the process $\eta(\cdot)$ provides non-deterministic predictions. Therefore, in practical applications, the process $\eta(\cdot)$ can be used as emulator built with prior covariance $c_\nu(\cdot,\cdot)$: this has a continuous mean (2.110) which does not go through the observed simulator outputs, and always provides non-deterministic predictions. For future reference, let us write below its mean and covariance functions (from Proposition 2.8.2):

$$m(\boldsymbol{x}) = h(\boldsymbol{x})^T\boldsymbol{b} + \boldsymbol{t_s}(\boldsymbol{x})^T\boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})\,, \tag{2.119}$$

$$v(\boldsymbol{x},\boldsymbol{x}') = \widehat{\sigma}^2\,\Big[c_\nu(\boldsymbol{x},\boldsymbol{x}') - \boldsymbol{t_s}(\boldsymbol{x})^T\boldsymbol{A}^{-1}\boldsymbol{t_s}(\boldsymbol{x}') + \boldsymbol{p_s}(\boldsymbol{x})^T\boldsymbol{B}^{-1}\boldsymbol{p_s}(\boldsymbol{x}')\Big]. \tag{2.120}$$

The matrix $\boldsymbol{A}$ (and all other quantities depending on it, such as $\boldsymbol{B}$, $\boldsymbol{b}$, $\widehat{\sigma}^2$) are computed with prior covariance $c_\nu(\cdot,\cdot)$.

Proof. (of Theorem 2.8.3) From the definition of φ_j, we immediately have:

$$\mathbb{E}(\varphi_j) = m_c(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) + 0 = m_\nu(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})\,,$$

since, by Proposition 2.8.2, $m_\nu(\cdot) \equiv m_c(\cdot)$ on $\mathcal{P}\setminus\mathcal{D}$. This proves equation (2.118.a).

Now, assuming without loss of generality that $\widetilde{\boldsymbol{x}}_{\boldsymbol{1}} \neq \widetilde{\boldsymbol{x}}_{\boldsymbol{2}}$, denote by $\boldsymbol{\Sigma_\varphi}$ the covariance matrix of the random vector (φ_1,φ_2), and by $\boldsymbol{\Sigma_\eta}$ the covariance matrix of the random vector $\big(\eta(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}),\eta(\widetilde{\boldsymbol{x}}_{\boldsymbol{2}})\big)$. From (2.117), we have:

$$\boldsymbol{\Sigma_\varphi} = \boldsymbol{\Sigma_\eta} + \nu\,\widehat{\sigma}^2_\nu\,\boldsymbol{I_2} \in \mathbb{R}^{2\times 2}, \tag{2.121}$$

where $\boldsymbol{I_2}$ denotes the 2×2 identity matrix. Any of the two off-diagonal elements in the previous equation reads as follows:

$$\mathrm{Cov}(\varphi_1,\varphi_2) = v_c(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}},\widetilde{\boldsymbol{x}}_{\boldsymbol{2}})\,. \tag{2.122}$$

This shows (2.118.b) in the $j \neq h$ case, since $v_c(\boldsymbol{x},\boldsymbol{x}') = v_\nu(\boldsymbol{x},\boldsymbol{x}')$ if $\boldsymbol{x}\neq\boldsymbol{x}'$ and $\boldsymbol{x},\boldsymbol{x}'\notin\mathcal{D}$: compare with Proposition 2.8.2, point 2.

Along the diagonal of (2.121), we have instead:

$$\begin{aligned}
\mathrm{Var}(\varphi_j) &= v_c(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) + \nu\, \widehat{\sigma}_\nu^2 \\
&= \widehat{\sigma}_\nu^2 \left[c_s(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \; - \; \boldsymbol{t_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})^T \boldsymbol{A}^{-1} \boldsymbol{t_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \; + \; \boldsymbol{p_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})^T \boldsymbol{B}^{-1} \boldsymbol{p_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \right] + \nu\, \widehat{\sigma}_\nu^2 \\
&= \widehat{\sigma}_\nu^2 \left[c_\nu(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \; - \; \boldsymbol{t_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})^T \boldsymbol{A}^{-1} \boldsymbol{t_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \; + \; \boldsymbol{p_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})^T \boldsymbol{B}^{-1} \boldsymbol{p_s}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \right],
\end{aligned}$$

given the definition of $c_\nu(\cdot, \cdot)$ in equation (2.108). Moreover, since $\boldsymbol{t_s}(\boldsymbol{x}) = \boldsymbol{t_\nu}(\boldsymbol{x})$ and $\boldsymbol{p_s}(\boldsymbol{x}) = \boldsymbol{p_\nu}(\boldsymbol{x})$ if $\boldsymbol{x} \notin \mathcal{D}$, we get:

$$\begin{aligned}
\mathrm{Var}(\varphi_j) &= \widehat{\sigma}_\nu^2 \left[c_\nu(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \; - \; \boldsymbol{t_\nu}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})^T \boldsymbol{A}^{-1} \boldsymbol{t_\nu}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \; + \; \boldsymbol{p_\nu}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}})^T \boldsymbol{B}^{-1} \boldsymbol{p_\nu}(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \right] \\
&= v_\nu(\widetilde{\boldsymbol{x}}_{\boldsymbol{j}}, \widetilde{\boldsymbol{x}}_{\boldsymbol{j}}) \,,
\end{aligned}$$

for $j = 1, 2$. This completes the proof. □

2.8.2. A Glimpse on Potential Identifiability Issues

We conclude this chapter with a brief analysis of the emulator model in specific limit cases, which are of relevance in practical applications. We include a nugget term in our analysis to be as general as possible within the setting presented so far. However, with the only aim of simplifying the notation, we suppose that the inputs are one-dimensional, $\boldsymbol{x} = x \in \mathbb{R}$, and that the prior mean is a linear (affine) function of x: that is, $\boldsymbol{h}(x) = (1, x)^T \in \mathbb{R}^2$. The coming analysis applies perfectly to the case of higher dimensional $\boldsymbol{x}$ or nonlinear basis functions $\boldsymbol{h}(\cdot)$.

The analysis is centred around the role played in the model by the correlation lengths, introduced in Section 1.5. Under our one-dimensional assumption, there will only be one correlation length, which we call $d > 0$. This is a measure of how far apart from each other two inputs $x, x' \in \mathcal{P} \subseteq \mathbb{R}$ need to be, in order for their prior correlation to decrease significantly. If d tends to 0, then all pairs of different inputs tend to be uncorrelated. On the other side, if d tends to infinity, all inputs become highly

correlated. Let us therefore set the notation. We consider:

$$c_\nu(x, x') = c(x, x') + \nu\, \delta_{x,x'}\,,^{3} \tag{2.123}$$

and we suppose that $c(\cdot,\cdot)$ is a correlation function that depends on a parameter $d > 0$, such that the following holds:

$$\lim_{d\to 0} c(x, x') = 0 \quad \forall\, x, x' \in \mathcal{P},\ x \neq x', \tag{2.124.a}$$

$$\lim_{d\to \infty} c(x, x') = 1 \quad \forall\, x, x' \in \mathcal{P}\,. \tag{2.124.b}$$

Under this notation and these assumptions, the prior model for the emulator can be written as follows:

$$\eta(x) = a + bx + \sigma\big[\psi(x) + \sqrt{\nu}\,\varepsilon(x)\big], \quad \text{where} \begin{cases} \psi(\cdot) \sim \mathcal{GP}\big(0, c(\cdot,\cdot)\big)\,, \\ \varepsilon(\cdot) \sim \mathcal{GN}(1)\,. \end{cases} \tag{2.125}$$

If $d \to 0$, then (2.125) tends to the following model:

$$\eta(x) = a + bx + \sigma\,\widetilde{\varepsilon}(x), \qquad \widetilde{\varepsilon}(\cdot) \sim \mathcal{GN}\big(1 + \nu\big)\,, \tag{2.126}$$

since the process $\psi(\cdot)$ tends itself to Gaussian noise. The one above is a simple linear regression model. In fact, it is not difficult to check that the classical linear regression formulas are recovered in this case from the emulation ones.

The case where $d \to \infty$ is instead more interesting. In such a case, any two inputs $x, x' \in \mathcal{P}$ have prior correlation one, which entails that the process $\psi(\cdot)$ becomes constant in $x \in \mathcal{P}$. Of course, by constant, we mean a random constant: in terms of the notation introduced in Subsection 1.3.2, we refer here to the randomness coming from the different ω of the sample space Ω. Hence, in distribution, we can write:

$$\psi(\cdot) \xrightarrow{d\to\infty} Z, \quad Z \sim N(0, 1)\,. \tag{2.127}$$

[3] It is not necessary in this section to add the subscript "s" to the function $c(\cdot,\cdot)$ on the RHS. We will not need to use the formulas (2.95)–(2.101), for which the subscripts "ν" and "s" had been previously introduced.

The model (2.125) in the case $d \to \infty$ therefore reads as follows:

$$\eta(x) = a + bx + \sigma\big[Z + \sqrt{\nu}\,\varepsilon(x)\big], \quad \varepsilon(\cdot) \sim \mathcal{GN}(1). \tag{2.128}$$

We can interpret this in two different (but necessarily equivalent) ways, as we outline below:

(a) $\eta(x) = a + bx + \sigma Y(x), \quad Y(x) = Z + \sqrt{\nu}\,\varepsilon(x);$

(b) $\eta(x) = \widetilde{a} + bx + (\sigma\sqrt{\nu})\varepsilon(x), \quad \widetilde{a} = a + \sigma Z.$

Model **(a)** can be read as a linear model. However, as opposed to the case of standard regression, the "residuals" are in this case correlated. It is indeed straightforward to check that

$$\mathrm{Corr}\big(Y(x), Y(x')\big) = \frac{1}{1+\nu} \quad \text{if } x \neq x'\,.$$

The same model looked through **(b)** may instead be naturally described as a linear model, with uncorrelated residuals, but with the peculiarity of having a random intercept. Of course, as we have stressed, these are two equivalent interpretations of the same model. Both of them, and especially **(b)**, may however point out a caveat. If we generate data from this model, and subsequently fit an emulator to the data, we may not be able to recover the "true" value of the intercept a. Using classical statistical wording, we refer to this as to an identifiability issue.

Data generated from (2.128) may be interpreted as having, equally likely, come from a continuous spectrum of different models. One of such models may be characterised by an intercept a whose corresponding regression line is far from the observed data, but the data points are interpreted as being heavily correlated: essentially, a realisation of model **(a)**, where ν is very small with respect to the random value taken by Z. On the other side of the spectrum, the same data could be interpreted as coming from a linear model with an intercept which fits the data well, hence with the regression line mostly going through the points, and with the local variation around the line being explained by lack of correlations between the points (essentially, the intercept of the model is close to the value of $\widetilde{a}$ in interpretation **(b)**).

We can therefore see how the two extreme cases $d \to 0$ and $d \to \infty$ may be potentially confused in practical situations. Much to the author's dismay, he now finds himself in the urgent need to complete the present work, and he cannot investigate the potential issue further, as the topic deserves. Nonetheless, we thought it worth mentioning the issue here, leaving it as a topic of further investigation.

3. Principal Component Analysis Adapted to a Spherical Setting

Abstract: In practical applications, the output of the simulator to be emulated is often multi-dimensional. Principal Component Analysis (PCA) is commonly used to reduce the original problem into a small number of one-dimensional problems. In this chapter, we point out one important issue that the use of PCA is likely to cause, specifically when climate models are emulated. Hence, building on PCA ideas, we propose an alternative approach, where elements of $\mathbb{R}^s$ are identified with real-valued maps defined on the sphere S^2. This naturally endows $\mathbb{R}^s$ with a Hilbert-space structure, whose (non-Euclidean) geometry is appropriate to the problem and can therefore be used to find relevant variance-maximising directions. In the last section, we provide the details to implement the procedure.

3.1. Motivation

The setting introduced in Chapter 2 allows us to build emulators of computer models which can be represented as a function

$$f : \mathcal{P} \to \mathbb{R}\,. \tag{3.1}$$

In practical applications, the output of the computer model is rarely one-dimensional. In the special case of climate models, which is of particular interest in this work, the simulator output corresponding to any input $\boldsymbol{x} \in \mathcal{P}$ is provided on a number s of grid cells in which the Earth's surface has been discretised. An example of such output, representing simulated annual average temperature, is provided in Figure 3.1 where $s = 73 \times 96 = 7,008$. In similar cases, the simulator can be represented as a map $\boldsymbol{f}$ of the following form:

$$\boldsymbol{f} : \mathcal{P} \to \mathbb{R}^s\,, \tag{3.2}$$

where each of the s coordinates of $\mathbb{R}^s$ is associated with the output at one grid cell. The common case of a simulator as in (3.2) is generally dealt with in one of two ways.

1. Independently, s one-dimensional emulators are built, one for each output grid cell.

2. A suitable subspace $V \subseteq \mathbb{R}^s$ and a basis $\mathcal{B}$ of V are identified, such that each model output can be approximated as an element of V. Hence, the coefficients of the outputs with respect to the basis $\mathcal{B}$ are emulated.

Within the climate literature, an example where choice 1 is adopted is represented by the work Lee et al. [2012], where $s = 8192$ independent emulators are built and validated. Such an approach may however be computationally expensive to carry out. Moreover, in this case, the intrinsic covariance structure between outputs corresponding to neighbour or close grid cells is (at least in principle) lost. Nonetheless, note that the approach may be appropriate if either s is small, or only the outputs at a few, sparse grid cells are of interest.

In the majority of cases, choice 2 is adopted (*e.g.*, Bonceur et al. [2015], Tran et al. [2016], Lord et al. [2017]). See also Higdon et al. [2008] and Chang and Guillas [2019]

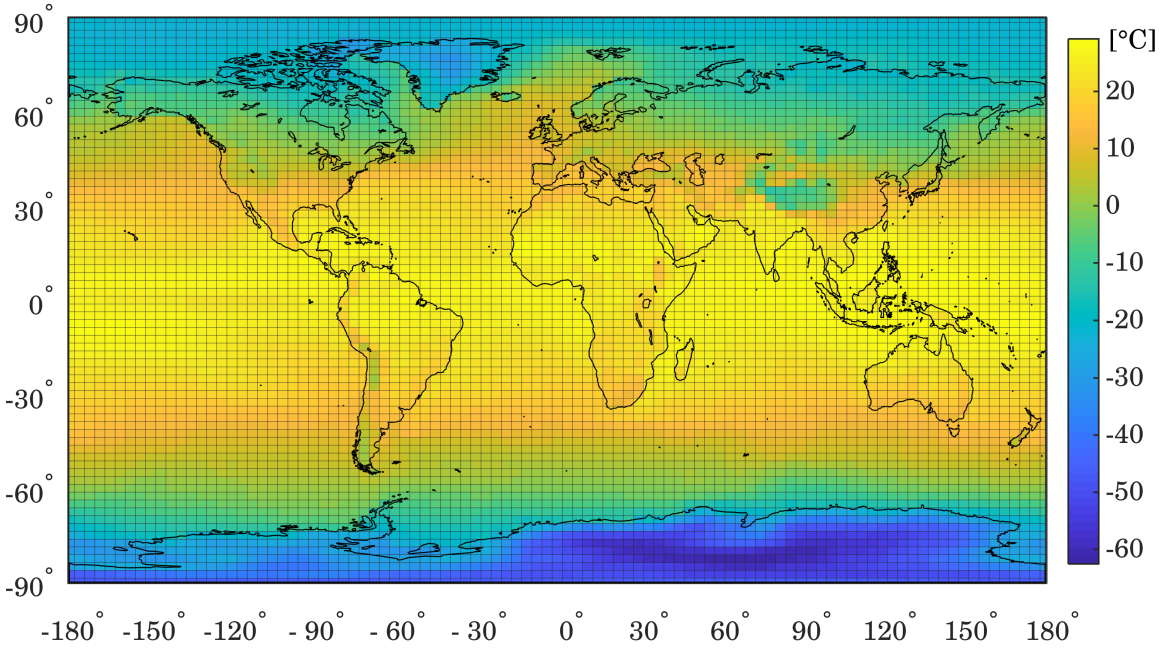


Figure 3.1: Structure of a typical climate simulator output. The Earth surface is discretised into a number of grid cells, and one output value is associated to each cell. In the illustration above: simulated annual average pre-industrial temperature, over a 73×96 grid; values from a HadCM3 simulation (more on this simulator in Chapter 4).

for examples of application within the context of calibration of computer models, where the multivariate outputs of the simulator are compared with physical observations. Since choice 2 requires the further effort of identifying an appropriate subspace $V \subseteq \mathbb{R}^s$ and a basis of this, it is clear that it becomes advantageous only in cases where the dimension s' of V is remarkably lower than the original dimension s. To the end, Principal Component Analysis (PCA) is often used. Given a set of m points $\boldsymbol{y_i} \in \mathbb{R}^s$, PCA considers the affine space that they span and identifies the directions which, sequentially, explain most-to-least of the variability of the data set $\mathcal{Y} = \{\boldsymbol{y_i}\}_{i=1,\dots,m}$.

In this chapter, we stress one issue that is prone to arise when PCA is performed on vectors $\boldsymbol{y_i}$ whose components represent values of a physical quantity over different grid cells of the Earth (as in Figure 3.1). Hence, we propose an alternative approach, which adapts the ideas underlying PCA to a non-Euclidean geometry. The case of the geometry of the sphere $S^2 \subseteq \mathbb{R}^3$ is of particular relevance in this work and is therefore stressed out, but the procedure we propose adapts to other cases.

Simulator grids on the Earth are specified in terms of latitude and longitude values,

often uniformly in each of the two directions. Figure 3.1 provides one such example. Such latitude-longitude uniform grids yield, however, highly non-uniform grids on the Earth surface: cells will be more densely concentrated at higher latitudes, where they account for smaller areas. Performing PCA under such a grid, therefore, identifies directions that automatically privilege the variability displayed by the data set at high latitudes, where most of the cells are located. To tackle the issue, in this chapter we propose an alternative approach. This is based on the following observation: although conveniently represented by a vector $\boldsymbol{y} \in \mathbb{R}^s$, a simulator output of the form discussed above is in reality a discretisation of a real map defined on the Earth. This can be represented as a function:

$$\varphi\colon S^2 \to \mathbb{R}\,, \tag{3.3}$$

where the sphere $S^2 \subseteq \mathbb{R}^3$ is used as mathematical model of the Earth. Starting from this idea, we carry out a dimension reduction that takes into account the spherical geometry of the problem and the aforementioned differences in cell areas.

Our procedure is illustrated in Section 3.3. Since this builds on the ideas of classical PCA, we provide a brief summary of the latter in Section 3.2, with the aim to set as well the basic notation.

3.2. Classical PCA: Review of Theory and Formulas

Principal Component Analysis is a classical topic of multivariate statistics, probably the most widely employed methodology for dimension reduction. Here we only provide a brief account of PCA: no proof is supplied, but the interpretation is stressed throughout. More details on PCA can be found in any undergraduate text on multivariate statistics, see for example Mardia et al. [1979, Chap. 8].

We start from m vectors $\boldsymbol{y_1}, \dots, \boldsymbol{y_m} \in \mathbb{R}^s$, which will be fixed throughout the section. Notice that, as points in $\mathbb{R}^s$, these span an affine space $\widetilde{V}$ of dimension, at most, $m-1$. We have used the tilde, since we denote with V the underlying vector space obtained by translating $\widetilde{V}$ to the origin:

$$\widetilde{V} \;=\; \overline{\boldsymbol{y}} + V \;=\; \big\{\, \overline{\boldsymbol{y}} + \boldsymbol{v} \,\big|\, \boldsymbol{v} \in V \,\big\}, \tag{3.4}$$

where

$$\overline{\boldsymbol{y}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{y_i} \in \mathbb{R}^s. \tag{3.5}$$

Given any direction $\boldsymbol{u} \in \mathbb{R}^s$, $\|\boldsymbol{u}\| = 1$, we can consider the projections of the elements $\boldsymbol{y_i}$ onto $\boldsymbol{u}$:

$$\langle \boldsymbol{y_1}, \boldsymbol{u}\rangle, \ldots, \langle \boldsymbol{y_m}, \boldsymbol{u}\rangle \in \mathbb{R}\,, \tag{3.6}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product on $\mathbb{R}^s$.

Observation (terminology)

Strictly speaking, the projection of a vector $\boldsymbol{y}$ onto the liner space generated by $\boldsymbol{u} \in \mathbb{R}^s$ is itself a vector, by definition proportional to $\boldsymbol{u}$: if $\|\boldsymbol{u}\| = 1$, the projection is $\langle \boldsymbol{y}, \boldsymbol{u}\rangle \boldsymbol{u}$. In this chapter, it is however convenient to identify the projection of $\boldsymbol{y}$ onto $\boldsymbol{u}$ directly with the scalar $\langle \boldsymbol{y}, \boldsymbol{u}\rangle$.

We can now can define the empirical variance of the m projections, and look at this as function of the direction $\boldsymbol{u} \in \mathbb{R}^s$:

$$G(\boldsymbol{u}) = \mathrm{Var}\Big\{ \langle \boldsymbol{y_1}, \boldsymbol{u}\rangle, \ldots, \langle \boldsymbol{y_m}, \boldsymbol{u}\rangle \Big\} \geq 0\,. \tag{3.7}$$

The quantity $G(\boldsymbol{u})$ is informative of the orientation of $\boldsymbol{u}$ with respect to the space $\widetilde{V}$. If $G(\boldsymbol{u}) = 0$, then $\boldsymbol{u}$ is orthogonal to $\widetilde{V}$ (and vice versa). Otherwise, the bigger $G(\boldsymbol{u})$, the more $\boldsymbol{u}$ represents a direction of particular variability of the data set $\mathcal{Y} = \big\{\boldsymbol{y_i}\big\}_i$.

PCA allows us to identify an orthonormal basis $\mathcal{B} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_{m-1}}\}$ of V, whose elements, sequentially, represent directions of maximal variability within the data set. More formally, the elements $\boldsymbol{v_j} \in \mathbb{R}^s$ satisfy the following:

$$\boldsymbol{v_1} = \underset{\boldsymbol{u}\in\mathbb{R}^s,\, \|\boldsymbol{u}\|=1}{\arg\max}\; G(\boldsymbol{u})\,, \tag{3.8}$$

and, recursively for $j = 2, \ldots, m-1$,

$$\boldsymbol{v_j} = \underset{\substack{\boldsymbol{u}\in\mathbb{R}^s,\, \|\boldsymbol{u}\|=1 \\ \boldsymbol{u}\perp\{\boldsymbol{v_1},\ldots,\boldsymbol{v_{j-1}}\}}}{\arg\max}\; G(\boldsymbol{u})\,. \tag{3.9}$$

The basis vectors $\boldsymbol{v_j}$ are called Principal Components (PCs). Notice that the maximisation takes place on $\mathbb{R}^s$, but will automatically select elements in V (assuming no singularity of the data set $\mathcal{Y}$, so that $\dim(V) = m - 1$).

Notation convention on indices

In this chapter, we make frequent reference to the elements of the original set $\mathcal{Y} = \{\boldsymbol{y_1}, \ldots, \boldsymbol{y_m}\} \subset \mathbb{R}^s$ and to the elements of the basis $\mathcal{B} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_{m-1}}\} \subset \mathbb{R}^s$, both introduced above. To notationally ease the distinction between these, we use:

- the index $i \in \{1, \ldots, m\}$ to refer to the elements $\boldsymbol{y_i} \in \mathbb{R}^s$ or to quantities associated with these;
- the index $j \in \{1, \ldots, m-1\}$ to refer to the principal components $\boldsymbol{v_j} \in \mathbb{R}^s$ or to quantities associated with these.

When needed, we will use the index $c \in \{1, \ldots, s\}$ to refer to the components of a generic vector $\boldsymbol{y} \in \mathbb{R}^s$ (each associated with a cell of the simulator Earth grid).

The theory of PCA ensures that the PCs are eigenvectors of the empirical covariance matrix $\boldsymbol{C}$ of the data set $\mathcal{Y}$; that is, of the $s \times s$ matrix

$$\boldsymbol{C} = \frac{1}{m-1} \boldsymbol{Y}^T \boldsymbol{Y} \in \mathbb{R}^{s \times s}, \tag{3.10}$$

where the i^{th} row of $\boldsymbol{Y} \in \mathbb{R}^{m \times s}$ is $(\boldsymbol{y_i} - \overline{\boldsymbol{y}})^T$. Moreover, if λ_j denotes the eigenvalue associated with $\boldsymbol{v_j}$, then it holds:

$$\lambda_j = G(\boldsymbol{v_j}) \geq 0\,. \tag{3.11}$$

Notice the following: the matrix $\boldsymbol{C}$ is real and symmetric, hence, by the classical spectral theorem of linear algebra, its eigenvectors form an orthogonal basis of $\mathbb{R}^s$. Moreover, the eigenvalues are non-negative, since $\boldsymbol{C}$ is positive semi-definite. Here, we are more specifically saying that the eigenvectors $\boldsymbol{v_j} \in \mathbb{R}^s$ associated with the first $m-1$ largest eigenvalues satisfy (3.8) and (3.9), and that the eigenvalues are the empirical variances associated to each $\boldsymbol{v_j}$ through the function $G(\cdot)$ in (3.7). The remaining eigenvalues are necessarily zero, since the rank of $\boldsymbol{C}$ is at most $m-1$ (and,

indeed, $G(\boldsymbol{u}) = 0$ if $\boldsymbol{u} \perp V$).

In practice, especially when $m \ll s$ (*i.e.*, number of points $\boldsymbol{y_i}$ much lower than dimension of the space to which they belong), it is computationally more stable and efficient to compute the PCs via the singular value decomposition (SVD) of $\boldsymbol{Y}$. This reads as follows:

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T \in \mathbb{R}^{m\times s}, \tag{3.12}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ is orthogonal ($\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I_m}$), $\boldsymbol{D} \in \mathbb{R}^{m\times m}$ is diagonal with non-negative elements d_i, and $\boldsymbol{V} \in \mathbb{R}^{s\times m}$ has orthonormal columns ($\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I_m}$).

It is straightforward to check that the columns of $\boldsymbol{V}$ are eigenvectors of $\boldsymbol{Y}^T\boldsymbol{Y}$, with associated eigenvalues d_j^2 (*i.e.*, $(\boldsymbol{Y}^T\boldsymbol{Y})\boldsymbol{V} = \boldsymbol{V}\boldsymbol{D}^2$). Hence, the columns of $\boldsymbol{V}$ are also eigenvectors of $\boldsymbol{C}$, with associated eigenvalues $d_j^2/(m-1)$. Finally, notice that the i^{th} row of (3.12), after transposing, reads as follows:

$$\boldsymbol{y_i} - \overline{\boldsymbol{y}} = \sum_{j=1}^{m} Q_{ij}\boldsymbol{v_j}, \qquad i = 1, \dots, m\,, \tag{3.13}$$

where $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{D}$, and $\boldsymbol{v_j}$ is the j^{th} column of $\boldsymbol{V}$. Hence, the matrix $\boldsymbol{Q}$ contains the coefficients of the linear combinations expressing the elements $\boldsymbol{y_i} \in \mathbb{R}^s$ in terms of the basis $\{\boldsymbol{v_j}\}$.

3.3. PCA on a Different Geometry

In this section we propose a variant of PCA, appropriate to geometries on $\mathbb{R}^s$ that are different to the Euclidean one. While still working on s-dimensional vectors, the idea behind our procedure is to interpret these as discretisation of infinite-dimensional objects, specifically real-valued maps defined on the sphere S^2 (or on a subset thereof).

Before proceeding, I[4] would like to make a note. Albeit in a different setting, formulas similar to the ones that our procedure recovers are discussed and used in Salter et al. [2019], which references the book Jolliffe [2002]. Upon consulting this, I discovered that mathematical ideas similar to the ones proposed below are

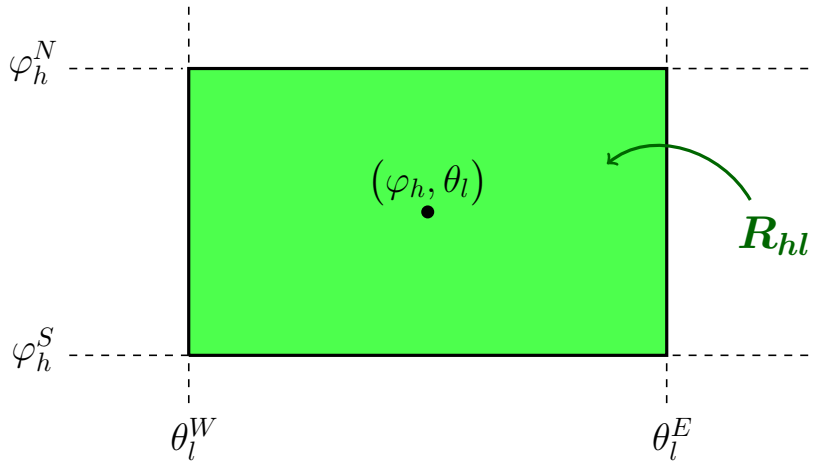
[4] The author of this work.

introduced in the last chapter of the book. The present chapter has nonetheless been independently developed by myself, and no result or proof has been adapted from the previous references, which I became aware of only after completing the present chapter. Moreover, such ideas have not yet been applied within an emulation framework, as we do in Part II of this work.

3.3.1. Immersing $\mathbb{R}^s$ Into a Space of Functions

As illustrated in Figure 3.1, a typical output of a climate simulator is provided over a grid of the Earth. We represent such an output as a vector $\boldsymbol{y} \in \mathbb{R}^s$, whose components are simulated values at the different grid cells. In the following, we assume for convenience a uniform grid, and denote by N_{Lat} and N_{Lon} the number of grids cells along the latitude and longitude dimensions, respectively. Moreover, let us denote by (φ_h, θ_l) the coordinates of the grid cell centres, for $h = 1, \ldots, N_{\text{Lat}}$ and $l = 1, \ldots, N_{\text{Lon}}$. Hence, each cell, R_{hl}, is a (spherical) rectangle which in latitude-longitude coordinates reads as follows:

$$R_{hl} = \left[\varphi_h^S, \varphi_h^N\right] \times \left[\theta_l^W, \theta_l^E\right]. \tag{3.14}$$



The S, N, W and E superscripts are meant to remind of the four cardinal points. In the case of a uniform grid, we have:

$$\varphi_h^S = \varphi_h - \frac{\delta}{2}, \quad \varphi_h^N = \varphi_h + \frac{\delta}{2}, \qquad h = 1, \ldots, N_{\text{Lat}} \tag{3.15.a}$$

$$\theta_l^W = \theta_l - \frac{\varepsilon}{2}, \qquad \theta_l^E = \theta_l + \frac{\varepsilon}{2}, \qquad l = 1, \ldots, N_{\text{Lon}}\,, \tag{3.15.b}$$

where $\delta = \pi/N_{\text{Lat}}$ and $\varepsilon = 2\pi/N_{\text{Lon}}$ are the latitude and longitude step sizes, respectively, both measured in radians.

Remark 3.1. According to the grid of the simulator in use, the definition of φ_h^N (respectively, φ_h^S) for the cells closest to the North (respectively, South) Pole may differ from the one in equation (3.15.a). For example, in the case of the HadCM3 simulator, employed in Chapter 4 and Chapter 5 of this work, the northernmost cells are characterised by a value of φ_h equal to $\pi/2$: for these cells, we have $\varphi_h^N = \pi/2$. This corresponds to a grid which has N_{Lon} "triangular" cells around the North Pole, each extending in longitude for $\delta/2$ radians. The case of the South Pole is symmetric.

A vector $\boldsymbol{y} \in \mathbb{R}^s$, with grid cell simulator outputs as components, can be naturally interpreted as a function from S^2 to $\mathbb{R}$: the function f_y which is constant on each cell, with value given by the appropriate component of $\boldsymbol{y}$. If we denote by $c(h,l) \in \{1, \ldots, s\}$ the index corresponding to the cell R_{hl}, we can write f_y as:

$$f_y(\boldsymbol{z}) = \sum_{h,l} y_{c(h,l)}\, \mathbb{1}_{R_{hl}}(\boldsymbol{z})\,, \quad \boldsymbol{z} \in S^2, \tag{3.16}$$

where $\mathbb{1}_{R_{hl}} \colon S^2 \to \mathbb{R}$ is the indicator function of cell R_{hl}. Any such f_y is an element of a much bigger space, which has a natural inner-product structure. This is the vector space of real, square-integrable functions on S^2:

$$\mathcal{H} = \left\{ f \colon S^2 \to \mathbb{R} \;\middle|\; \int_{S^2} f^2(\boldsymbol{z})\, d\boldsymbol{z} < \infty \right\}. \tag{3.17}$$

If endowed with the following inner product:

$$\langle f, g \rangle_{\mathcal{H}} = \int_{S^2} f(\boldsymbol{z})\, g(\boldsymbol{z})\, d\boldsymbol{z}\,, \quad f, g \in \mathcal{H}, \tag{3.18}$$

the set $\mathcal{H}$ becomes a Hilbert space (*i.e.*, it is complete, in the standard sense that every Cauchy sequence converges, under the metric induced by the inner product).

Equation (3.16) allows us to interpret vectors in $\mathbb{R}^s$ as elements of $\mathcal{H}$. This naturally induces a different inner product on $\mathbb{R}^s$ than the Euclidean one. Indeed, for any

$\boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^s$, the integral (3.18) of the associated functions $f_y, f_{y'} \in \mathcal{H}$ translates into the following finite sum:

$$\langle f_y, f_{y'} \rangle_{\mathcal{H}} = \int_{S^2} f_y(\boldsymbol{z})\, f_{y'}(\boldsymbol{z})\, d\boldsymbol{z} \;=\; \sum_{c=1}^{s} w_c\, y_c\, y'_c\,, \tag{3.19}$$

where the weight w_c is equal to the area of grid cell c. Seen as bilinear function of $\boldsymbol{y}$ and $\boldsymbol{y}' \in \mathbb{R}^s$, equation (3.19) defines an inner product on $\mathbb{R}^s$. We denote this by $\langle \cdot, \cdot \rangle_W$:

$$\langle \boldsymbol{y}, \boldsymbol{y}' \rangle_W = \sum_{c=1}^{s} y_c\, w_c\, y'_c = \boldsymbol{y}^T \boldsymbol{W} \boldsymbol{y}'\,, \quad \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^s\,, \tag{3.20}$$

where $\boldsymbol{W} \in \mathbb{R}^{s \times s}$ is the diagonal matrix with diagonal elements $w_c > 0$.

The reasoning carried out so far is valid for any grid. However, in the case of a rectangular grid, a simple formula can be obtained for the area of the cell R_{hl} defined in (3.14) and (3.15). We derive the formula below, via a surface integral. To the aim, we parameterise the sphere using latitude and longitude polar coordinates:

$$\begin{array}{rccc} \boldsymbol{\Phi}\colon & \left[-\dfrac{\pi}{2}, \dfrac{\pi}{2}\right] \times [0, 2\pi] & \longrightarrow & S^2 \\ & (\varphi, \theta) & \mapsto & \begin{pmatrix} \cos\varphi\, \cos\theta \\ \cos\varphi\, \sin\theta \\ \sin\varphi \end{pmatrix} \end{array} \tag{3.21}$$

Indeed, a point at latitude φ and longitude θ has z component equal to $\sin\varphi$, and distance from the z axis equal to $\cos\varphi$: from this last figure, it follows that the x and y components are the ones in (3.21). Locally, the area-scaling factor of the transformation is given by the norm of the cross product between the two tangent vectors:

$$\left\| \frac{\partial \boldsymbol{\Phi}}{\partial \varphi} \times \frac{\partial \boldsymbol{\Phi}}{\partial \theta} \right\| = \cos\varphi\,. \tag{3.22}$$

Hence, the area of the spherical rectangle R_{hl} is as follows:

$$\begin{aligned} \mathrm{Area}(R_{hl}) = \int_{R_{hl}} 1\, dA &= \int_{\varphi_h^S}^{\varphi_h^N} \int_{\theta_l^W}^{\theta_l^E} \cos\varphi\, d\varphi\, d\theta \\ &= (\theta_l^E - \theta_l^W)\, (\sin\varphi_h^N - \sin\varphi_h^S) \\ &= \varepsilon\, (\sin\varphi_h^N - \sin\varphi_h^S)\,, \end{aligned} \tag{3.23}$$

where $\varepsilon = 2\pi/N_{\text{Lat}}$ is the longitude step size previously introduced.

Notice that, as expected, $\text{Area}(R_{hl})$ does only depend on the latitude φ_h at which R_{hl} is placed, and not on the longitude θ_l: the area shrinks towards the poles ($\varphi_h \to \pm\pi/2$), where the sine approaches zero derivative. This mathematically supports what already stressed in Section 3.1, *i.e.*, that a polar region of a given geographical area consists of many more cells than an equatorial region of the same area.

3.3.2. Theoretical Formula for the Principal Components

In the previous Section, we have defined an inner product on $\mathbb{R}^s$,

$$\langle \boldsymbol{y}, \boldsymbol{y}' \rangle_{W} = \boldsymbol{y}^T \boldsymbol{W}\, \boldsymbol{y}\,, \quad \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^s, \tag{3.24}$$

where the matrix $\boldsymbol{W} \in \mathbb{R}^{s\times s}$ is diagonal with weights chosen to resemble the L^2 inner product of the Hilbert space $\mathcal{H}$ of square-integrable, real-valued maps on the sphere. We now go back to the problem introduced in Section 3.1: performing PCA on a data set $\mathcal{Y} = \{\boldsymbol{y_i}\}_{i=1,\dots,m} \subset \mathbb{R}^s$, which consists of vectors approximating maps on the sphere. To accomplish the aim, we follow the same ideas underlying PCA, but use the more natural inner product $\langle \cdot, \cdot \rangle_{W}$ in order to account properly for the variability displayed by the elements of the data set in the different geographical areas.

Remark 3.2. The validity of this section's results is not limited to the case where the matrix $\boldsymbol{W} \in \mathbb{R}^{s\times s}$ in equation (3.24) is chosen to resemble the L^2 inner product of real functions defined on S^2. The results remain valid for any diagonal $\boldsymbol{W}$ with positive diagonal entries and, in fact, for any symmetric and positive-definite matrix $\boldsymbol{W}$. The procedure we outline below is therefore of relevance to any dimension-reduction problem, where the meaning of the vectors on which dimension-reduction is performed suggests the use of an inner product different to the Euclidean one.

Let us therefore consider m starting points $\boldsymbol{y_1}, \dots, \boldsymbol{y_m} \in \mathbb{R}^s$. Our aim is to look for the directions of maximal variability that characterise the data set, with respect to the inner product (3.24). That is, we define, recursively:

$$\boldsymbol{v_1} = \underset{\boldsymbol{v}\in\mathbb{R}^s,\, \|\boldsymbol{v}\|_W = 1}{\arg\max}\ \text{Var}\Big\{ \langle \boldsymbol{y_1}, \boldsymbol{v} \rangle_{W}, \dots, \langle \boldsymbol{y_m}, \boldsymbol{v} \rangle_{W} \Big\}, \tag{3.25.a}$$

and

$$\boldsymbol{v_j} = \underset{\substack{\boldsymbol{v}\in\mathbb{R}^s,\, \|\boldsymbol{v}\|_W=1 \\ \boldsymbol{v}\perp_W\{\boldsymbol{v_1},\dots,\boldsymbol{v_{j-1}}\}}}{\arg\max} \operatorname{Var}\Big\{\langle \boldsymbol{y_1},\boldsymbol{v}\rangle_W,\dots,\langle \boldsymbol{y_m},\boldsymbol{v}\rangle_W\Big\}, \tag{3.25.b}$$

for $j = 2,\dots,m-1$. The symbol $\|\cdot\|_W$ denotes the norm induced by the inner product (3.24) ($\|\boldsymbol{v}\|_W^2 = \langle \boldsymbol{v},\boldsymbol{v}\rangle_W$ for $\boldsymbol{v}\in\mathbb{R}^s$); similarly, the symbol $\perp_W$ denotes orthogonality with respect to this inner product.

Theorem 3.3.1 shows that problem (3.25) can be reformulated as an eigenvector problem. This is analogous to the PCA case, but the problem is, in our case, asymmetric. The proof has been autonomously developed by the author of this work, and may therefore differ, also in the ideas and tools used, from proofs of the analogous result of classical PCA. Before stating Theorem 3.3.1, let us conveniently introduce the quantities $\overline{\boldsymbol{y}}$ and $\boldsymbol{Y}$, as follows:

$$\overline{\boldsymbol{y}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y_i}\in\mathbb{R}^s, \qquad \boldsymbol{Y} = \begin{pmatrix} (\boldsymbol{y_1}-\overline{\boldsymbol{y}})^T \\ \vdots \\ (\boldsymbol{y_m}-\overline{\boldsymbol{y}})^T \end{pmatrix} \in \mathbb{R}^{m\times s}. \tag{3.26}$$

To keep the notation compact, for any $\boldsymbol{v}\in\mathbb{R}^s$ let us also define:

$$G(\boldsymbol{v}) := \operatorname{Var}\Big\{\langle \boldsymbol{y_1},\boldsymbol{v}\rangle_W,\dots,\langle \boldsymbol{y_m},\boldsymbol{v}\rangle_W\Big\}. \tag{3.27}$$

Theorem 3.3.1. *Let $\boldsymbol{y_1},\dots,\boldsymbol{y_m}\in\mathbb{R}^s$ be m vectors, and define $\boldsymbol{Y}\in\mathbb{R}^{m\times s}$ as in (3.26). Moreover, let $\boldsymbol{W}\in\mathbb{R}^{s\times s}$ be a symmetric, positive definite matrix, so that the bilinear map*

$$\langle \boldsymbol{y},\boldsymbol{y}'\rangle_W = \boldsymbol{y}^T\boldsymbol{W}\boldsymbol{y}', \quad \boldsymbol{y},\boldsymbol{y}'\in\mathbb{R}^s, \tag{3.28}$$

defines an inner product on $\mathbb{R}^s$. Then, the matrix

$$\boldsymbol{C} = \frac{1}{m-1}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W} \ \in\mathbb{R}^{s\times s} \tag{3.29}$$

has real, non-negative eigenvalues $\lambda_1\geq\cdots\geq\lambda_s\geq 0$, and the corresponding eigenvectors $\boldsymbol{v_j}$ satisfy equations (3.25.a), (3.25.b). Moreover, for each $j=1,\dots,s$, it holds:

$$G(\boldsymbol{v_j}) = \lambda_j \geq 0\,. \tag{3.30}$$

Proof. We divide the proof into four small blocks.

1. Part 1: Show that $G(\boldsymbol{v}) \propto \|\boldsymbol{X}\boldsymbol{W}\boldsymbol{v}\|^2$ for any $\boldsymbol{v} \in \mathbb{R}^s$.
2. Part 2: Show that the eigenvectors $\boldsymbol{v_j}$ form a (special) basis of $\mathbb{R}^s$.
3. Part 3: For any $\boldsymbol{v} \in \mathbb{R}^s$, write $G(\boldsymbol{v})$ in terms of its basis coefficients.
4. Part 4: Maximise $G(\cdot)$ and show the claim.

Part 1: For any $\boldsymbol{v} \in \mathbb{R}^s$, we have the following:

$$\begin{aligned} G(\boldsymbol{v}) &= \mathrm{Var}\Big\{ \langle \boldsymbol{y_1}, \boldsymbol{v} \rangle_{\boldsymbol{W}}, \dots, \langle \boldsymbol{y_m}, \boldsymbol{v} \rangle_{\boldsymbol{W}} \Big\} \\ &= \mathrm{Var}\Big\{ \langle \boldsymbol{y_1} - \overline{\boldsymbol{y}}, \boldsymbol{v} \rangle_{\boldsymbol{W}}, \dots, \langle \boldsymbol{y_m} - \overline{\boldsymbol{y}}, \boldsymbol{v} \rangle_{\boldsymbol{W}} \Big\}, \end{aligned} \tag{3.31}$$

since the inner product is linear in the first argument, and $\langle \overline{\boldsymbol{y}}, \boldsymbol{v} \rangle_{\boldsymbol{W}}$ is a constant. Moreover, again by the inner product linearity and by the definition of $\overline{\boldsymbol{y}}$, we have:

$$\frac{1}{m} \sum_{i=1}^{m} \langle \boldsymbol{y_i} - \overline{\boldsymbol{y}}, \boldsymbol{v} \rangle_{\boldsymbol{W}} = 0 \,. \tag{3.32}$$

Hence, the variance in (3.31) becomes the sum of the square of each element, normalised. That is:

$$(m-1)\, G(\boldsymbol{v}) \;=\; \left\| \begin{pmatrix} (\boldsymbol{y_1} - \overline{\boldsymbol{y}})^T \boldsymbol{W} \boldsymbol{v} \\ \vdots \\ (\boldsymbol{y_m} - \overline{\boldsymbol{y}})^T \boldsymbol{W} \boldsymbol{v} \end{pmatrix} \right\|^2 \;=\; \|\boldsymbol{Y}\boldsymbol{W}\boldsymbol{v}\|^2 \,, \tag{3.33}$$

where $\|\cdot\|$ is the standard Euclidean norm on $\mathbb{R}^m$.

Part 2: This part shows that the eigenvectors $\boldsymbol{v_j}$, as defined in the statement, form a $\langle \cdot, \cdot \rangle_{\boldsymbol{W}}$-orthonormal basis of $\mathbb{R}^s$. To the aim, we use the general version of the spectral theorem which is recalled in Appendix, Theorem B.2. To apply the theorem, we need to show that the matrix $\boldsymbol{C}$ is *symmetric with respect to the inner product* $\langle \cdot, \cdot \rangle_{\boldsymbol{W}}$. That is, we need to show that the following holds:

$$\langle \boldsymbol{C}\boldsymbol{y}, \boldsymbol{y}' \rangle_{\boldsymbol{W}} = \langle \boldsymbol{y}, \boldsymbol{C}\boldsymbol{y}' \rangle_{\boldsymbol{W}} \quad \forall\, \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^s. \tag{3.34}$$

This is a simple check, by using that $\boldsymbol{C} = (\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W})/(m-1)$:

$$\begin{aligned}\langle \boldsymbol{C}\boldsymbol{y}, \boldsymbol{y}' \rangle_{\boldsymbol{W}} &= (\boldsymbol{y}^T\boldsymbol{C}^T)\boldsymbol{W}\boldsymbol{y}' \\ &= \frac{1}{m-1}\,\boldsymbol{y}^T\boldsymbol{W}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W}\boldsymbol{y}' \\ &= \boldsymbol{y}^T\boldsymbol{W}\boldsymbol{C}\boldsymbol{y}' = \langle \boldsymbol{y}, \boldsymbol{C}\boldsymbol{y}' \rangle_{\boldsymbol{W}}\,. \end{aligned} \tag{3.35}$$

Hence, by the spectral theorem, the[5] set $\mathcal{B} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_s}\}$ of eigenvectors of $\boldsymbol{C}$ forms a basis of $\mathbb{R}^s$, orthonormal with respect to $\langle \cdot, \cdot \rangle_{\boldsymbol{W}}$. We can compactly write this using matrix notation. Denoting by $\boldsymbol{V} \in \mathbb{R}^{s\times s}$ the matrix whose j^{th} column is $\boldsymbol{v_j}$, by $\boldsymbol{\Lambda} \in \mathbb{R}^{s\times s}$ the diagonal matrix with diagonal elements the eigenvalues λ_j of $\boldsymbol{C}$, and by $\boldsymbol{I_s}$ the identity matrix of order s, we have:

$$\boldsymbol{C}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda} \in \mathbb{R}^{s\times s}\,, \tag{3.36.a}$$

$$\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{V} = \boldsymbol{I_s} \in \mathbb{R}^{s\times s}\,. \tag{3.36.b}$$

Equation (3.36.a) asserts that the columns of $\boldsymbol{V}$ are eigenvectors of $\boldsymbol{C}$, with associated eigenvalues λ_j. Equation (3.36.b) states that the columns of $\boldsymbol{V}$ are orthonormal with respect to (3.28).

Part 3: Since $\mathcal{B}$ is a basis of $\mathbb{R}^s$, any vector $\boldsymbol{v}$ can be written as linear combination of the $\boldsymbol{v_j}$. That is, for any $\boldsymbol{v} \in V$, we can write:

$$\boldsymbol{v} = \boldsymbol{V}\boldsymbol{\alpha}, \tag{3.37}$$

for some vector of coefficients $\boldsymbol{\alpha} \in \mathbb{R}^s$. We can then compute the variance $G(\boldsymbol{v})$ in terms of the coefficients $\boldsymbol{\alpha}$. Starting from (3.33), we have:

$$\begin{aligned} G(\boldsymbol{v}) = &\quad \frac{1}{m-1}\,(\boldsymbol{v}^T\boldsymbol{W}\boldsymbol{Y}^T)(\boldsymbol{Y}\boldsymbol{W}\boldsymbol{v}) \\ \overset{(3.37)}{=} &\quad \frac{1}{m-1}\,\boldsymbol{\alpha}^T\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W}\boldsymbol{V}\boldsymbol{\alpha} \\ = &\quad \boldsymbol{\alpha}^T\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{C}\boldsymbol{V}\boldsymbol{\alpha} \end{aligned}$$

[5] Notice that, if the multiplicity of an eigenvalue is greater than one (if $m < s$, $\lambda = 0$ is one such eigenvalue), we *choose* the corresponding eigenvectors so that they are orthogonal to each other, among the infinite choices available.

$$\begin{aligned} &\overset{(3.36.a)}{=} \boldsymbol{\alpha}^T \boldsymbol{V}^T \boldsymbol{W} \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{\alpha} \\ &\overset{(3.36.b)}{=} \boldsymbol{\alpha}^T \boldsymbol{\Lambda} \boldsymbol{\alpha} \\ &= \sum_{j=1}^{s} {\alpha_j}^2 \lambda_j \,. \end{aligned} \tag{3.38}$$

Notice, in particular, that equation (3.38) immediately shows that the eigenvalues λ_j are all non-negative (choose $\alpha_j = 1$ and all other α_k equal to zero). If needed, we reorder both them and the corresponding eigenvectors, so that $\lambda_1 \geq \cdots \geq \lambda_s \geq 0$.

Part 4: We can finally maximise $G(\cdot)$, over the subspaces of interest. From $\boldsymbol{v} = \boldsymbol{V}\boldsymbol{\alpha}$, and by exploiting (3.36.b), it is immediate to see that

$$\|v\|_{\boldsymbol{W}} = 1 \iff \sum_{j=1}^{s} {\alpha_j}^2 = 1 \,. \tag{3.39}$$

Hence, equation (3.38) in particular says the following: for any $\|\cdot\|_{\boldsymbol{W}}$-unit vector $\boldsymbol{v} \in \mathbb{R}^s$, the value $G(\boldsymbol{v})$ is a convex combination of the positive eigenvalues λ_j of $\boldsymbol{C}$. Given the order of the λ_j, it immediately follows that

$$\max_{\|\boldsymbol{v}\|_{\boldsymbol{W}}=1} G(\boldsymbol{v}) = \lambda_1, \tag{3.40}$$

and that the maximum of (3.38) is attained at $\boldsymbol{\alpha} = (1, 0, \ldots, 0)$. In terms of vectors $\boldsymbol{v}$, such $\boldsymbol{\alpha}$ yields $\boldsymbol{V}\boldsymbol{\alpha} = \boldsymbol{v_1}$. Therefore, we have proved that:

$$\boldsymbol{v_1} = \operatorname*{arg\,max}_{v \in \mathbb{R}^s,\, \|\boldsymbol{v}\|_{\boldsymbol{W}}=1} G(\boldsymbol{v}) \quad \text{and} \quad G\big(\boldsymbol{v_1}\big) = \lambda_1 \,. \tag{3.41}$$

We can now maximise $G(\cdot)$ in the subspace orthogonal to $\boldsymbol{v_1}$. If we suppose that $\boldsymbol{v} = \boldsymbol{V}\boldsymbol{\alpha}$ satisfies $\|\boldsymbol{v}\|_{\boldsymbol{W}} = 1$ and that $\langle \boldsymbol{v}, \boldsymbol{v_1} \rangle_{\boldsymbol{W}} = 0$, then we get:

$$\alpha_1 = 0 \quad \text{and} \quad \sum_{j=2}^{s} {\alpha_j}^2 = 1 \,. \tag{3.42}$$

Among all such $\boldsymbol{v}$, we have

$$G(\boldsymbol{v}) = \sum_{j=2}^{s} {\alpha_j}^2 \lambda_j \,. \tag{3.43}$$

Hence, the same reasoning of before shows the following:

$$\boldsymbol{v_2} = \underset{\substack{v\in\mathbb{R}^s,\, \|\boldsymbol{v}\|_W=1 \\ \langle \boldsymbol{v},\boldsymbol{v_1}\rangle_W}}{\arg\max} G(\boldsymbol{v}) \quad \text{and} \quad G\big(\boldsymbol{v_2}\big) = \lambda_2\,. \tag{3.44}$$

The reasoning can be replicated for all unit vectors orthogonal to $\big\{\boldsymbol{v_1},\dots,\boldsymbol{v_{j-1}}\big\}$, and this completes the proof.

□

Theorem 3.3.1 provides an explicit solution to the constrained optimisation problem in (3.25). Notice that the result only requires $\boldsymbol{W}\in\mathbb{R}^{s\times s}$ to be symmetric and positive-definite, in order to define a proper inner product; it does not require $\boldsymbol{W}$ to be diagonal as in the setting of Subsection 3.3.1. Of course, if $\boldsymbol{W}=\boldsymbol{I_s}$, then the matrix $\boldsymbol{C}$ is simply the covariance matrix of the data set $\{\boldsymbol{y_i}\}$ and the results of classical PCA are recovered.

We conclude with a remark. It is probably intuitive, also in analogy with what stated in Section 3.2, that each vector $\boldsymbol{y_i}-\overline{\boldsymbol{y}}$ belongs to the span of the first $m-1$ principal components only. This could be proven easily[6], but also follows from what we show in Subsection 3.3.3. Hence we skip an independent, redundant proof. The main geometrical idea should nonetheless remain, and we summarise it in the following box.

Geometrical Overview

The m centred vectors $\boldsymbol{y_i}-\overline{\boldsymbol{y}}\in\mathbb{R}^s$ span a linear space V of dimension at most $m-1$: to be precise, of dimension $r=\operatorname{rank}(\boldsymbol{C})$. Exactly the first r eigenvectors of $\boldsymbol{C}$ form a $\|\cdot\|_W$-orthogonal basis of V, explaining sequentially most-to-least of the data set variance, and have eigenvalues equal to these variances. The remaining eigenvectors are orthogonal to any of the data set elements, and are indeed associated with a zero eigenvalue.

[6] Main idea: i) observe that $\operatorname{rank}(\boldsymbol{C})\le m-1$, hence $\lambda_m=\dots=\lambda_s=0$; ii) relate these λ_j, through (3.30), to the coefficients $\langle \boldsymbol{y_i}-\overline{\boldsymbol{y}},\boldsymbol{v_j}\rangle_W$ of the expansion of $\boldsymbol{y_i}-\overline{\boldsymbol{y}}$ with respect to the basis $\{\boldsymbol{v_j}\}$.

3.3.3. Computing the Principal Components

In practical applications, the square matrix $\boldsymbol{C}$ in (3.29) is of order s of several thousands: in Chapter 4 and Chapter 5, we deal both times with settings where s is of the order of 4×10^4 (Table 4.1 and Table 5.2). In such cases, storing in double precision the approximately 10^9 matrix elements already requires a notable amount of computer memory; a straight computation of all its eigenvectors becomes computationally unaffordable. Since the matrix is asymmetric, it is not directly possible to apply the SVD to its "square root": a square root, in the sense of a matrix $\boldsymbol{X}$ such that $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{X}^T$, clearly does not exist, since such a $\boldsymbol{C}$ would be symmetric. However, we can apply the SVD to a linear transformation of the original data $\mathcal{Y} = \{\boldsymbol{y_i}\}_{i=1,\dots,m}$ and then transform back, as explained below.

Supposing for convenience that the matrix $\boldsymbol{W}$ is diagonal, as the case of actual interest is (Subsection 3.3.1), the procedure to compute the eigenvectors of $\boldsymbol{C} = (\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W})/(m-1) \in \mathbb{R}^{s\times s}$ is as follows:

1. Define $\widetilde{\boldsymbol{Y}} = \boldsymbol{Y}\boldsymbol{W}^{1/2} \in \mathbb{R}^{m\times s}$.
2. Apply the usual SVD to $\widetilde{\boldsymbol{Y}}$: $\widetilde{\boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{D}\widetilde{\boldsymbol{V}}^T$.
3. Define $\boldsymbol{V} = \boldsymbol{W}^{-1/2}\widetilde{\boldsymbol{V}} \in \mathbb{R}^{s\times m}$.

We show below that this yields the eigenvectors of $\boldsymbol{C}$ as columns of $\boldsymbol{V}$, and the corresponding eigenvalues (up to a square and a scaling factor) as diagonal elements of the matrix $\boldsymbol{D}$ in point 2.

The SVD applied to $\widetilde{\boldsymbol{Y}} = \boldsymbol{Y}\boldsymbol{W}^{1/2}$ reads as follow:

$$\widetilde{\boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{D}\widetilde{\boldsymbol{V}}^T \in \mathbb{R}^{m\times s}, \tag{3.45}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ is orthogonal, $\boldsymbol{D} = (d_j)_{jj} \in \mathbb{R}^{m\times m}$ is diagonal, and $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{s\times m}$ satisfies $\widetilde{\boldsymbol{V}}^T\widetilde{\boldsymbol{V}} = \boldsymbol{I_m}$. First, let us observe that the m columns of $\widetilde{\boldsymbol{V}}$ are eigenvectors of the matrix $\boldsymbol{Q} = \boldsymbol{W}^{1/2}\boldsymbol{C}\boldsymbol{W}^{-1/2} \in \mathbb{R}^{s\times s}$:

$$(m-1)\,\boldsymbol{Q}\widetilde{\boldsymbol{V}} \;=\; (n-1)\,\boldsymbol{W}^{1/2}\boldsymbol{C}\boldsymbol{W}^{-1/2}\widetilde{\boldsymbol{V}}$$

$$\begin{aligned} &\overset{(\text{def of } C)}{=} \boldsymbol{W}^{1/2}\,\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W}^{1/2}\,\widetilde{\boldsymbol{V}} \\ &\overset{(\text{def of } \widetilde{Y})}{=} \widetilde{\boldsymbol{Y}}^T\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{V}} \\ &\overset{(3.45)}{=} \widetilde{\boldsymbol{V}}\boldsymbol{D}\boldsymbol{I_m}\boldsymbol{D}\widetilde{\boldsymbol{V}}^T\widetilde{\boldsymbol{V}} \\ &= \widetilde{\boldsymbol{V}}\boldsymbol{D}^2\,. \end{aligned} \tag{3.46}$$

This equivalently reads:

$$\boldsymbol{Q}\widetilde{\boldsymbol{V}} = \widetilde{\boldsymbol{V}}\left(\boldsymbol{D}^2/(m-1)\right), \tag{3.47}$$

i.e., the columns of $\widetilde{\boldsymbol{V}}$ are eigenvectors of $\boldsymbol{Q}$, with associated eigenvalues $\lambda_j = d_j^2/(m-1)$. Let us denote with $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ the diagonal matrix with diagonal elements λ_j.

Defining $\boldsymbol{V} = \boldsymbol{W}^{-1/2}\widetilde{\boldsymbol{V}}$, we want to show that its columns are eigenvectors of $\boldsymbol{C}$. From (3.47) and from the above definition of $\boldsymbol{Q}$, we see the following:

$$\begin{aligned} \boldsymbol{C}\boldsymbol{V} &= \boldsymbol{W}^{-1/2}\boldsymbol{Q}\boldsymbol{W}^{1/2}\boldsymbol{V} \\ &= \boldsymbol{W}^{-1/2}\boldsymbol{Q}\widetilde{\boldsymbol{V}} \\ &\overset{(3.47)}{=} \boldsymbol{W}^{-1/2}\widetilde{\boldsymbol{V}}\boldsymbol{\Lambda} \qquad (3.48) \\ &= \boldsymbol{V}\boldsymbol{\Lambda}\,. \qquad (3.49) \end{aligned}$$

Hence, as it was our aim, we have shown that the columns of $\boldsymbol{V}$ are eigenvectors of the matrix $\boldsymbol{C} = (\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W})/(m-1)$, with corresponding eigenvalues $\lambda_j = d_j^2/(m-1)$. For convenience and reference, we summarise the result below.

Proposition 3.3.2. *Given $\boldsymbol{Y} \in \mathbb{R}^{m\times s}$ and given $\boldsymbol{W} \in \mathbb{R}^{s\times s}$ diagonal with positive elements, define $\boldsymbol{V} \in \mathbb{R}^{s\times m}$ as per steps 1–3 above (beginning of Section). Then, the m columns of $\boldsymbol{V}$ are eigenvectors of the matrix*

$$\boldsymbol{C} = \frac{1}{m-1}\,\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{W} \in \mathbb{R}^{s\times s}.$$

The associated eigenvalues are $\lambda_j = d_j^2/(m-1)$, where d_j is the diagonal element of $\boldsymbol{D} \in \mathbb{R}^{m\times m}$, as in point 2 and equation (3.45).

Notice that, from the orthonormality of the columns of $\widetilde{\boldsymbol{V}}$ with respect to the Euclidean inner product ($\widetilde{\boldsymbol{V}}^T\widetilde{\boldsymbol{V}} = \boldsymbol{I_m}$), it follows that the columns of $\boldsymbol{V} = \boldsymbol{W}^{-1/2}\widetilde{\boldsymbol{V}}$ are

orthonormal with respect to the inner product $\langle \cdot, \cdot \rangle_{\boldsymbol{W}}$, as expected:

$$\boldsymbol{V}^T \boldsymbol{W} \boldsymbol{V} = (\widetilde{\boldsymbol{V}}^T \boldsymbol{W}^{-1/2}) \boldsymbol{W} (\boldsymbol{W}^{-1/2} \widetilde{\boldsymbol{V}}) = \widetilde{\boldsymbol{V}}^T \widetilde{\boldsymbol{V}} = \boldsymbol{I_m} \,. \tag{3.50}$$

Finally, from the SVD in (3.45), we can write the original elements of the data set as linear combinations of the PCs $\boldsymbol{v_j}$, columns of $\boldsymbol{V}$. By multiplying both sides of (3.45) on the right by $\boldsymbol{W}^{-1/2}$, we get:

$$\boldsymbol{Y} = \boldsymbol{U} \boldsymbol{D} \boldsymbol{V}^T \in \mathbb{R}^{m \times s}. \tag{3.51}$$

Notice that this is not a singular value decomposition, since $\boldsymbol{V}^T \boldsymbol{V} \neq \boldsymbol{I_m}$. The transpose of the i^{th} row of (3.51) then shows what we have claimed:

$$\boldsymbol{y_i} - \overline{\boldsymbol{y}} = \sum_{j=1}^{m} (\boldsymbol{U}\boldsymbol{D})_{ij} \boldsymbol{v_j} = \sum_{j=1}^{m} q_{ij} \boldsymbol{v_j}, \qquad q_{ij} = u_{ij}\, d_j \,. \tag{3.52}$$

That is, the elements $\boldsymbol{y_i} - \overline{\boldsymbol{y}}$ belong to the linear space generated by the first n PCs. In reality, the first $m-1$ PCs already span the space where each $\boldsymbol{y_i} - \overline{\boldsymbol{y}}$ lies. This follows from the observation that the rank of $\boldsymbol{Y}$, and therefore of $\widetilde{\boldsymbol{Y}}$, is at most $m-1$, since the m rows of $\boldsymbol{Y}$ are trivially dependent (they sum up to zero). The last singular value of $\boldsymbol{D}$ (equation (3.45)) is therefore $d_m = 0$. Hence, for any $i = 1, \ldots, m$, the coefficient q_{im} in (3.52) is zero. This also shows the claim made just before the start of Subsection 3.3.3.

An example of code implementing PCA, either in its classical form or in our variant, is provided in the Matlab Appendix D.2. The code returns the PCs, the matrix of coefficients $\boldsymbol{Q} = (q_{ij}) \in \mathbb{R}^{m \times (m-1)}$ and the standard deviations $\sqrt{\lambda_j}$ associated with each PC (see Proposition 3.3.2).

Part II

Applications to Past Climate Reconstruction

4. The Role of Orbital Variability in Ocean Temperature Reconstruction

Abstract: To understand the dynamics and consequences of current climate change, systematic scientific efforts are being undertaken to reconstruct the climate of past warm periods. To this aim, both complex climate models and geological records are employed. In this chapter, we employ Gaussian process emulation to investigate the mismatch between: *a*) simulated mid-Pliocene (∼3 million years ago) ocean temperatures; *b*) proxy reconstructions from geological records. The comparison takes full advantage of the nature of the emulator as stochastic process, by drawing sample trajectories from its distribution to resemble the way geological proxies are obtained. This way, we are able to account for the significant changes in ocean temperature induced by the varying orbital forcing. We also compare our results to the case where no emulator is employed.

4.1. Learn From the Past to Understand the Future

4.1.1. Motivation for the Interest in Mid-Pliocene Climate

Different scientific studies in recent years have ascertained that the current Earth's climate is undergoing a radical change: notable is in this regard the series of works by the Intergovernmental Panel on Climate Change (IPCC), among which IPCC [2007], IPCC [2013]. To face the changes, on the 12$^{\text{th}}$ of December 2015 a number of nations signed in Paris the famous Agreement, committing to:

> *[...] holding the increase in the global average temperature to well below 2°C above pre-industrial levels, and pursuing efforts to limit the temperature increase to 1.5°C above pre-industrial levels.*

These lines can be found in United Nations [2015], page 2. While exact predictions are impossible, the seemingly-small half degree of difference between the two scenarios is likely to yield very different consequences in terms of the number of days of extreme heat, severity of species loss, decline in coral reefs, sea-level rise: see for example Jahn [2018], Zhang et al. [2018].

Although these drastic changes are undoubtedly caused by the human activity, our planet has naturally experienced warmer-than-today periods during its lifetime. Understanding the nature of past warm climates, and the associated response of the different components of the Earth system, provides us with an excellent ground to gain insight on future changes. The mid-Pliocene, from around 3.3 to 3 million years ago, represents in this regard an ideal case study.

At the time, temperatures were warmer than during Pre-Industrial times (PI; around 1750–1800) and atmospheric CO_2 concentrations were much higher than then, see Dowsett et al. [1996]. Since the PI, atmospheric CO_2 concentrations have risen anomalously, from around 280 parts per million (ppm) to more than 400 ppm today. Prior to this present peak, the last time in history characterised by so high concentrations is the mid-Pliocene. The mid-Pliocene climate is therefore often regarded as the most similar analogue of the climate we are experiencing in this first half of

the twenty-first century. For this reasons, the climate community has endeavoured to gain a deeper understanding of that climate and of its effects[7], to as well enable more informed policy decision in tackling the current climate crisis.

4.1.2. The Combined Use of Models and Geological Data

Information from both geological records and climate simulations has been used to study the mid-Pliocene. On the geological side, the US Geological Survey launched in the late 1980's the PRISM (Pliocene Research, Interpretation, and Synoptic Mapping) project, Dowsett et al. [1994]. Its aim was to use marine and terrestrial records to investigate the magnitude and variability of the mid-Pliocene climate, with particular emphasis on North Atlantic marine records. Future generations of PRISM reconstructions, up to PRISM4 (Dowsett et al. [2016]), have extended the focus to the northern hemisphere, and later to the whole globe. On the climate modelling side, the Pliocene Model Intercomparison Project (PlioMIP, Haywood et al. [2011]) has coordinated the planning and execution of a number of different simulations, run with the most recent boundary conditions (vegetation, ice sheets) provided by the PRISM data sets. In 2016, its second phase was launched (PlioMIP2, Haywood et al. [2016a]).

While undoubtedly crucial to study the past climate, it must be acknowledged that climate simulators are not ideal, error-free tools. The climate system is the result of a number of complex interacting processes which take place in different components: atmosphere, oceans, ice sheets, vegetation. Studying in a systematic way the ability of a climate simulator to reproduce past warm climates becomes therefore a very important, but extremely challenging task. In recent years, information from models and geological archives has been combined to tackle this task. Alongside helping to achieve more reliable climate reconstructions (Chandler et al. [2008]), this step has been crucial to identify deficiencies and geographical biases of simulators: in Salzmann et al. [2013], for example, the authors suggest that most climate models are inclined to display a cold bias at high latitudes in reproducing warm past climate.

[7] Another period of great interest to the climate community, as means of comparison to the near future, is the Last Interglacial (around 125 thousand years ago). At the time, temperatures were at least comparable to the current levels, although CO_2 concentrations were much lower. We will expand on this in Chapter 5.

4.1.3. The Role of Statistics

One area in which the mid-Pliocene has been object of particular investigation is the one of data-model comparison (DMC; Lunt et al. [2010], Dowsett et al. [2013], Salzmann et al. [2013], Haywood et al. [2016b]). Nonetheless, the field still stands as one of the most challenging within the area of past climate reconstruction: not only from the climate point of view, but also from the statistical one. Remarkable challenges come from the uncertainty affecting the chronology of geological records, the cost of climate simulations (in terms of both time and computational power), as well as the intrinsic discrepancy between model predictions and reality. Moreover, as demonstrated in Prescott et al. [2014], the mid-Pliocene, covering approximately 300 thousand years, cannot be regarded as a period of stable climate conditions.

The aforementioned paper suggest that significant changes took place in the average annual temperature, as a consequence of the varying orbital forcing affecting the amount of solar radiation received by our planet (more on this in Subsection 4.4.1); in addition, the work argues that warm peaks during the mid-Pliocene were reached at different times in different locations. In light of these results, comparing geological archives to the outputs of one or few snapshot simulations may not allow to capture the highlighted orbitally-induced variability and the asynchronous warming behaviour.

Within this context, it becomes clear that a relevant contribution may be provided by the setting of GP emulation. A well-calibrated emulator allows to reliably predict the simulator outputs at different past times in fractions of a second, identify an underlying regular pattern in the simulator dynamics and encode uncertainty information on the predictions. For these reasons, the setting of GP emulation has been applied to various past climate reconstruction problems (Lee et al. [2011], Bonceur et al. [2015], Lord et al. [2017]). However, to the best of the author's knowledge, the field of mid-Pliocene DMC has never benefited from the GP emulation contribution.

4.1.4. Contribution of This Chapter

In the present chapter, we develop and use GP emulation techniques to provide a novel contribution to the framework of DMC during the mid-Pliocene. We emulate the sea

surface temperature (SST) output field of the HadCM3 climate model, as a function of those orbital parameters identified in Prescott et al. [2014] to be a significant source of temperature variability. We compare, at a number of marine sites, the emulator SST predictions to temperature reconstructions derived by ocean sediments and identify geographical patterns of data-model (mis)match. The use of GP emulation allows us to account for orbital forcing in explaining temperature variability and for potential asynchronicity between sites. Moreover, we analyse the results obtained by comparing the geological archive to single control simulations, run with fixed orbital forcing, to assess the role played by orbital variability in explaining the data-model mismatch.

More in detail, we proceed as follows. In Section 4.2 we present the geological records we use. In Section 4.3 and Section 4.4 we describe the structure, inputs and outputs of the simulator we employ, alongside an illustration of the relevant astronomical phenomena to which the inputs are linked. We then provide details of the construction of the emulators used to compare records and simulations. We describe the design in Section 4.5; apply the procedure illustrated in Chapter 3 to our case and further reduce the dimensionality in Section 4.6; illustrate and justify our prior emulator choices in Section 4.7; estimate the values of emulator hyperparameters in Section 4.8. In Section 4.9 we incorporate the uncertainty from left-out PCs and illustrate how to generate emulated trajectories at any location. Finally, we carry out the DMC in Section 4.10, illustrate our results and compare these to the ones obtained without emulators in Section 4.11, and conclude the chapter in Section 4.12.

4.2. Description of Marine Geological Archive

In this section we describe the archive of reconstructed mid-Pliocene ocean temperatures, to which the emulator predictions will be later compared. The archive we use comes from the PRISM3D data set (Dowsett et al. [2010]): for 51 marine sites, shown in Figure 4.1, it provides an estimate of warm peaks reached by the SST during the mid-Pliocene. The procedure followed to extract the information summarised in the data set can be schematically described by the following three steps[8]. For each site:

[8] Expert insight on this has been provided to the author by Prof. Harry Dowsett, personally involved in most of the data-collection process. Awareness of the procedure is at the basis of the way we

1. A time series of estimated SSTs is extracted from marine paleontological records (ocean sediments). The estimates are mostly fauna-based and correspond to mid-Pliocene times within the PRISM3D time slab, between 3,264 and 3,025 kya[9].

2. Warm peaks are identified within the series: these are defined as estimated temperatures which are preceded and followed by lower estimates.

3. For each location, the empirical mean of the subset of warm peaks is reported. The number N_p of peaks constituting the subset and the number N_s of samples of the original time series are also recorded in the data set.

We use the acronym WPA (Warm Peak Average) to refer to the empirical mean computed in step 3. We note that the times corresponding to each element of the time series are not easily inferred from the geological data, and in particular not provided in the dataset. The chronological order of the elements is however known, and clearly fundamental in the procedure of extracting warm peaks defined in step 2.

We now describe the simulator employed to reproduce the mid-Pliocene climate, and the process to construct a statistical emulator of this. We will come back to the data in Section 4.10, when we compare the emulator predictions to the geological data described above.

4.3. The Climate Simulator and its Output Field

The climate simulator used in this work to study the change in the Earth SST during the mid-Pliocene is the Hadley Centre Coupled Model, version 3 (HadCM3, Gordon et al. [2000]). The model was developed at the UK Met Office in 1999 and was extensively used in the Third and Fourth Assessments of the IPCC, in 2001 and 2007. It is a coupled atmosphere-ocean general circulation model.

The term general circulation model (GCM) refers to a numerical model describing the evolution of the main physical processes which develop in the atmosphere, ocean and

carry out the data-model comparison, in Section 4.10.

[9] kya: Thousand of years ago.

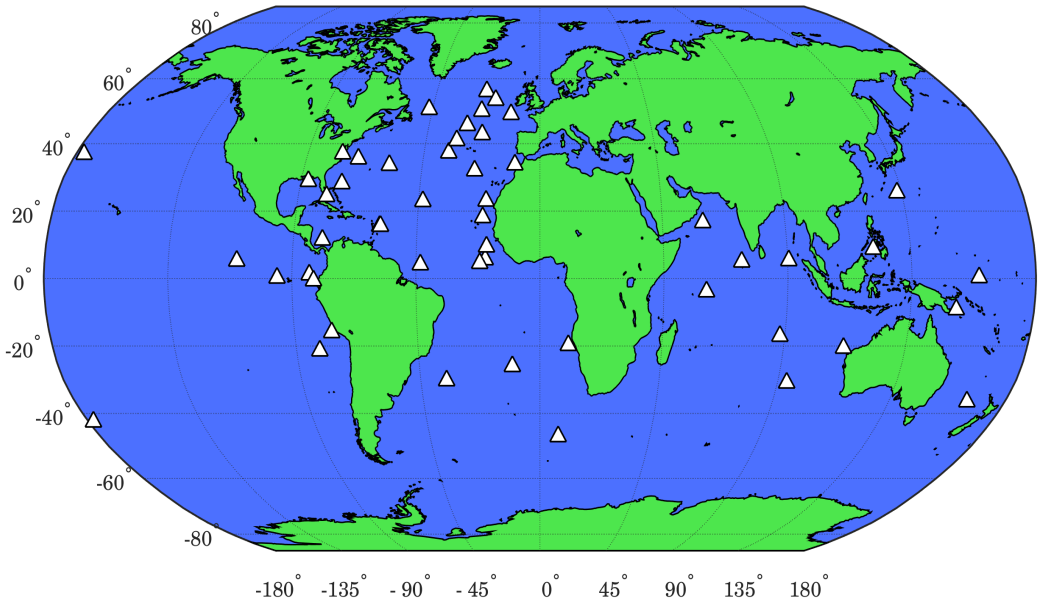


Figure 4.1: Illustration of the 51 marine sites (white triangles) at which mid-Pliocene ocean temperature estimates are available. These are mainly derived from the analysis of planktonic foraminifer assemblages (fauna) in ocean sediments.

land components of the Earth system. Similar numerical models integrate forward in time systems of coupled partial differential equations (mainly of fluid-dynamics type, such as Navier-Stokes), on a non-inertial system such as a rotating sphere. A number of other thermodynamic sources (solar radiation, albedo) and the so-called boundary conditions (presence of vegetation, sea-ice, amount of CO_2 in the atmosphere, topography) are also accounted for in the model.

In the case of HadCM3, the outputs of the numerical integration are provided within a three-dimensional grid. The Earth surface is divided into a two-dimensional grid, and the third "vertical" dimension (high in the atmosphere or deep in the ocean) is in turn divided into a number of layers: see Figure 4.2 for an illustration. The vertical layers are 19 for the atmosphere, and 20 for the ocean. We consider here the model output corresponding to the upper level of the ocean component, the SST. As explained above, values for this are provided over a two-dimensional grid. This has a resolution of 1.25° in longitude by 1.25° in latitude, yielding a total of $s = 144 \times 288$ cells.

The model can be run with a variety of initial and boundary conditions, which can be set to match configurations of past or future epochs. Within a climate simulation

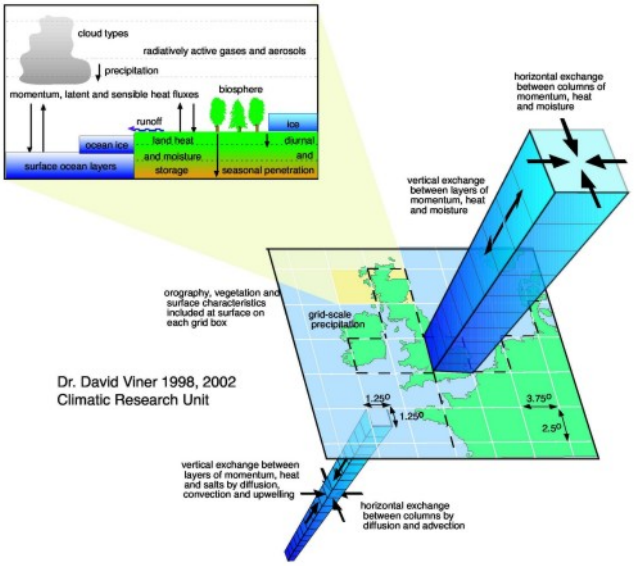


Figure 4.2: Illustration of the HadCM3 grid structure. Figure kindly provided by the IPCC. Above sea level, the atmospheric component of the model is divided into 19 layers; below sea level, the oceanic component is divided into 20 layers. For the atmospheric component, the grid-size resolution is 3.75° in longitude by 2.5° in latitude. That refines to 1.25° in both longitude and latitude for the oceanic component.

context, the term boundary condition refers to the set of vegetation, orbital forcing, CO_2 concentration, topography which a given run is configured with. These are usually kept fixed throughout a run. The term initial condition is instead to be interpreted in the classical mathematical way, as the prescribed state at time $t = 0$ of the variables appearing in a system of differential equations. In practice, these are usually extracted as the final condition of a previous simulation, which has been specifically run in order for the quantity of interest to reach a quasi-stationary behaviour.

4.4. Simulator Inputs: Orbital Parameters

In order to study the evolution of the mid-Pliocene SST, we run different simulations. The same boundary conditions are shared between any two of them, with the only exception of some astronomical quantities, which we refer to as orbital parameters: these are adjusted to the specific time simulated. Orbital parameters are therefore to be considered the input of our simulations. In this section we introduce these, alongside an illustration of the relevant astronomical phenomena which are responsible of long

time-scale changes in the Earth's climate.

The temperature on our planet is deeply affected by the distribution of solar radiation reaching the top of the atmosphere. This quantity, known as insolation (with dimensions of $[E]/([L]^2 \times [T])$), is subject to significant changes over millennia scales. The varying astronomical configuration of our planet during its revolution around the Sun is at the basis of these changes, and of the consequent alternation of colder and warmer eras known as glacial and interglacial cycles. As theorised by the astronomer, mathematician and climatologist Milutin Milanković (Milanković [1930]), three astronomical phenomena are responsible for the succession of different climate patterns on the Earth. These are: changes in the *eccentricity* of Earth's orbit, changes in the *obliquity* of Earth's axis, and the *precession of equinoxes*. We illustrate these in the following subsection and specify variables to measure them.

4.4.1. Description of Relevant Astronomical Phenomena

Changes in Eccentricity

The eccentricity of Earth's orbit characterises how close to a circle the elliptical orbit of our planet is. We denote it by e. From its geometrical definition ($e = \sqrt{a^2 - b^2}/a$, where a and b are the major and minor semi-axes of the ellipse, respectively), it is clear that $e \in [0, 1)$, with $e = 0$ representing the perfect circle case.

Earth's eccentricity is not constant over time. Currently, it is about 0.0167 and decreasing. Its value has always been below 0.06 in the last 25 million years (Laskar et al. [2004], Berger and Loutre [1991]), a figure that shows how Earth's orbital shape has never been too dissimilar from a circular one. However, even within such a small range, higher eccentricity values induce greater variations in the amount of insolation received by the Earth during different times of the year. The greatest difference is observed between the times of perihelion and aphelion: in a non-circular orbit, these represent the closest point to the Sun, and the farthest point from the Sun, respectively. It is also worth noting that a non-zero eccentricity affects the duration of seasons. Earth's orbital velocity is indeed faster near perihelion than near aphelion, as per the second Kepler's law.

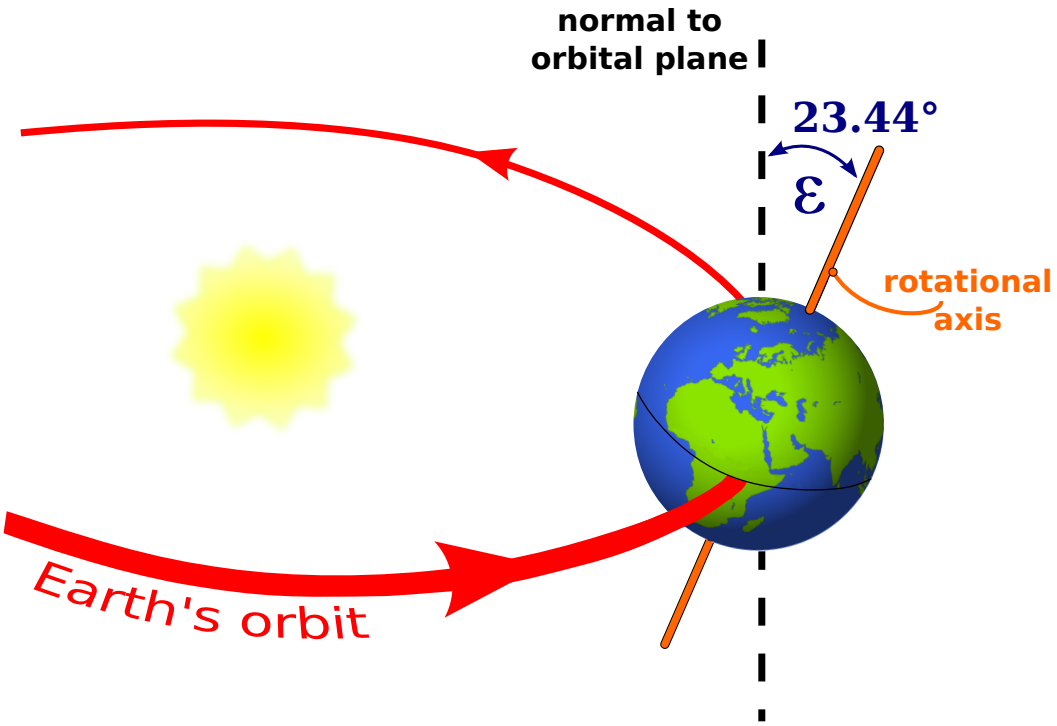


Figure 4.3: Obliquity ε, defined as the positive angle between Earth's rotational axis and the normal to Earth's orbital plane. The illustration displays its current value, $\varepsilon = 23.44°$.

Changes in Obliquity

The obliquity of Earth's axis is defined as the angle between the plane going through Earth's equator (equatorial plane) and the plane to which Earth's orbit belongs (orbital plane). We denote it by ε. Equivalently, it is the angle between Earth's rotational axis and the normal to the orbital plane, see Figure 4.3.

A non-zero obliquity causes each of the two hemispheres to receive more insolation in certain periods of the year than in others, causing the seasons' alternation. Higher obliquity values increase the contrast between seasons. As it was the case for eccentricity, obliquity is not constant over time. The current value is around 23.44° and it is decreasing. However, the range of possible obliquity values is quite constrained: according to Laskar et al. [2004], values for the last 250 million years and predictions for the forthcoming 250 million years have been and are between 21.5° and 25°.

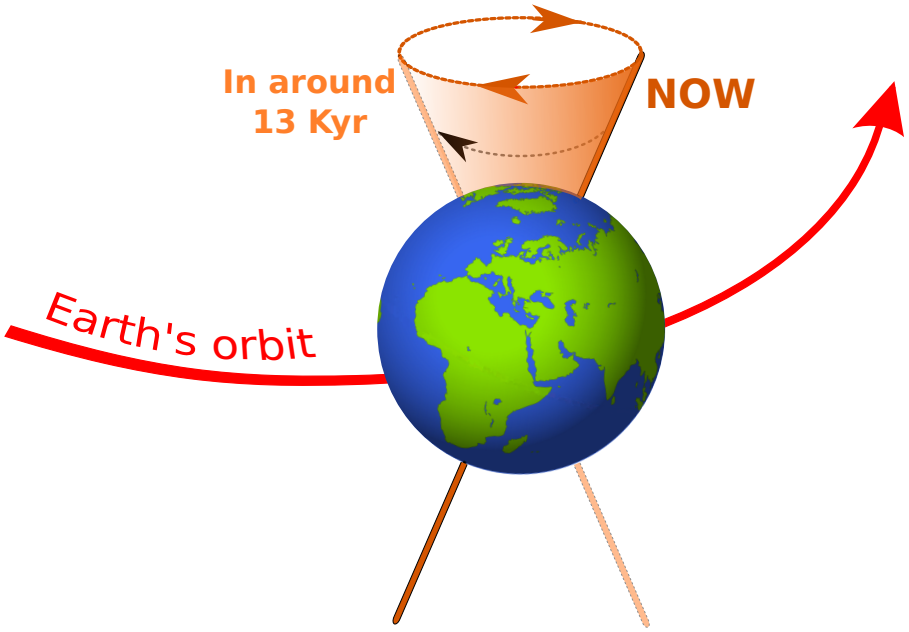


Figure 4.4: Representation of the axial precession, a wobbling movement of the axis spacial direction with periodicity of approximately 26 Kyr. Notice the clockwise motion, as opposed to the counter-clockwise one of Earth's revolution.

Precession of Equinoxes

According to Milanković theory, the last relevant phenomenon affecting Earth's climate is the so-called precession of equinoxes. Differently from the two previous cases, it is not straightforward to identify a variable to characterise it. Hence, we first describe the phenomenon and then give a precise definition of the quantity we use to measure it.

The Sun's and the Moon's gravitational attraction, alongside the not perfectly spherical shape of our planet, induce a slow change over time in the spatial direction of Earth's rotational axis: in around 26 thousands years, this completes a full rotation around the normal to the orbital plane, spanning the surface of an imaginary cone. This phenomenon is known as axial precession, see Figure 4.4 for an illustration. From a mathematical-physics point of view, it is the same phenomenon characterising the motion of a rotating spinning top on a flat surface.

As a consequence of axial precession, the position that the Earth occupies during key

astronomical times of the year, such as equinoxes and solstices, slowly shifts along Earth's orbit. The two equinoxes are the times of the year at which Earth's equatorial plane goes exactly through the centre of the Sun: the northern and southern hemispheres receive the same amount of insolation, and day and night have equal length at all locations. The two solstices are instead the times at which the distance between the Sun and the equatorial plane is at its maximum: they are characterised by the largest difference between the duration of day and night. These astronomical events currently befall around the 20^{th} of March (spring equinox), the 21^{st} of June (summer solstice), the 22^{nd} of September (autumnal equinox) and the 21^{st} of December (winter solstice).

If seen from above Earth's orbital plane, the axial precession is a clockwise motion; on the contrary, the revolution of our planet around the Sun follows a counter-clockwise motion. Compare to Figure 4.4. The overlap of these two "counteracting" phenomena causes the equinoxes to occur each year around 20 minutes earlier than the year before. This resulting phenomenon is known as *Precession of the Equinoxes*. We uniquely identify the direction of Earth's rotational axis via the heliocentric angle ω, in the orbital plane, going from the position of Earth during autumnal equinox to the perihelion. See Figure 4.5.

Due to the precession of Equinoxes, the angle ω slowly increases over time. Currently, it is approximately $102.9°$ (Figure 4.5 depicts this situation). In terms of angles, this means that $12.9°$ after the winter solstice the perihelion is reached: this indeed happens on the 3^{rd} of January, which not surprisingly is approximately 13 days after the winter solstice. In general, the value of ω affects the strength of seasons in the different hemispheres: due to what just described, the northern hemisphere currently experiences milder winters and cooler summers than the southern hemisphere. But the situation would be reversed if ω was close to $270°$ (that is, perihelion close to the position of summer solstice).

Summing up, the three parameters we are going to consider to identify the astronomical configuration of the Earth at different times are as follows: the eccentricity e of Earth's orbit; the obliquity ε of Earth's axis; the angle ω between autumnal equinox and perihelion. These will be used as inputs to the HadCM3 simulations.

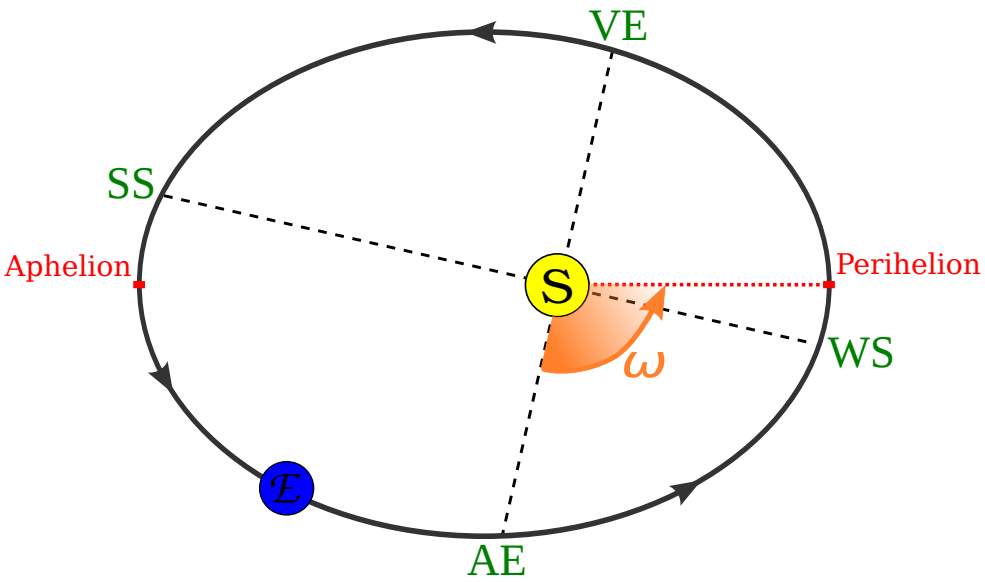


Figure 4.5: Schematic representation of the angle ω we use to measure axial precession and the related precession of equinoxes phenomenon. View is from the northern half-space delimited by the Earth's orbital plane. The drawn positions of Earth at Winter Solstice (WS), Spring or Vernal Equinox (VE), Summer Solstice (SS), and Autumnal Equinox (AE) portray the current configuration. In this configuration, $\omega \approx 102.9°$. The eccentricity of Earth's orbit is exaggerated for illustrative purposes.

4.5. Experimental Design

In order to build an emulator of the SST output field of the HadCM3 simulator, specifically during the mid-Pliocene, a relatively small number of simulations must be run to train the emulator. The choice of initial simulations is very important. These need to be low in number, due to the high cost that each simulation requires, both in terms of time (around two weeks in our case) and computational power. At the same time, within the areas of the input space that are of interest, a sufficient number of initial simulations must be run to ensure a reliable calibration of the emulator. As discussed in Subsection 1.1.2, the problem of choosing the parameters in the input space at which to run the initial simulations is referred to as the "design problem" (Santner et al. [2003]). We call *experimental design* the set $\mathcal{D} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ of these parameters: in our case, each $\boldsymbol{x_i}$ is obtained from a triple (e, ε, ω) of orbital parameters (see Section 4.4), via the transformation detailed in Subsection 4.5.2.

The problem of designing computer experiments (Sacks et al. [1989], Santner et al.

[2003]) requires a trade-off between minimising the number of runs, and maximising the amount of information that can be extracted from these. If there is no particular reason to concentrate design runs in one specific region of the space, a common approach is to implement a design with a space-filling property: informally, this scatters points evenly within the space to explore with the aim of covering all areas, while ensuring that any two points are never too close. This is often reached by minimising a measure of discrepancy of a given sequence. We will not go into the details here, which can be found in Santner et al. [2003, Chap. 5]: while similar designs are popular in the emulation literature (Bonceur et al. [2015], Holden et al. [2018], Wilson et al. [2018]), for the present study we make use a different design[10]. Before providing a description of this, we point out the reasons underlying the choice.

1. When the author started his PhD, a set of mid-Pliocene simulations were available. These had been run and processed by Caroline Prescott, at the time Earth Science PhD student at Leeds, as part of her PhD work (Prescott [2017]). They had been specially designed to explore the varying mid-Pliocene orbital forcing and the associated temperature response. It was therefore natural to include the results of these simulations in the emulator construction.

2. The author re-ran and processed two of the simulations, using the most recent version of the HPC facilities available at Leeds. These had undergone an update since the time of Caroline's experiments. Although the same parameters were used, the results differed: unsurprisingly, even tiny numerical representation differences were becoming notable when propagated forward in time by the model, which is, by its nature, chaotic.

3. Given the above incompatibility, including the results of new simulations in the emulator calibration was deemed unsound, since it could have undermined the results. The design originally developed by Caroline Prescott was therefore left unchanged and used to construct the SST emulator of this chapter.

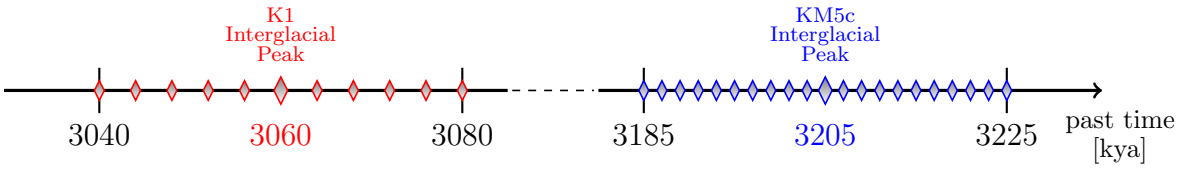
We provide details of the design in Subsection 4.5.1 and Subsection 4.5.2.

[10] An example of emulation whose design is obtained through the use of low-discrepancy sequences with the space-filling property is provided in Chapter 5.

4.5.1. Uniform Sampling in Time

The design adopted here aims at assessing the climate variability around two interglacial events of the mid-Pliocene: these are referred to as marine isotope stages K1 and KM5c, which date back to around 3,060 and 3,205 kya, respectively. In Figure 4.6, their position within the PRISM3D time slab is highlighted via a red border. As all interglacials, K1 and KM5c were warm periods[11]: the times 3,060 and 3,205 kya identify the interglacial peaks. As shown in Figure 4.6, the KM5c peak was characterised by an orbital configuration relatively similar to the current one. This makes the time 3,205 kya a particular reference in DMC.

The experimental design consists of a total of $n = 32$ simulations, run with orbital configurations corresponding to times up to 20 thousand years before and after each of the two interglacial peaks. Specifically, one time every four thousand years is sampled in the interval around the K1 peak, and one time every two thousand years is sampled in the interval around the KM5c peak. This yields 11 time points in the interval 3,040–3,080 kya, and 21 time points in the interval 3,185–3,225 kya, as depicted below.



In order to obtain the orbital parameters (a triple of the form (e, ε, ω)) corresponding to each of these times, the web-based interface available at http://vo.imcce.fr/insola/earth/online/earth/online/ is used. This implements the astronomical solution developed in Laskar et al. [2004], which returns estimates of the orbital parameters and insolation values for any time between 100 million years ago and 20 million years in the future. Any two design simulations differ only in the orbital configuration that is imposed; all other boundary conditions – imposing vegetation, land-sea mask, CO_2 concentrations and vegetation appropriate for the mid-Pliocene – are shared among the simulations.

[11] Note that the letters K and M, alongside the digit after them, help locate the two periods with respect to the Kaena and Mammoth epochs; compare to Figure 4.6.

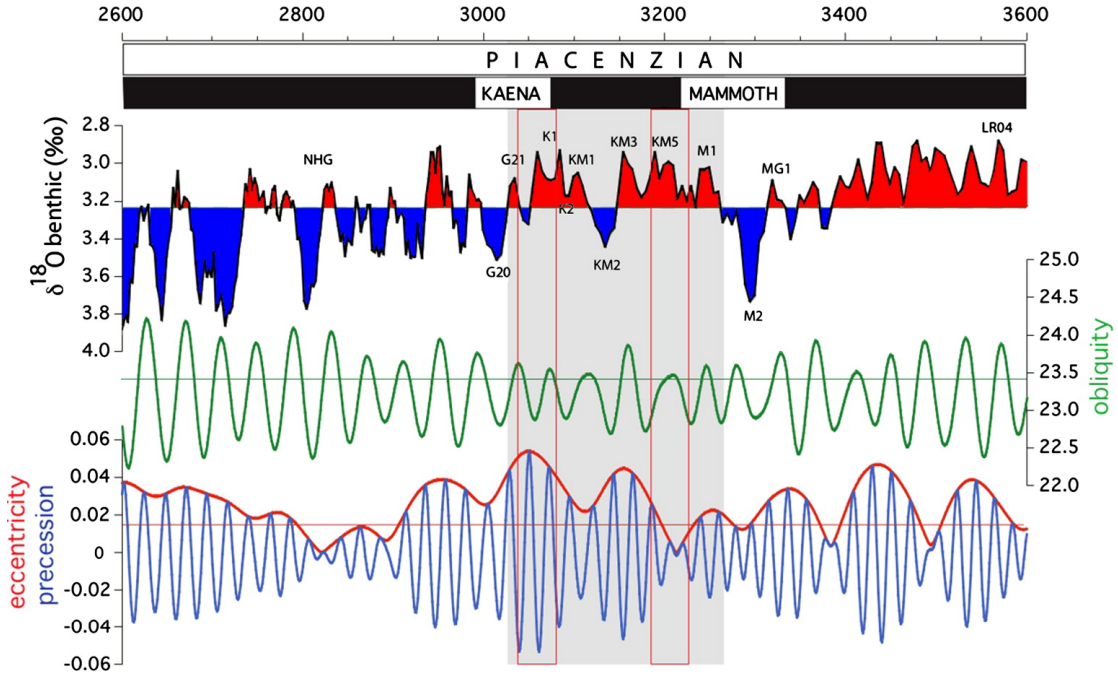


Figure 4.6: Illustration taken from Prescott et al. [2014]. Top plot: benthic oxygen isotope excursions over the period 2,600–3,600 kya; lower values (red areas) are associated with higher temperatures. Bottom three plots: evolution of obliquity (green), eccentricity (red) and precession (blue). Values from Laskar et al. [2004]. The precession ω is plotted as $\sin(\omega)$ and further modulated by eccentricity. Horizontal green and red lines show the current obliquity and eccentricity values. Throughout the plots, red borders highlight the time intervals associated with our simulations, around the two interglacial events K1 and KM5c. The broader shaded band identifies the PRISM3D time slab (3,025–3,264 kya).

4.5.2. Transformed input variables

Let us recall from Section 4.4 that the orbital parameters here considered are the following:

- the eccentricity $e \in [0, 1)$ of Earth's orbit;
- the obliquity $\varepsilon \in [0, \pi]$ of Earth's axis;
- the angle $\omega \in S^1$ measuring precession.

The intensity of climatic differences induced by precession – in particular, the strength of seasons in each of the two hemispheres – is naturally dampened by eccentricity: if the orbit is very close to a circle ($e \approx 0$), the Earth-Sun distance is almost constant

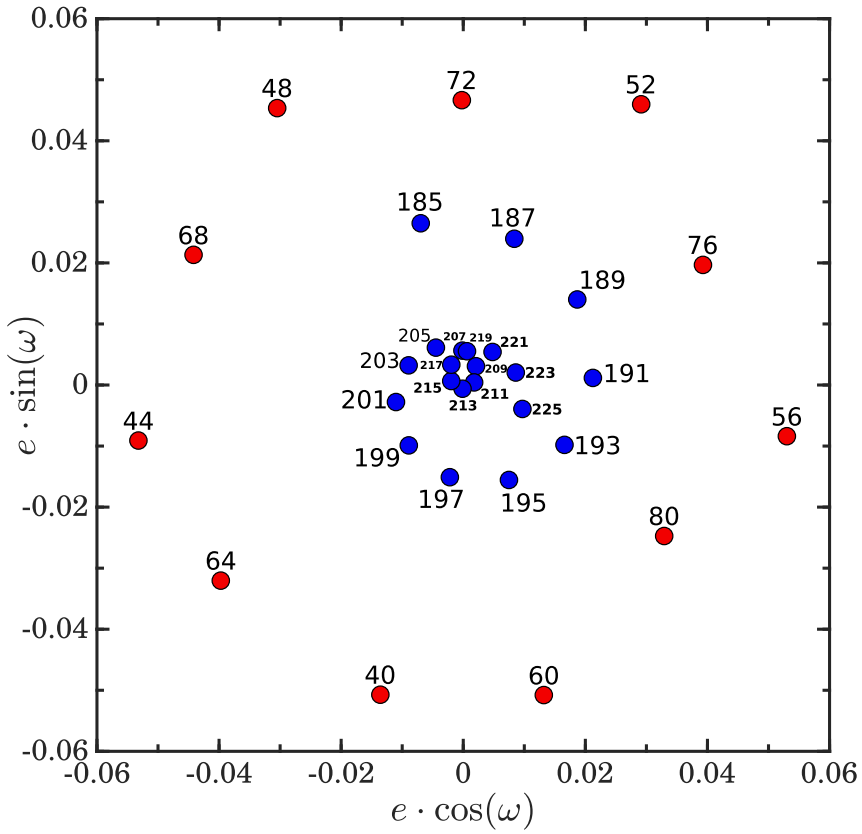


Figure 4.7: Projection of the $n = 32$ points used as design to build the emulators of this chapter, onto the coordinate plane spanned by the last two coordinates of (4.1). Projections onto the two other coordinate planes are shown in Figure 4.8. Red points correspond to times around the K1 peak, blue points correspond to times around the KM5c peak. Label x identifies the time (3000+x) kya.

throughout the year, hence little difference will be present between the two cases where summer happens close to aphelion or to perihelion. For this reason, rather than via a triple (e, ε, ω), we parameterise the inputs of each simulation as follows:

$$\boldsymbol{x} = \begin{pmatrix} \varepsilon \\ e\cos(\omega) \\ e\sin(\omega) \end{pmatrix} \in \mathcal{P}. \tag{4.1}$$

As in Chapter 2, we denote by $\mathcal{P} \subseteq \mathbb{R}^p$ the set of input parameters. In this case, we have $p = 3$. For ease of reference, Table 4.1 reports the value of this and other constants used throughout this chapter. The choice of representing an angle $\omega \in S^1$ via the pair

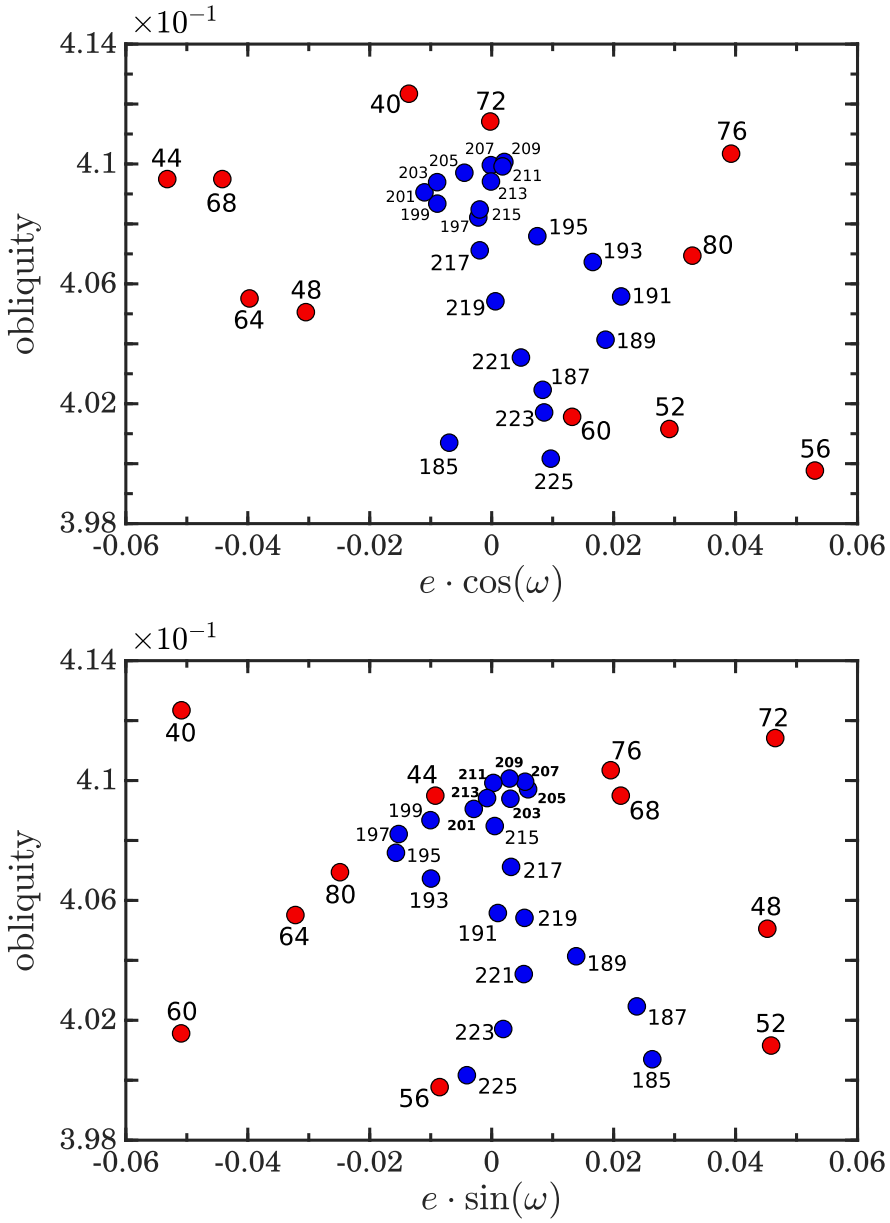


Figure 4.8: Same content of Figure 4.7. Here, projections are onto the two coordinate planes containing obliquity, denoted as ε in equation (4.1). Red points are associated with times around the K1 peak, blue points with times around the KM5c peak.

$(\cos(\omega), \sin(\omega))$ is convenient to avoid discontinuities; moreover, the eccentricity factor in the last two components of (4.1) reflects the dampening effect of eccentricity onto precession, as described above.

Figure 4.7 and Figure 4.8 show the coordinates of each design input parameter, with

Table of Constants and Notation

Name	Meaning	Value
n	Number of design points	32
p	Dimension of simulator input space	3
s	Dimension of simulator output space	144×288 $(41,472)$
r	Number of PCs used to approximate simulator outputs (equation (4.6))	6
N_s	Number of samples of geologically reconstructed SST time series at a given marine site	site dependent
N_p	Number of peaks in geologically reconstructed SST time series at a given marine site	site dependent
N	Number of WPA estimates from emulator at a given marine site (Section 4.10)	$1,000$
i	Index used to denote quantities associated with each of the n simulations	$i \in \{1, \dots, n\}$
j	Index used to denote quantities associated with the first r PCs	$j \in \{1, \dots, r\}$

Table 4.1: Table reporting the meaning and value of the main constants used in the chapter. Most of these will be introduced in later pages, but are here reported to provide a compact reference. The last two lines concern the use of two indices frequently employed in the chapter.

respect to the parameterisation in (4.1). From the pictures, it can be appreciated that the regularity with which the simulations are scattered in time, in each of the two intervals around the K1 and KM5c interglacials, is partly lost in the coordinates used to build the emulator. Once again, we stress that a design based on low-discrepancy sequences would cover the space in a more uniform way than the one presented here. On the other side, it is worth noting that our design allows to reduce the uncertainty of predictions in the region around the KM5c interglacial (characterised by low eccentricity), when the orbital configuration was similar to today.

4.6. Reducing Output Dimensionality

Each of the n simulations is initialised with orbital forcing corresponding to one of the times described in Subsection 4.5.1, and run for subsequent 500 simulated years. This allows the simulated surface climatology to reach an equilibrium under a particular forcing condition. The average of the last 100 years of the simulated SST is then computed, at each of the s cells constituting the simulator output grid. We can therefore represent each output as a s-dimensional vector, and the simulator as a function $\boldsymbol{\gamma}$ associating a vector of this form to any input parameter in $\mathcal{P}$:

$$\boldsymbol{\gamma}\colon \mathcal{P} \to \mathbb{R}^s\,. \tag{4.2}$$

In reality, as discussed in Chapter 3, for any $\boldsymbol{x} \in \mathcal{P}$ each vector $\boldsymbol{\gamma}(\boldsymbol{x})$ represents a discretisation of a map from the sphere S^2 into the real numbers. In order to reduce the high dimensionality of the problem, we can therefore apply the procedure described therein, to the data set $\mathcal{Y}$ formed by the n simulator outputs corresponding to the n design points:

$$\mathcal{Y} = \left\{\boldsymbol{\gamma}(\boldsymbol{x_i}) \in \mathbb{R}^s\right\}_{i=1,\dots,n}. \tag{4.3}$$

Recall from Chapter 3

The approach looks at the affine space generated by the vectors in (4.3) and finds a sequence of orthonormal directions explaining respectively most-to-least of the data set variance. The procedure differs from a classical PCA approach, in that it considers each vector in (4.3) as a function from S^2 to $\mathbb{R}$, weighting the different components of $\boldsymbol{\gamma}(\boldsymbol{x_i})$ by the area associated with the cell they represent.

The orthogonal directions identified by our procedure are the Principal Components (PCs). Although they are here introduced as s-dimensional vectors, given their importance in the chapter and their natural parallel interpretation as 144×288 matrices, we denote them via a capital letter: $\boldsymbol{V_1}, \dots, \boldsymbol{V_{n-1}}$.

As per equation (3.52), each $\boldsymbol{\gamma}(\boldsymbol{x_i}) \in \mathbb{R}^s$ can be written as affine combination of the

Order of PC	**1st**	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**
% Explained Variance	48.43	24.47	10.92	7.65	2.34	1.16	0.73	0.69	0.51
Cumulative % Expl. Var.	48.43	72.90	83.87	91.52	93.86	95.02	95.75	96.43	96.95

Table 4.2: Percentage of variance explained by the first nine PCs (total number of PCs: $n-1=31$). Both single and cumulative percentages are reported.

PCs, with constant intercept term given by the average $\bar{\boldsymbol{\gamma}}$ of $\boldsymbol{\gamma}(\boldsymbol{x_1}), \ldots, \boldsymbol{\gamma}(\boldsymbol{x_n})$. That is:

$$\boldsymbol{\gamma}(\boldsymbol{x_i}) = \bar{\boldsymbol{\gamma}} + \sum_{j=1}^{n-1} f_{ij} \boldsymbol{V_j}\,, \qquad i = 1, \ldots, n\,, \tag{4.4}$$

for some coefficients $f_{ij} \in \mathbb{R}$ identified by the PCA procedure (denoted by q_{ij} in equation (3.52)). Note that, as summarised in Table 4.1, we use the index i to refer to the n observed simulator outputs (and associated quantities), and the index j to refer to the $n-1$ PCs (and associated quantities) or to a subset of these.

If we only retain a number $r < n-1$ of PCs, then the RHS of (4.4) provides an approximation of $\boldsymbol{\gamma}(\boldsymbol{x_i})$. One of the advantages of the PCA approach is that relatively small values of r may already yield excellent approximations, since higher PCs account for less of the variability displayed by the data set. In our case, the first six PCs together explain more than 95% of the data set variance, while each of the remaining PCs alone explains less than 1%, see Table 4.2. In this regard, let us recall that the variance associated with the j^{th} PC is the empirical variance

$$\sigma_j^2 = \mathrm{Var}\Big(\{f_{ij}\}_{i=1,\ldots,n}\Big)\,. \tag{4.5}$$

Values for σ_j^2 are provided as part of the spectral decomposition detailed in Chapter 3 (specifically, see Proposition 3.3.2 where they are referred to as λ_j).

A plot of the first six PCs is shown in Figure 4.9. Important physical patterns can be recognised in some of them. The first PC reflects the great variability displayed

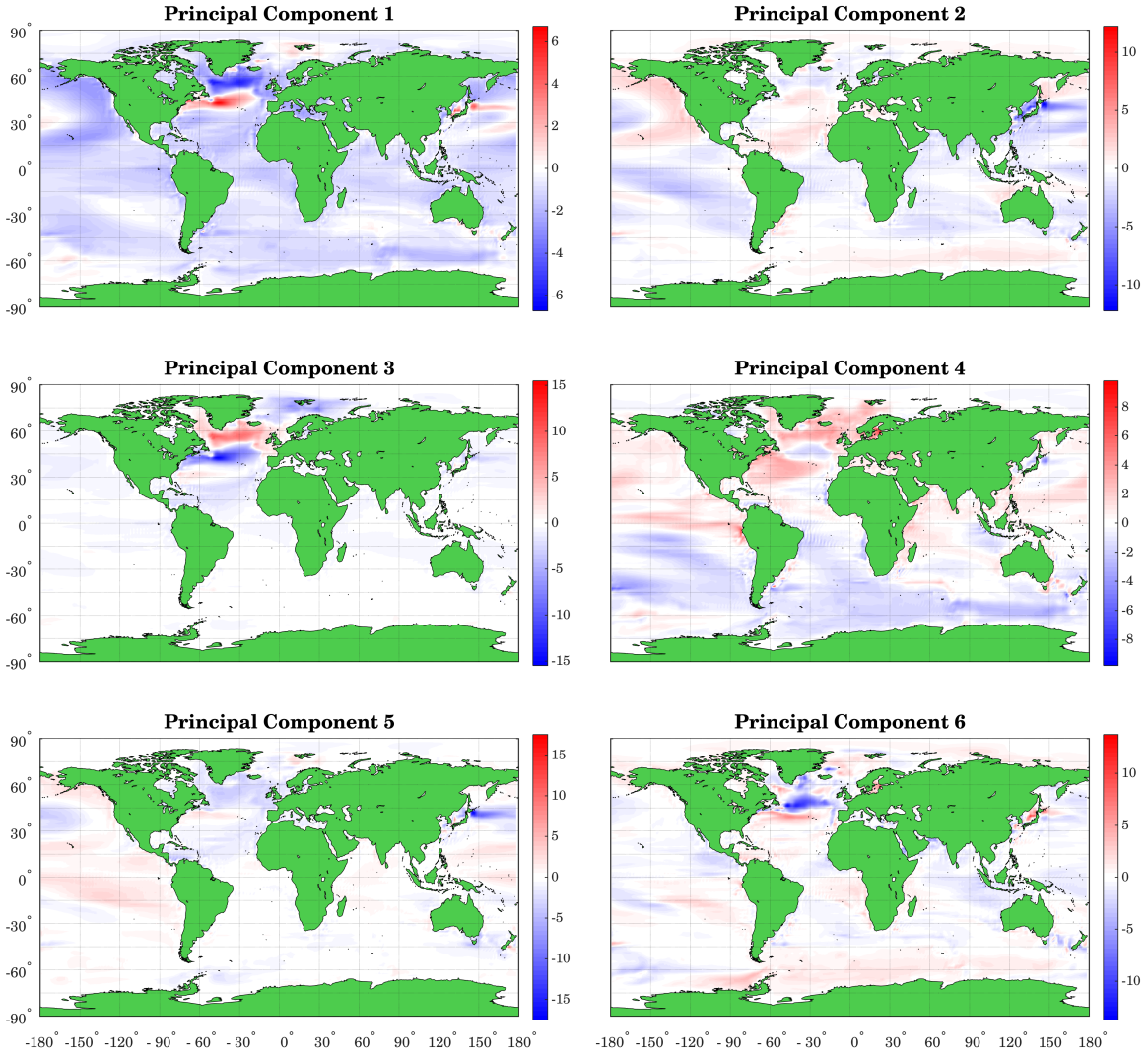


Figure 4.9: Plot of the first six PCs, with scales on the side. Values have however no physical meaning: each PC has norm equal to one, with respect to the scalar product used to carry out the PCA (details in Subsection 3.3.1).

by the different runs in two North Atlantic areas; this is a well-known feature of HadCM3, in Prescott et al. [2014] referred to as North Atantic dipole feature. Also the third PC appears to be linked to the phenomenon, probably reflecting temperature changes associated with the Gulf Stream. The second PC captures more of the Pacific oscillations. The fourth PC seems to encode most of the North-South temperature variability present among the different runs, which is likely to reflect precession changes. Meaningful patterns can also be recognised in higher PCs. However, visual inspection of the PCs higher than the 7^{th} hardly reveals physically relevant features.

From a mathematical point of view, equation (4.4) says that the PCs $\boldsymbol{V_1}, \dots, \boldsymbol{V_{n-1}}$ form a basis of the space generated by the n elements $\boldsymbol{\gamma}(\boldsymbol{x_i}) \in \mathcal{Y}$ (more precisely, these generate an $(n-1)$-dimensional affine subspace in $\mathbb{R}^s$, and the PCs are a basis of its underlying vector space). Moreover, once projected onto the space generated by the first six PCs, the data set still retains 95% of its original variability. In light of this consideration, and of the previous one on the physical meaning of the first PCs, we approximate any unknown simulator output $\boldsymbol{\gamma}(\boldsymbol{x}) \in \mathbb{R}^s$ by an element $\widetilde{\boldsymbol{\gamma}}(\boldsymbol{x})$ belonging to the subspace generated by the first $r = 6$ PCs. That is, we consider:

$$\widetilde{\boldsymbol{\gamma}}(\boldsymbol{x}) := \bar{\boldsymbol{\gamma}} + \sum_{j=1}^{r} f_j(\boldsymbol{x})\, \boldsymbol{V_j} \in \mathbb{R}^s, \qquad \boldsymbol{x} \in \mathcal{P}. \tag{4.6}$$

For general $\boldsymbol{x} \in \mathcal{P}$, the coefficients $f_1(\boldsymbol{x}), \dots, f_r(\boldsymbol{x})$ are unknown. However, if $\boldsymbol{x} = \boldsymbol{x_i} \in \mathcal{P}$ is one of the design points, then from equation (4.4) it is natural to impose the following condition[12]:

$$f_j(\boldsymbol{x_i}) = f_{ij}, \qquad i = 1, \dots, n, \;\; j = 1, \dots, r. \tag{4.7}$$

This reduces an s-dimensional problem (predicting $\boldsymbol{\gamma}(\boldsymbol{x}) \in \mathbb{R}^s$) to r one-dimensional problems (predicting $f_j(\boldsymbol{x})$ for $j = 1, \dots, r$).

It is therefore apparent that the setting we have reduced our original problem to is exactly the one of univariate emulation presented in Chapter 2. Indeed, for each $j = 1, \dots, r$, we have a scalar field

$$f_j \colon \mathcal{P} \to \mathbb{R}, \tag{4.8}$$

whose outputs at n design points $\boldsymbol{x_i} \in \mathcal{P}$ are known (we will denote these by $y_i^{(j)} := f_j(\boldsymbol{x_i})$ in analogy with the notation of Chapter 2), and whose outputs at all other $\boldsymbol{x} \in \mathcal{P}$ must be predicted. In the next sections, we provide the details of the choices adopted to construct the r emulators of the PC scores $f_j(\cdot)$. Equation (4.6) will then allow us to approximate $\boldsymbol{\gamma}(\boldsymbol{x}) \in \mathbb{R}^s$, for any $\boldsymbol{x} \in \mathcal{P}$.

[12] Note that condition (4.7) is the choice which minimises the norm $\|\widetilde{\boldsymbol{\gamma}}(\boldsymbol{x}) - \boldsymbol{\gamma}(\boldsymbol{x})\|_{L^2(\mathcal{H})}$, when $\widetilde{\boldsymbol{\gamma}}(\cdot)$ varies in the space of functions from S^2 to $\mathbb{R}$ generated by the first r PCs.

4.7. Prior Specifications for PC Scores

We have described the procedure to emulate a scalar field, such as each of the functions $f_j \colon \mathcal{P} \to \mathbb{R}$, in Chapter 2. Formulas (2.96) and (2.97) in particular provide expressions for the posterior mean and covariance functions of the emulator. In order to apply them, we need to choose the prior mean and covariance functions for our specific problem. This is done in the next two subsections.

4.7.1. Mean Function

The prior mean function of an emulator (equation (2.4)) is specified as follows:

$$m_\beta(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T \boldsymbol{\beta}\,, \quad \boldsymbol{x} \in \mathcal{P}, \tag{4.9}$$

where $\boldsymbol{h}(\cdot)$ is a vector of q basis functions of the input parameter $\boldsymbol{x} \in \mathcal{P}$. Let us recall that, within the Bayesian framework of GP emulation presented in Chapter 2, the coefficient $\boldsymbol{\beta} \in \mathbb{R}^q$ is marginalised out (Section 2.5 and Section 2.7).

As far as the function $\boldsymbol{h}(\cdot)$ is concerned, a common choice in the literature is to set $\boldsymbol{h}(\boldsymbol{x}) = (1, \boldsymbol{x})$, *i.e.*, to have the prior mean linear in the input $\boldsymbol{x}$ (Bonceur et al. [2015], Araya-Melo et al. [2015]). In our case (recall equation (4.1)), the coordinates of the vector $\boldsymbol{x}$ are as follows:

$$\begin{aligned} x_1 &= \varepsilon, \\ x_2 &= e \cdot \cos(\omega), \\ x_3 &= e \cdot \sin(\omega)\,. \end{aligned}$$

In order to evaluate whether a linear prior mean is appropriate in our context, for each PC $j = 1, \ldots, r$ we perform a linear regression of the response vector $\boldsymbol{y}^{(j)} \in \mathbb{R}^n$ with components

$$y_i^{(j)} := f_{ij}\,, \quad i = 1, \ldots, n\,, \tag{4.10}$$

as a function of the regressors x_1, x_2, x_3. The results are generally poor, with a coefficient of determination R^2 less than 0.25 in four of the first six cases of interest ($j = 1, \ldots, r$). Further inspection shows that the variability of the components of $\boldsymbol{y}^{(j)}$

corresponding to higher eccentricities (times around K1) is often higher than the one shown by components corresponding to lower eccentricity (times around KM5c): see Figure 4.7. Within the (x_2, x_3)-plane, this reveals a dependence of the response vector on the distance of a point from the origin (indeed, $e^2 = x_2^2 + x_3^2$). In order to account for this, we consider the following set of potential regressors:

$$\mathcal{R} = \{x_1,\, x_2,\, x_3,\, {x_2}^2,\, {x_3}^2\}\,. \tag{4.11}$$

For each response vector $\boldsymbol{y}^{(j)}$, we run all linear regressions comprising exactly three of the five regressors in $\mathcal{R}$ plus intercept. A total of $\binom{5}{3} = 10$ linear regressions are therefore run (for fixed j). Hence, we select the set of three regressors which yields the greatest R^2 and use these as regressors for the prior mean $m_\beta(\cdot)$ in (4.9). In other words, if the selected subset of regressors corresponding to the vector $\boldsymbol{y}^{(j)}$ is $\{x_{j1}, x_{j1}, x_{j3}\} \subseteq \mathcal{R}$, then we define:

$$\boldsymbol{h_j}(\boldsymbol{x}) = \begin{pmatrix} 1 \\ x_{j1} \\ x_{j2} \\ x_{j3} \end{pmatrix}, \qquad \boldsymbol{x} \in \mathcal{P}. \tag{4.12}$$

The subscript j highlights that the set of regressors depends on the PC in question.

As explained before equation (4.11), the choice of adding ${x_2}^2$ and ${x_3}^2$ to the set of potential regressors was made to include dependence on the square of eccentricity. In hindsight, we remark that this choice should have been accompanied by the inclusion of the cross-product x_2x_3 among the set $\mathcal{R}$ of potential regressors (equation (4.11)). This would account for the fact that the choice of representing an angle $\omega \in S^1$ by a pair $(\cos\omega, \sin\omega)$ is arbitrary and any other non-zero phase could have been chosen.

Table 4.3 shows which regressors our procedure selects for the first six PCs, alongside the R^2 value of the corresponding linear regression. At least one of ${x_2}^2$ and ${x_3}^2$ is always selected, and both regressors are selected in four of the six cases. Also notice that the choice of limiting the number of regressors to three is adopted with the aim of avoiding overfitting, in consideration of the limited amount of training data available for each emulator ($n = 32$ data points).

	ε	$e\cos(\omega)$	$e\sin(\omega)$	$e^2\cos^2(\omega)$	$e^2\sin^2(\omega)$	R^2
1st PC			✓	✓	✓	0.95
2nd PC			✓	✓	✓	0.98
3rd PC		✓	✓		✓	0.14
4th PC	✓	✓		✓		0.96
5th PC	✓			✓	✓	0.49
6th PC	✓			✓	✓	0.38

Table 4.3: Set of regressors used as basis of the emulator prior mean, for each of the first six PC scores to be emulated. The last column shows the R^2 value of a standard linear regression carried out with the specified regressors. Variables to construct the regressors: obliquity (ε); eccentricity (e); precession (ω).

4.7.2. Covariance Function

The prior covariance function of an emulator (equation (2.5)) is specified as follows:

$$v_{\sigma^2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \, c(\boldsymbol{x}, \boldsymbol{x}') \, . \tag{4.13}$$

Within the GP emulation setting, the coefficient σ^2 is marginalised out alongside the coefficient $\boldsymbol{\beta} \in \mathbb{R}^q$ used to specify the prior mean (Section 2.5). The function $c(\cdot, \cdot)$ should instead be specified. Here, we choose it to be the Matérn correlation function with parameter $\nu = 5/2$: its expression is provided in equation (1.58). Moreover, we add a nugget term to the prior covariance. As described in Section 2.8, this allows the emulator to return probabilistic predictions even at the design points $\boldsymbol{x_i} \in \mathcal{P}$.

The choice to include a nugget term is mainly motivated by the chaotic behaviour displayed by the simulator. This can be seen by running simulations with almost identical input parameters, or, similarly, by running on different machines two simulations with the same input parameters. In both cases, the results of the pair of simulations will be significantly different from each other: as mentioned in point 2 of Section 4.5, this was checked by the author in the second case. It is however important to point out that the simulator is deterministic: running the same simulation twice,

on the same machine, yields the same output. The chaotic behaviour displayed by the simulator is not surprising, since the physics driving the climate system is governed by chaotic differential equations.

To sum up, the function $c(\cdot, \cdot)$ we use to build each emulator is as follows:

$$c(\boldsymbol{x}, \boldsymbol{x}') = k(\tilde{r}) + \nu\, \delta_{\boldsymbol{x},\boldsymbol{x}'}, \qquad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}, \tag{4.14}$$

where $k(\cdot)$ is the Matérn Kernel with parameter $\nu = 5/2$ (equation (1.58)), $\delta_{\cdot,\cdot}$ is the Kronecker-δ function, and

$$\tilde{r} = \sqrt{\sum_{h=1}^{p} \left(\frac{x_h - x'_h}{{d_h}^2} \right)^2}, \qquad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}. \tag{4.15}$$

The tilde over the argument $\tilde{r}$ of the kernel $k(\cdot)$ has been used to highlight the distinction from the fixed constant $r = 6$ used in the rest of this chapter, to denote the number of PCs used to approximate simulator outputs, equation (4.6). The positive parameters d_h in (4.15) are referred to as correlation lengths. As discussed in more detail in Section 1.5, their size is a measure of the strength of correlation between elements of the input space $\mathcal{P}$.

4.8. Estimation of Correlation Lengths and Nugget

In this section, we illustrate how to simultaneously estimate the unknown parameters appearing in the expression of the prior emulator covariance: the nugget term ν and the correlation lengths d_i. We note here that a common approach in the literature (Andrianakis and Challenor [2012], Bonceur et al. [2015]) is to maximise the integrated likelihood of the model, as reported in Berger et al. [2001]. For completeness and reference, we report and prove the formula in Appendix C. In our case, however, the method of maximising the integrated likelihood proved particularly unstable, due to a remarkable irregularity of the objective surface. We therefore implement a slightly different methodology, consisting in maximising the posterior density of the data, estimated via cross-validation (CV). We explain this in the following.

It is likely that the reader is already familiar with the CV methodology, but if unfamiliar they can find a brief description of the latter in the following box.

Cross Validation: A Few Words

Cross-validation (CV) is a statistical methodology, commonly used to assess the predictive ability of a statistical model. It applies when a data set of observations $\{(\boldsymbol{x_i}, y_i)\}_{i=1,\dots,n}$ is available, and a model to predict y_i as a function of $\boldsymbol{x_i}$ can be built based on the information provided in the data set. The idea behind CV is to consider the model built on a subset of the data only (the *training* set), and to compare the model predictions for the remaining data points (the *test set*) with the actual responses available in the original data set. In the special case where the training set consists of all pairs $(\boldsymbol{x_i}, y_i)$ except for one, and the test set consists of the left-out pair, the methodology is referred to as leave-one-out cross-validation (LOOCV). LOOCV is applied to all left-out pairs in turn, and a summarising measure of the goodness of the n predictions is returned.

In our case, for each $j \in \{1, \dots, r\}$, we want to build an emulator on a data set $\mathcal{D}$ consisting of n design points and n PC scores (equation (4.10)):

$$\mathcal{D} = \left\{ \left(\boldsymbol{x_i}, y_i^{(j)}\right) \right\}_{i=1,\dots,n} . \tag{4.16}$$

To simplify the notation, in the following we omit the fixed superscript j from the vector $\boldsymbol{y^{(j)}} \in \mathbb{R}^n$. To estimate the parameters ν and d_h to be used in the emulator construction, we proceed as follows. Consider the emulator calibrated on the data set $\mathcal{D}$ where the i^{th} entry of (4.16) has been left out. Let $\rho^{(i)}_{\boldsymbol{d},\nu}(\cdot)$ be the posterior density at the point $\boldsymbol{x_i}$ of this emulator, where nugget term ν and correlation lengths $\boldsymbol{d} = (d_1, d_2, d_3)$ are used. This is the probability density function of a univariate Student-t distribution, with mean and variance given by (2.96) and (2.97) under the position $\boldsymbol{x} = \boldsymbol{x'} := \boldsymbol{x_i}$. Hence, consider the function:

$$g(\boldsymbol{d}, \nu) = \prod_{i=1}^{n} \rho^{(i)}_{\boldsymbol{d},\nu}(y_i) . \tag{4.17}$$

Each term of the product is a measure of the emulator capability of predicting one

output that was not included in the calibration set. Hence, the function $g(\cdot, \cdot)$ represents a global measure of the goodness of the emulator predictions, when nugget ν and correlation lengths $\boldsymbol{d}$ are used.

The idea is to choose the parameters $\boldsymbol{d}$ and ν that maximise the function $g(\cdot, \cdot)$. We follow the main idea, with two modifications.

1. First, we impose $d_2 = d_3$, since the last two components of an input $\boldsymbol{x} \in \mathcal{P}$ have comparable scales ($x_2 = e\cos(\omega)$, $x_3 = e\sin(\omega)$). This has the further advantage of reducing the number of parameters to be estimated from four to three.

2. Second, we multiply the function $g\big(\ (d_1, d_2, d_2),\, \nu\ \big)$ by the product of three Gamma densities in each of the input parameters d_1, d_2, ν.

We now explain the reasons behind the adoption of the second choice. The surface obtained as graph of the function $g(\cdot, \cdot)$ is essentially flat when any of the correlation lengths d_i is close to zero: in this case, any two parameters $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}$ become essentially uncorrelated, regardless of the values of the other correlation lengths. Moreover, in such a case, the contribution coming from the nugget term becomes irrelevant (the matrix $\boldsymbol{A}$ of prior correlations is anyway a multiple of the identity). In addition to this, in-depth inspection of sections of the graph of $g(\cdot, \cdot)$ shows that for some PCs this has more than one local maxima, sometimes close to flat regions. A straight maximisation of $g(\cdot, \cdot)$ can therefore be numerically problematic to perform. Multiplying the function by a product of Gamma densities enables to regularise the surface, and to shift the new maximum towards a small region around the modes of the different Gamma densities. In this regard, recall from equation (2.8) that the expression of a Gamma density function with shape parameter $a > 0$ and rate parameter $\rho > 0$ is as follows:

$$h_{\Gamma}(x) = \frac{\rho^{\,a}}{\Gamma(a)}\, x^{a-1}\, e^{-\rho x}, \quad x > 0\,. \tag{4.18}$$

In our case, for each of the three variables d_1, d_2, ν, we choose the shape parameter a and the mode M of the distribution, and consequently compute the rate parameter ρ via the identity $\rho = (a-1)/M$. We choose the value $a = 4$ for all three parameters, and the modes $M = 3 \times 10^{-3}$ for d_1, $M = 2 \times 10^{-2}$ for d_2, $M = 0.5$ for ν. The mode

	d_1 MAP estimate ($\times 10^{-3}$)	d_2 MAP estimate ($\times 10^{-2}$)	ν MAP estimate
1st PC	4.01	3.96	0.962
2nd PC	2.23	0.21	0.611
3rd PC	4.29	3.74	0.956
4th PC	1.19	1.42	0.614
5th PC	5.22	2.51	0.580
6th PC	4.01	1.77	0.399

Table 4.4: Maximum a Posteriori (MAP) estimates of the parameters d_1, d_2, ν, needed in the specification of the emulator covariance function. Values are obtained by maximising the function $v(\cdot,\cdot,\cdot)$ in (4.19).

values for d_1 and d_2 represent around a fifth of the maximum distance between any two design points along the relevant axis; compare for example with Figure 4.8.

In conclusion, the procedure we use to estimate the parameters d_1, d_2 and ν is to maximise the following function:

$$v(d_1, d_2, \nu) = g\big(\boldsymbol{d}, \nu\big) \times h_{\Gamma}^{(1)}(d_1)\, h_{\Gamma}^{(2)}(d_2)\, h_{\Gamma}^{(3)}(\nu)\,, \tag{4.19}$$

where $\boldsymbol{d} = (d_1, d_2, d_2)$, and each Gamma density $h_{\Gamma}^{(l)}(\cdot)$ has shape parameter and mode specified above, $l = 1, 2, 3$. In practice, we maximise the logarithm of $v(\cdot,\cdot,\cdot)$, to avoid incurring in underflow or overflow numerical problems. The code implementing the maximisation is reported in Appendix E.2. The specific values of a and M are set within the function carrying out the emulation of the PC scores $f_j(\cdot)$, Appendix E.1.

Table 4.4 shows the estimated values of d_1, d_2, ν. With classical Bayesian terminology, we say that these are Maximum a Posteriori (MAP) estimates of the parameters of interest: we assign a prior density to the parameters (Gamma, in our case), multiply this by (a cross-validated estimate of) the likelihood function $g(\cdot)$, and select the maximum of the posterior density obtained this way.

4.9. Recombining the PC Scores

Once an emulator of each of the PC scores $f_j(\cdot)$ is built, formula (4.6) allows to predict the SST field $\boldsymbol{\gamma}(\boldsymbol{x})$ corresponding to any input $\boldsymbol{x} \in \mathcal{P}$. We recall the formula here:

$$\widetilde{\boldsymbol{\gamma}}(\boldsymbol{x}) \;=\; \bar{\boldsymbol{\gamma}} + \sum_{j=1}^{r} f_j(\boldsymbol{x}) \boldsymbol{V_j} \,. \tag{4.20}$$

We use $r = 6$. For simplicity of notation, consider each $f_j(\cdot)$ in equation (4.20) to represent directly the emulated coefficient, rather than the original, unknown, scalar field. Each $f_j(\cdot)$ is therefore a real-valued stochastic process with parameter space $\mathcal{P}$. Hence the emulator prediction $\widetilde{\boldsymbol{\gamma}}(\cdot)$ is itself a stochastic process, valued in $\mathbb{R}^s$. We use the index c to refer to a general component of $\widetilde{\boldsymbol{\gamma}}(\cdot)$, and denote this by $\widetilde{\gamma}^{(c)}(\cdot)$. Similarly, component c of $\boldsymbol{V_j} \in \mathbb{R}^s$ will be denoted by $V_j^{(c)}$. The choice of the letter c is meant to remind of the fact that each such component corresponds to a cell of the simulator output grid.

For any $\boldsymbol{x} \in \mathcal{P}$, the mean of component c of $\widetilde{\boldsymbol{\gamma}}(\boldsymbol{x})$ reads as follows:

$$\mathbb{E}\big[\widetilde{\gamma}^{(c)}(\boldsymbol{x})\big] \;=\; \bar{\gamma}^{(c)} + \sum_{j=1}^{r} \mathbb{E}\big[f_j(\boldsymbol{x})\big]\, V_j^{(c)} \,. \tag{4.21}$$

For any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{P}$, the 2×2 covariance matrix of the vector $\big(\widetilde{\gamma}^{(c)}(\boldsymbol{x}), \widetilde{\gamma}^{(c)}(\boldsymbol{x}')\big)$ reads instead as follows:

$$\mathbf{Cov}_{\boldsymbol{x},\boldsymbol{x}'}\big[\widetilde{\gamma}^{(c)}(\cdot)\big] = \sum_{j=1}^{r} \Big(V_j^{(c)}\Big)^2 \mathbf{Cov}_{\boldsymbol{x},\boldsymbol{x}'}\big[f_j(\cdot)\big] \,. \tag{4.22}$$

We have denoted by $\mathbf{Cov}_{\boldsymbol{x},\boldsymbol{x}'}[f_j(\cdot)]$ the 2×2 covariance matrix of the random vector with components $f_j(\boldsymbol{x})$ and $f_j(\boldsymbol{x}')$. Notice that formula (4.22) assumes that the coefficients $f_{j_1}(\cdot)$ and $f_{j_2}(\cdot)$ are uncorrelated if $j_1 \neq j_2$: this is a reasonable assumption given the orthogonality of the PCs $\boldsymbol{V_{j_1}}$ and $\boldsymbol{V_{j_2}}$.

Formula (4.22) only accounts for the variability displayed by the first r PCs, since the remaining components are not involved in reconstructing the emulator response. Nonetheless, as for example in Bonceur et al. [2015], we can account for the additional

variability that their inclusion would introduce, by adding a term to emulator covariance. That is, we replace formula (4.22) by the following:

$$\mathbf{Cov}_{\boldsymbol{x},\boldsymbol{x}'}\left[\widetilde{\gamma}^{(c)}(\cdot)\right] = \sum_{j=1}^{r}\left(V_j^{(c)}\right)^2 \mathbf{Cov}_{\boldsymbol{x},\boldsymbol{x}'}\left[f_j(\cdot)\right] + \sum_{j=r+1}^{n-1}\left(V_j^{(c)}\right)^2 \sigma_j^2 \boldsymbol{I_2}\,, \tag{4.23}$$

where σ_j is the standard deviation associated to the j^{th} PC (equation (4.5)), and $\boldsymbol{I_2}$ is the identity matrix of order 2. Together with (4.21), formula (4.23) and its straightforward generalisation to more than two inputs in $\mathcal{P}$ is of crucial importance to sample trajectories from the emulator.

Finally, we notice that equations (4.21) and (4.23) can be recognised to be the mean and covariance functions of the following process:

$$\widetilde{\gamma}^{(c)}(\boldsymbol{x}) = \bar{\gamma}^{(c)} + \sum_{j=1}^{r} f_j(\boldsymbol{x})\, V_j^{(c)} + \sum_{j=r+1}^{n-1} \sigma_j \varepsilon_j(\boldsymbol{x})\, V_j^{(c)}\,, \quad \boldsymbol{x} \in \mathcal{P}. \tag{4.24}$$

The $\varepsilon_j(\cdot)$ are independent Gaussian processes whose finite-dimensional distributions are multivariate normal with zero mean and identity covariance matrix. Moreover, each $\varepsilon_j(\cdot)$ is independent of any of the $f_k(\cdot)$, for $k = 1, \ldots, r$ and $j = r+1, \ldots, n-1$.

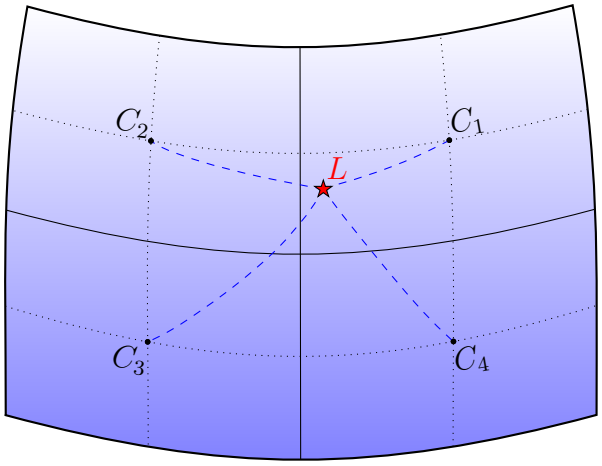
Loosely speaking, formula (4.24) can be interpreted as follows: the expressions of the mean and covariance that we use ((4.21) and (4.23)) reflect the original decomposition (4.4), where only the first r PC scores are emulated, while the remaining ones are replaced by Gaussian noise with the proper variance.

4.9.1. Prediction for a General Location

Formulas (4.21) and (4.23) provide the emulator mean and covariance, at any of the p grid cells constituting the simulator output. In order to emulate the SST at a general location L, not necessarily falling on the output grid, we can use the same formulas where we replace the value of the j^{th} PC at the c^{th} grid cell, $V_j^{(c)}$, with an estimated value of the j^{th} PC at the location of interest, $V_j^{(L)}$. We obtain the latter by interpolating among neighbouring cells, as explained below.

The location L belongs to one of the s grid cells of the simulator output. Consider the

other three closest grid cells to the location, so that the total four of them surround the point forming an approximate spherical "square" around it.



We compute the value $V_j^{(L)}$ of the j^{th} PC at the location of interest as weighted average of the PC values at the four grid cells, where the weights are inversely proportional to the geodesic distance between the location L of interest and the centre C_k of each grid cell. If any of the four cells corresponds to land in the simulator, then its value is not included in the average. The positive weights are normalised so that they sum up to one.

The geodesic distance between two points A and B of a sphere is the length of the shortest path lying on the sphere, which connects the two points. This path is found by considering the great circle[13] going through the two points, and in particular the shorter of the two arcs in which the two points divide it. The formula is as follows. Suppose for simplicity that the two points, A and B, belong to the unit sphere S^2. Let (θ_A, ϕ_A) and (θ_B, ϕ_B) be the latitude and longitude coordinates of each of the two points:

$$\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \quad \phi \in [0, 2\pi]. \tag{4.25}$$

Then, the geodesic distance $G(A, B)$ between the two points can be computed via the

[13] In a sphere, a great circle is obtained as intersection between the sphere and a plane going through its centre. Great circles have therefore the same radius as the sphere.

identity:

$$\cos\big[G(A,B)\big] \;=\; \sin(\theta_A)\sin(\theta_B) + \cos(\theta_A)\cos(\theta_B)\cos(\phi_A - \phi_B)\,. \tag{4.26}$$

Notice indeed that $G(A,B) \in [0,\pi]$ since the sphere has radius one; hence the cosine on the LHS can be inverted.

In the Matlab Appendix E.4, we show the code that, given the coordinates of the location L of interest, identifies the nearby cells (only sea cells if needed) and computes the weights inversely proportional to the relevant geodesic distances.

4.9.2. Sampling Trajectories from the Emulator

In this section we describe how to extract a multivariate sample from the emulator. At a general sequence of input parameters $\mathcal{S} = \big\{\widetilde{\boldsymbol{x}}_{\mathbf{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\boldsymbol{k}}\big\} \subseteq \mathcal{P}$, the emulator at location L is a k-dimensional random vector, which we denote by $\boldsymbol{\gamma^{(L)}(\mathcal{S})}$:

$$\boldsymbol{\gamma^{(L)}(\mathcal{S})} := \begin{pmatrix} \gamma^{(L)}(\widetilde{\boldsymbol{x}}_{\mathbf{1}}) \\ \vdots \\ \gamma^{(L)}(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}) \end{pmatrix} \in \mathbb{R}^k. \tag{4.27}$$

Its distribution is multivariate t-Student (Definition 2.5.1):

$$\boldsymbol{\gamma^{(L)}(\mathcal{S})} \;\sim\; t_\nu\Big(\boldsymbol{\mu}_{\boldsymbol{\mathcal{S}}}^{\boldsymbol{(L)}},\, \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{S}}}^{\boldsymbol{(L)}}\Big). \tag{4.28}$$

The number of degrees of freedom is $\nu = n - q$, where $q = 4$ in our case. The mean $\boldsymbol{\mu}_{\boldsymbol{\mathcal{S}}}^{\boldsymbol{(L)}}$ is immediately derived by extending equation (4.21) to case of k inputs and replacing the quantity $V_i^{(j)}$ by $V_L^{(j)}$:

$$\boldsymbol{\mu}_{\boldsymbol{\mathcal{S}}}^{\boldsymbol{(L)}} \;=\; \bar{\gamma}^{(L)}\mathbf{1}_{\boldsymbol{k}} \;+\; \sum_{j=1}^{r} \mathbb{E}\big[\,\boldsymbol{f_j(\mathcal{S})}\big]\, V_j^{(L)} \;\in \mathbb{R}^k\,. \tag{4.29}$$

We have denoted by $\boldsymbol{f_j(\mathcal{S})}$ the vector with components $f_j\big(\widetilde{\boldsymbol{x}}_{\mathbf{1}}\big), \ldots, f_j\big(\widetilde{\boldsymbol{x}}_{\boldsymbol{k}}\big)$. The symbol $\mathbf{1}_{\boldsymbol{k}} \in \mathbb{R}^k$ denotes the vector of all ones.

The kernel matrix $\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{S}}}^{\boldsymbol{(L)}}$ in (4.28) is obtained from the covariance matrix of $\boldsymbol{\gamma^{(L)}(\mathcal{S})}$

by the relationship (recall (2.49.b)):

$$\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{S}}}^{(\boldsymbol{L})} \;=\; \frac{\nu-2}{\nu}\,\mathbf{Cov}\big[\boldsymbol{\gamma}^{(\boldsymbol{L})}(\boldsymbol{\mathcal{S}})\big] \;\in \mathbb{R}^{k\times k}\,. \tag{4.30}$$

In turn, we notice that the expression of the covariance matrix is immediately derived by extending (4.23) to the case of k inputs:

$$\mathbf{Cov}\big[\boldsymbol{\gamma}^{(\boldsymbol{L})}(\boldsymbol{\mathcal{S}})\big] \;=\; \sum_{j=1}^{r}\Big(V_j^{(L)}\Big)^2\mathbf{Cov}\big[\boldsymbol{f_j}(\boldsymbol{\mathcal{S}})\big] \;+\; \sum_{j=r+1}^{n-1}\Big(V_j^{(L)}\Big)^2\sigma_j^2\boldsymbol{I_k} \;\in \mathbb{R}^{k\times k}. \tag{4.31}$$

Most mathematical and statistical softwares are able to sample from a multivariate t-Student distribution; however, definitions may not be equivalent from one software to the other, or may be given in terms of different parameters. Hence, we briefly describe here a simple and numerically stable way to sample from

$$\boldsymbol{Y} \sim t_\nu\big(\boldsymbol{\mu},\boldsymbol{\Sigma}\big)\,, \tag{4.32}$$

which only requires the ability to generate independent samples of the $\chi^2(\nu)$ and $N(0,1)$ distributions: this allows to sample from our emulator, with mean and kernel as in (4.29), (4.30).

Given our Definition 2.5.1, from (4.32) we can write:

$$\boldsymbol{Y} = \boldsymbol{\mu} + \sqrt{\frac{\nu}{V}}\,\boldsymbol{X} \;\in \mathbb{R}^k\,, \tag{4.33}$$

where $\boldsymbol{X} \sim N(\boldsymbol{0},\boldsymbol{\Sigma})$ and $V \sim \chi^2(\nu)$ are independent of each other. Since it is normally distributed, the random vector $\boldsymbol{X}$ can be written as linear transformation of a multivariate standard normal vector $\boldsymbol{Z} \sim N(\boldsymbol{0},\boldsymbol{I_k})$. Specifically, since $\boldsymbol{\Sigma}$ is symmetric and positive definite, we can consider its Cholesky decomposition:

$$\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T, \tag{4.34}$$

where $\boldsymbol{A}$ is lower triangular. The vector $\boldsymbol{X}$ can then be written as:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z}\,. \tag{4.35}$$

Notice indeed that from (4.35) it follows $\mathbf{Cov}[\boldsymbol{X}] = \boldsymbol{A}\,\mathbf{Cov}[\boldsymbol{Z}]\boldsymbol{A}^T$. Substituting (4.35) into (4.33), we get

$$\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{T}\,, \tag{4.36}$$

where $\boldsymbol{T} = \sqrt{\nu/V}\,\boldsymbol{Z}$ is a k-dimensional Student-t vector with zero mean and identity kernel matrix. A sample of such $\boldsymbol{T}$ can thus be obtained upon division of k independent $N(0,1)$ samples by the square root of a $\chi^2(\nu)$ sample (independent of the previous normal ones), and by rescaling the ratio by a factor $\sqrt{\nu}$.

Probabilistic Observation/Remark

Given what just pointed out, it is clear that the components of a random vector $\boldsymbol{T} \sim t_\nu(\boldsymbol{0}, \boldsymbol{I_k})$ are not independent of each other: the random factor $V^{-1/2}$ is common to all of k of them. However, they are uncorrelated: this provides a per se interesting example in probability, of a random vector with uncorrelated but non-independent components.

The fact that the components of a $\boldsymbol{T} \sim t_\nu(\boldsymbol{0}, \boldsymbol{I_k})$ are uncorrelated may be deduced, for example, from the following fact, trivial to check: the covariance between two random variables of the form $T_1 = Z_1 W$ and $T_2 = Z_2 W$ is zero, for independent Z_1, Z_2 and W with $\mathbb{E}[Z_i] = 0$.

The above procedure allows to sample from the emulator distribution at any finite subset $\mathcal{S} \subseteq \mathcal{P}$ of input parameters. In practice, it is convenient for us to look at the emulated SST as a function of time: we fix a sequence of past times, use the online interface available here to find the set $\mathcal{S}$ of corresponding orbital parameters, and employ the procedure above to sample from a t-distributed random vector with mean (4.29), kernel matrix (4.30), and $n - q$ degrees of freedom.

An example of emulator trajectory at one of the marine sites where data is available (Section 4.2) is shown in the middle panel of Figure 4.10. For illustration purposes, the "Gaussian noise" component of the trajectory is left out (last term of equation (4.24)). In addition, we recall that a nugget term has been used to build the emulated PC scores $f_j(\cdot)$: in order to display a continuous trajectory, only the continuous component of the emulated $f_j(\cdot)$ has been used – see the decomposition of Theorem 2.8.3 and comments

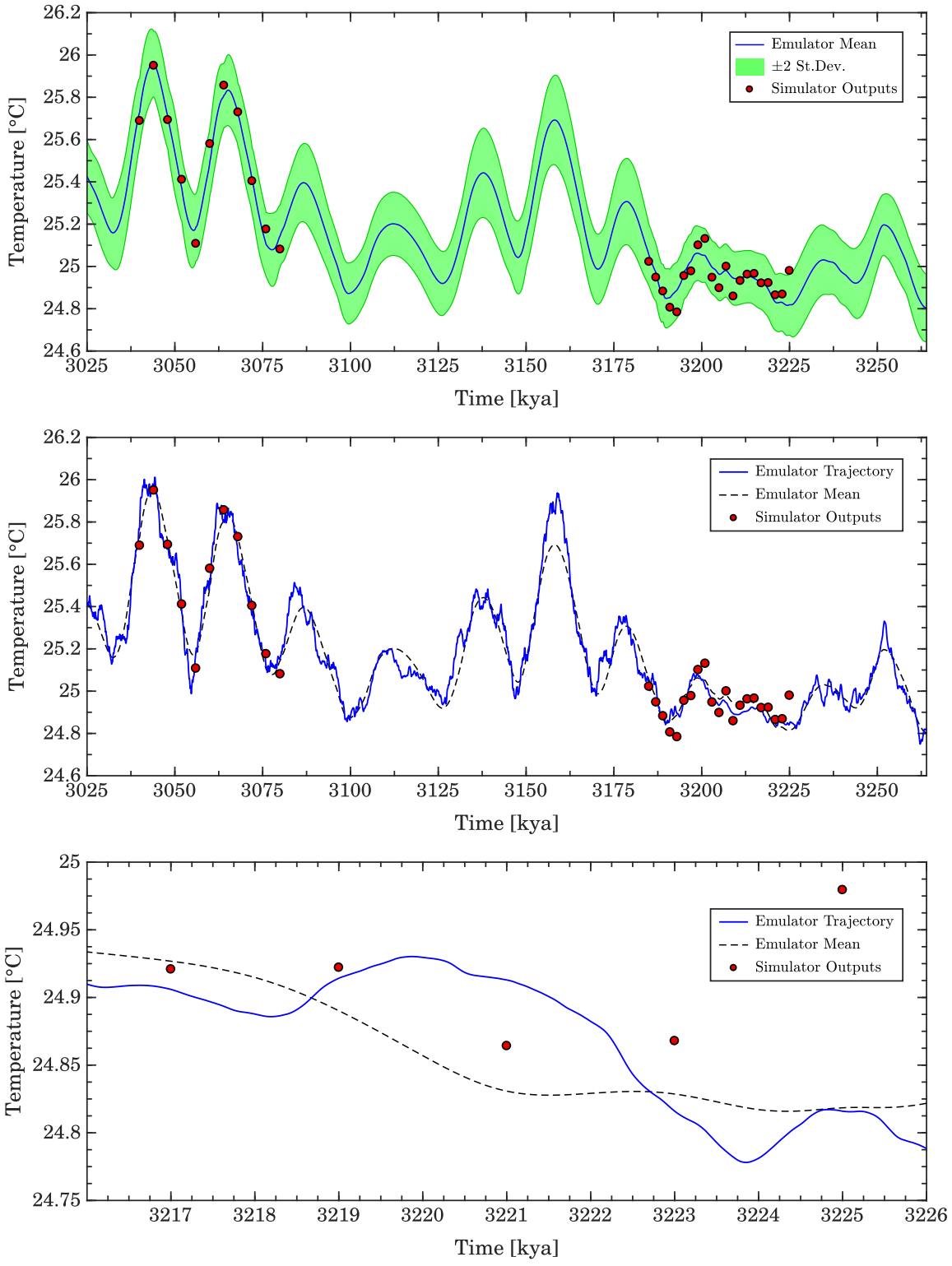


Figure 4.10: Emulator predictions over the PRISM3D time slab at one of the marine location where data are available (lat: 19.74°S, lon: 10.52°E; off the south-west African coast, in front of Namibia). Top panel: emulator mean and standard deviation, together with simulator outputs at the design points. Middle panel: one sample trajectory, continuous since the noise part (last term in equation (4.24)) has been left out. Bottom panel: zoom of a trajectory, to appreciate level of regularity.

thereafter (formulas (2.119), (2.120)). From the theory presented in Subsection 1.4.2, we know that the emulator trajectories are C^2 but not C^3. To appreciate this visually, the bottom panel of Figure 4.10 shows the zoom of a trajectory over a relatively small interval of time. Note, instead, that the emulator posterior mean (blue line in the top panel of Figure 4.10) is smoother than the trajectories. Indeed, from equation (2.96) we see that the regularity of the posterior mean function is determined by the regularity of the functions $\boldsymbol{h}(\cdot)$ and $\boldsymbol{t}(\cdot)$: in our case, $\boldsymbol{h}(\boldsymbol{x})$ is linear in $\boldsymbol{x}$, and the components of $\boldsymbol{t}(\boldsymbol{x})$ (*i.e*, $c(\boldsymbol{x}, \boldsymbol{x_i})$ for $i = 1, \ldots, n$) are C^∞ when $\boldsymbol{x} \neq \boldsymbol{x_i}$. The predicted mean is therefore C^∞ between design points.

4.10. Data-Model Comparison (DMC)

In this section we describe the procedure we use to compare the geological archive of reconstructed marine temperatures (Section 4.2) to the emulator SST predictions. The comparison is carried out over the whole time interval that the fossils used in the geological temperature reconstruction date back to (PRISM3D time slab: 3,025–3,264 kya). As pointed out in Section 4.1, the use of the emulator allows to account for the effects of varying orbital forcing during the interval; moreover, by emulating site by site, we are able to account for potential asyncronicity in the warming at different locations.

As detailed in Section 4.2, at each marine site depicted in Figure 4.1, the geological archive provides a warm peak average (WPA) estimate, the number N_p of peaks used to obtain the WPA, and the number N_s of samples constituting the original time series before peaks were extracted. In order to obtain estimates of average warm peaks from the emulator, at each location we proceed as follows:

1. Uniformly at random, we sample N_s times between 3,025 and 3,264 kya.
2. We generate a sample $\boldsymbol{z} = (z_1, \ldots, z_{N_s})$ of emulated SST from the distribution of the emulator at those times, as described in Subsection 4.9.2.
3. We then extract the peaks of the sample: a value z_i is considered a peak if greater than both z_{i-1} and z_{i+1}. Further specifications are provided below.

4. If the number of peaks equals N_p, the peak average is stored. Otherwise, we repeat steps 1–3 till the extracted number of peaks is N_p, and store the peak average.

For each location, we repeat the previous four steps till $N = 1{,}000$ WPA estimates from the emulator are obtained. Finally, we subtract from these the SST value, at the site of interest, obtained from a control simulation run with pre-industrial boundary conditions. This step is important to help remove potential biases in the simulation process. Similarly, at each site, from the geologically reconstructed mid-Pliocene WPA we subtract observed SST modern temperature. This last value is provided within the PRISM3D data set; it is derived from either Reynolds and Smith [1995] or Levitus [1982], according to the site of interest.

The sequence of steps 1–4 was adopted in order to reproduce as closely as possibly the methodology employed during the data collection process (Section 4.2). We need to make some remarks about the latter, which allow us to provide further details on our own procedure.

When peaks were extracted from a time series of marine temperature proxies, the first and last element of the sequence were not considered peaks. We therefore adopt the same convention in our procedure, specifically in step 3. This implies that the condition $N_p < N_s/2$ holds between the number of samples and the number of peaks at any site. For some of the data set locations, however, a number of peaks and total samples with $N_p \geq N_s/2$ were provided. Upon further inspection[14], this was revealed to be the consequence of "identical" consecutive estimates in the time series, in which case both elements had the potential to be considered peaks. We therefore adapt our procedure to this feature, by allowing two consecutive elements z_i and z_{i+1} of an emulator sample to be considered the same, if their difference is less than a given tolerance t. In such a case, z_i is considered a peak if greater than z_{i-1}; similarly, z_{i+1} is considered a peak if greater than z_{i+2}. We set the tolerance to $t = 0.01$°C, in consideration of the fact that the original marine proxies were analysed at a resolution of two decimal places.

[14] I (the author) would like to acknowledge here the effort of Prof. Harry Dowsett, whom I thank once again, for recovering and personally scrutinising the original time series of the "problematic" sites.

By adopting the above procedure, we are able to generate WPA emulator estimates for 42 of the 51 sites provided in the data set. The remaining sites show values of N_p and N_s that are practically unfeasible to match: the ratio N_p/N_s is either too close to zero, or very close to a half despite a large N_s (a long sequence with approximately a peak every other element), or even larger than a half. Although reasons behind some of these rare occurrences were provided to the author, it remains impossible for the emulator to reproduce such sequences: hence, we carry out the comparison for the 42 sites where the comparison is possible.

We can now describe the way the comparison is carried out. For each site, our procedure returns N WPA estimates obtained from the emulator: these have a natural interpretation as N independent samples of the emulator WPA distribution at the site, conditioned on having observed N_p peaks out of N_s samples. Let m and s be the empirical mean and standard deviation of the sample, respectively, and denote with I the interval $[m-2s, m+2s]$. For convenience, we define $a := m-2s$ and $b := m+2s$, hence $I = [a, b]$. We assess the agreement between the collection of N emulator WPA samples and the geological estimate z in the marine data set[15], by computing the signed distance between the interval I defined above and the value z. This distance is defined as follows:

$$\mathrm{dist}(I, z) = \begin{cases} a - z & \text{if } z < a \\ 0 & \text{if } z \in I \\ b - z & \text{if } z > b \end{cases}, \quad I = [a, b],\ z \in \mathbb{R}. \tag{4.37}$$

The signed distance in (4.37) is positive if the emulator WPA estimates are greater than the proxy data z (warm bias of the emulator) and it is negative if the emulator WPA estimates are lower than z (cold emulator bias). We note here that the emulator samples at the different sites show an approximately normal distribution, hence the interval I defined above has a loose interpretation as a 95% confidence interval of the emulator WPA distribution.

[15] As explained earlier, the appropriate PI values are preliminarily subtracted on both the emulator and the proxy data.

4.11. Results

By carrying out the procedure described in Section 4.10, at each site we obtain a measure of the discrepancy between emulator predictions and geological record. Figure 4.11 shows the results, commented below.

Around a quarter of the sites (11 out of 42) display a data-model mismatch in modulus less than 0.5°C according to the measure in (4.37). These are shown as white circles in Figure 4.11. Further inspection shows that for three of them the distance (4.37) is zero, while the remaining eight are equally split into sites showing either a cold small bias or a warm small bias of the simulator with respect to geological records. The data-model comparison (DMC) for the remaining locations shows a remarkable pattern. With only few exceptions, low-latitude sites reveal a warm simulator bias, while high-latitude sites reveal a cold simulator bias. This is particularly evident in the Northern Hemisphere, where most of the marine sites of the data set lie. In this region, by taking as convenient reference the parallel located at 40°N latitude, we see the following: 21 of the 23 sites south of the parallel show warm (or zero) simulator bias; 6 of the 7 sites north of the parallel show cold (or zero) simulator bias. An analogous, symmetrical, pattern is revealed in the Southern Hemisphere, although the number of sites is here more limited.

It is now of interest to compare our DMC procedure, built via the use of the emulator and accounting for the orbital variability characterising the mid-Pliocene, to the DMC where proxies are related to the outputs of single mid-Pliocene simulations. In particular, we select the peaks of the two interglacial events K1 and KM5c, happening at 3,060 and 3,205 kya respectively (compare to description in Subsection 4.5.1). In each of the two cases, we compute the difference between the snapshot simulation and the geological proxy, at the 42 marine sites (as usual, after subtracting the appropriate PI values from each source). Figure 4.12 allows to graphically compare the results.

While all three plots approximately reveal the same geographical pattern previously recognised (high-latitude cold bias and low-latitude warm bias for the simulator), differences in magnitude between the DMC involving the emulator and the DMC involving each of the snapshot simulations are often significant, especially in particular geographical areas. We now summarise some results, differentiating between the two

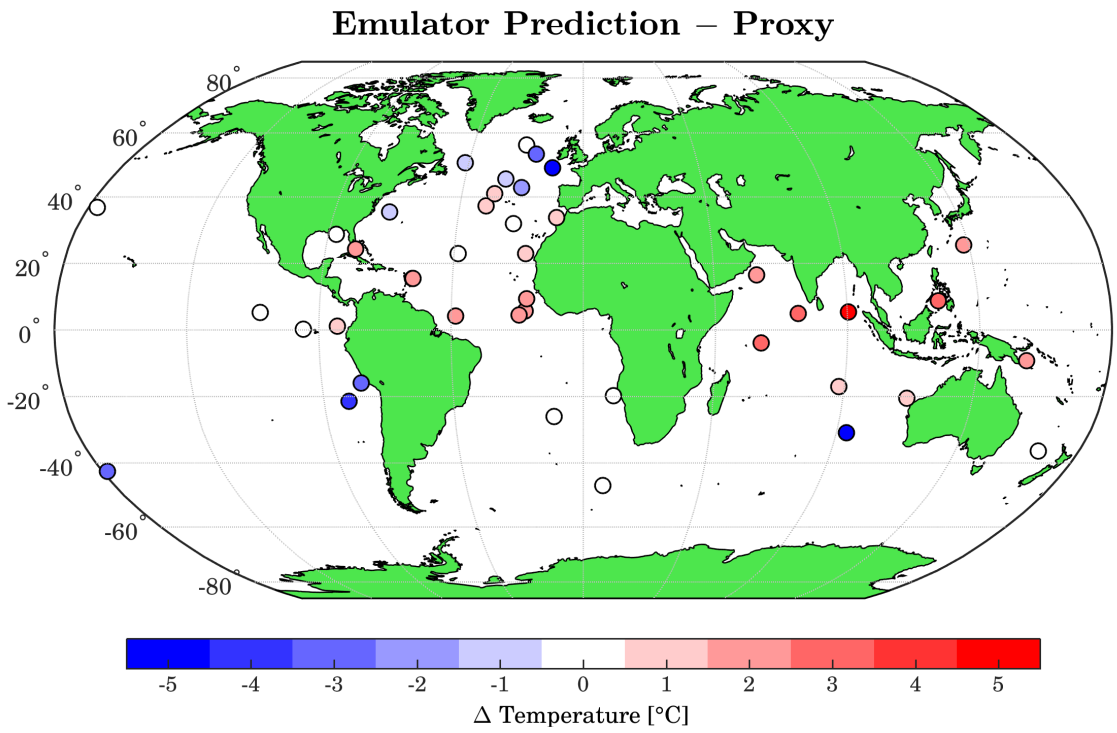


Figure 4.11: Each circle locates a marine site. The plot aims at comparing, at each of these sites, the set of $N = 1,000$ WPA estimates derived from the emulator to the single WPA estimate derived from geological records. The signed measure (4.37) is used to quantify the difference. Red denotes warm emulator bias, blue denotes cold emulator bias.

cases where the emulator shows a cold, or warm, bias.

Sites with cold or null emulator bias

There are 17 such sites, mostly located at high latitudes.

- For all of them, the DMC carried out via the 3205 simulation shows a stronger (still cold) bias than the one carried out via the emulator. In Table 4.5 we report the percentage of improvement from the 3205-based to the emulator-based DMC, alongside the absolute difference of the two: the percentage of improvement is computed as $(1 - d_{\text{Emul}}/d_{3205}) \times 100$, where d_{Emul} is the mismatch obtained from our DMC (equation (4.37)), and d_{3205} is the one obtained by using the 3205 peak.

- For the same sites, differences between the emulator DMC and the one carried out with simulated temperatures from the 3060 peak do not show significant patterns (last column of Table 4.5).

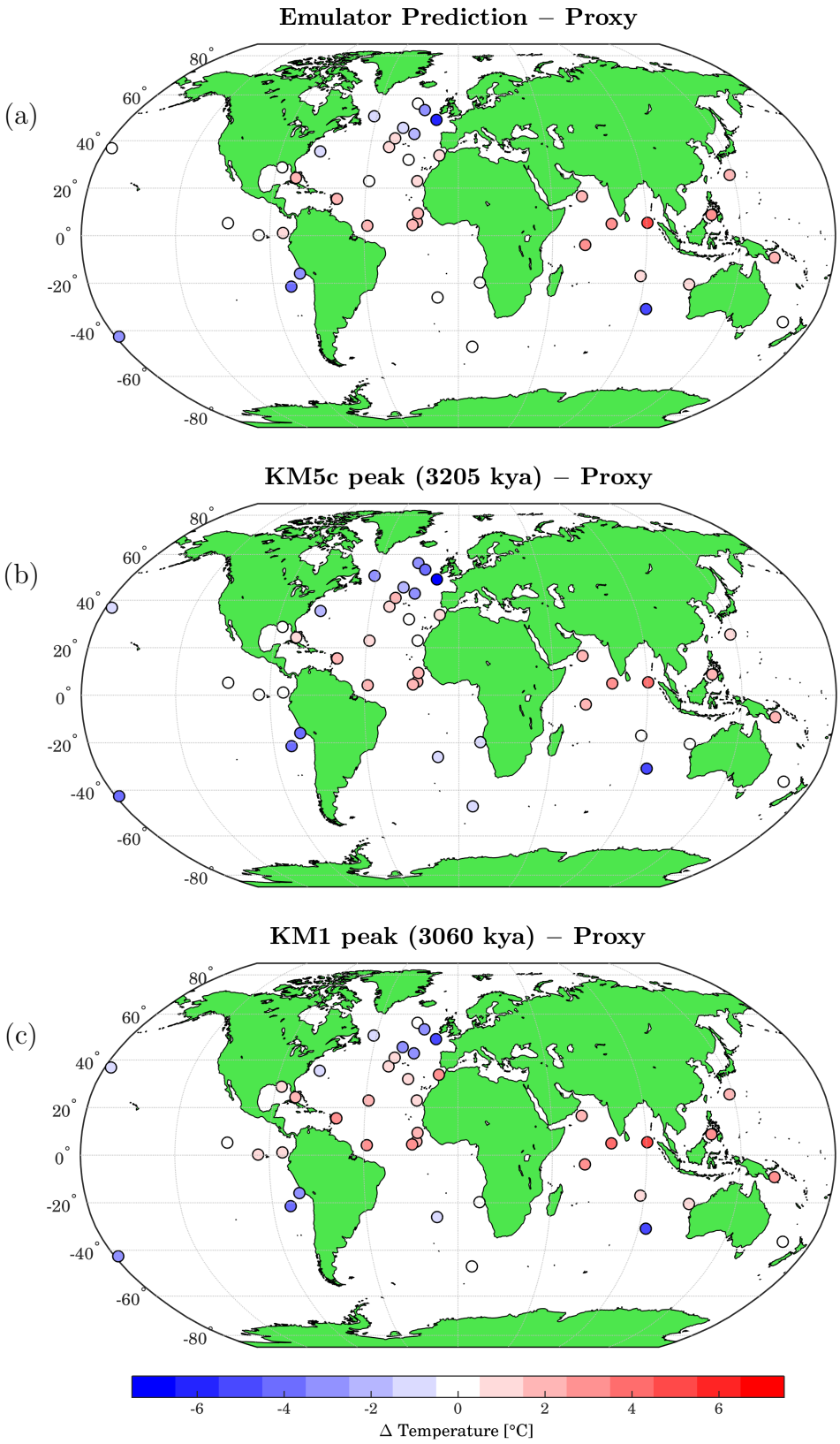


Figure 4.12: Following from Fig. 4.11, plot of the difference between: 1) WPA estimates obtained from (a) the emulator, (b) the KM5c interglacial peak simulation (3,205 kya), (c) the KM1 interglacial peak simulation (3,060 kya); and 2) WPA estimates from geological records (proxies). Panel (a) shows the same plot as Fig. 4.11, and is here reported to ease comparison with (b) and (c). Notice, in general, darker blue colours of panel (b) and darker red colours of panel (c), when compared to (a). See also Table 4.5 and Table 4.6.

Latitude	Longitude	Improvement on 3205 (Absolute Difference)	Improvement on 3060 (Absolute Difference)
56.04° N	23.33° W	100% (3.29°C)	100% (−0.47°C)
53.22° N	18.89° W	31% (1.36°C)	−15% (−0.39°C)
50.42° N	46.37° W	63% (2.18°C)	−46% (−0.40°C)
48.85° N	12.00° W	20% (1.37°C)	−8% (−0.40°C)
45.51° N	29.48° W	32% (0.63°C)	59% (2.01°C)
42.84° N	23.09° W	23% (0.58°C)	36% (1.08°C)
36.87° N	176.90° W	65% (0.62°C)	66% (0.66°C)
35.49° N	70.03° W	49% (1.00°C)	4% (0.04°C)
5.28° N	110.07° W	100% (0.43°C)	100% (−0.03°C)
16.00° S	76.37° W	28% (1.01°C)	5% (0.14°C)
19.74° S	10.52° E	89% (0.86°C)	64% (0.20°C)
21.37° S	81.43° W	18% (0.78°C)	16% (0.72°C)
26.07° S	10.27° W	64% (0.56°C)	44% (0.24°C)
30.93° S	93.57° E	8% (0.44°C)	3% (0.13°C)
36.47° S	165.44° E	100% (0.29°C)	100% (−0.11°C)
42.55° S	178.17° W	17% (0.72°C)	−3% (0.10°C)
46.88° S	7.42° E	88% (0.53°C)	35% (0.04°C)

Table 4.5: In the column on the right of the vertical line, the table shows the improvement from the DMC based on the KM5c experiment (3205 kya) to the DMC based on the emulator WPA predictions. Only sites showing cold emulator bias are reported here (signed distance in (4.37) negative). Both percentage values (see text for details) and absolute differences (emulator DMC minus 3205 DMC) are reported. All differences are positive, showing an improvement of the emulator on all reported sites. By using the emulator, the data-model mismatch seems generally reduced also with respect to the 3060-based DMC (values in the last column), although no strong pattern as in the 3205 column emerges.

Latitude	Longitude	Improvement on 3060 (Absolute Difference)	Improvement on 3205 (Absolute Difference)
41.00° N	32.96° W	−75% (+0.60°C)	35% (−0.74°C)
37.34° N	35.50° W	14% (−0.17°C)	−9% (0.08°C)
33.80° N	9.60° W	67% (−1.70°C)	23% (−0.25°C)
32.03° N	24.87° W	87% (−0.75°C)	156% (0.31°C)
28.83° N	87.17° W	59% (−0.50°C)	−165% (0.21°C)
25.52° N	133.20° E	10% (−0.16°C)	−16% (0.21°C)
24.40° N	79.46° W	28% (−0.63°C)	−23% (0.30°C)
23.00° N	20.00° W	39% (−0.37°C)	−31% (0.14°C)
22.90° N	43.50° W	76% (−1.42°C)	46% (−0.39°C)
16.62° N	59.80° E	2% (−0.04°C)	−11% (0.19°C)
15.52° N	58.72° W	42% (−1.27°C)	14% (−0.29°C)
15.50° N	58.50° W	39% (−1.23°C)	6% (−0.13°C)
9.45° N	19.39° W	10% (−0.21°C)	−15% (0.25°C)
8.78° N	121.29° E	14% (−0.41°C)	−9% (0.21°C)
5.68° N	19.85° W	20% (−0.46°C)	−4% (0.07°C)
5.38° N	90.37° E	10% (−0.49°C)	−8% (0.34°C)
4.93° N	73.28° E	4% (−0.16°C)	−7% (0.22°C)
4.55° N	21.90° W	13% (−0.35°C)	−11% (0.24°C)
4.20° N	43.49° W	15% (−0.40°C)	−12% (0.25°C)
1.20° N	83.74° W	54% (−0.73°C)	−79% (0.27°C)
0.18° N	95.32° W	88% (−0.92°C)	139% (0.46°C)
3.92° S	60.55° E	10% (−0.30°C)	−12% (0.29°C)
9.18° S	151.57° E	17% (−0.48°C)	−9% (0.18°C)
17.02° S	88.18° E	44% (−0.50°C)	−63% (0.24°C)
20.59° S	112.21° E	52% (−0.76°C)	−92% (0.34°C)

Table 4.6: Improvement from the 3060- to the emulator- based DMC, for sites showing warm emulator bias. Same specifications of Table 4.5 hold. With the only exception of the northern-most site, the emulator DMC reduces the mismatch. No systematic improvement can instead be recognised with respect to the 3205-based DMC (last column).

Sites with warm emulator bias

There are 25 such sites, all located within the latitude band between 21°S and 41°N.

- For all of them but one, the DMC carried out via the 3060 simulation shows a stronger (still warm) bias than the one carried out via the emulator. Specific numbers are reported in Table 4.6. The exceptional site is the northern most one of the set.
- For the same sites, difference between the emulator-based and the 3205-based DMCs are mostly negligible and do not seem to reveal significant patterns: difference values are in modulus less than 0.5°C, with the only exception of the same northern most site mentioned in the previous point (for which the difference is of 0.74°C).

The two points just mentioned suggest that the accounted-for orbital variability in our DMC is generally able to reduce the mismatch with respect to the DMC built on single shapshot simulations, even when these are the ones corresponding to the two warm peaks of the K1 and KM5c interglacials. Moreover, it can be deduced from the points above that simulated SST around the 3060 interglacial peak is generally higher than the simulated SST around the 3205 peak, although this is not true for all the locations.

4.12. Conclusions

In this chapter we have tackled the problem of DMC during the mid-Pliocene, analysing it within the statistical framework provided by GP emulation. While the latter has been used in diverse climate reconstruction problems, as well as in other DMC settings, the present work represents the first instance in which the emulation setting is employed to analyse the mismatch between mid-Pliocene climate simulations and geological records. The contribution of GP emulation is relevant to the problem, in that allows to account for the orbitally-induced changes in simulated temperature, which Prescott et al. [2014] have shown to be substantial.

By comparing the estimated WPAs from geological records to random samples drawn from a number of emulated SST trajectories, we are able to match, as close as we can, the way geological estimates are derived. We remark, indeed, that precise times are not associated to the single elements of the original geological time series, from which a WPA is extracted. In light of this, comparisons to single snapshot simulations may be inappropriate, carrying unavoidable biases. We are instead able to automatically account for the time uncertainty associated with the data, by sampling at random times (within the time interval of relevance) from the emulator SST distribution. By carrying out this procedure site by site, we are also able to account for potentially asynchronous warming between sites. The results highlight that the HadCM3 climate simulator typically shows a cold bias at high latitudes with respect to geological record, while a warm bias is displayed at low-latitudes. The mismatch is however generally reduced with respect to the case where the comparison is performed based on the output of a single climate simulation.

We would like to conclude by looking at what we think represents the next natural research step within the field. The recent work Dowsett et al. [2019] provides, for different marine sites, reasonably accurate estimates of past times within the mid-Pliocene, corresponding to which a time series of reconstructed temperatures is extracted. The inclusion of time estimates of course represents a major improvement with respect to the PRISM3D data set, and could be easily merged within our DMC procedure by sampling only at the relevant times from the emulator distribution. Such detailed information comes currently at the price of having only few (eight) sites where the data has been processed and stored. All of these are located in the North Atlantic region. Nonetheless, we believe that the availability of this information will allow to shed further light on the mid-Pliocene DMC. Once again, the use of GP emulation seems the natural way to incorporate the additional time information in the DMC framework, further allowing to account for the uncertainty affecting the simulator predictions.

5. Greenland Ice Sheet Reconstruction During the Last Interglacial

Abstract: This chapter employs Gaussian process emulation to tackle the following problem: to reconstruct the shape of the Greenland ice sheet during the last Interglacial, the last period in Earth's history characterised by warmer-than-today temperatures. We treat this as an inverse problem. We emulate the so-called $\delta^{18}O$ output of the HadCM3 climate simulator, as a function of ice shapes. Hence, we seek the shapes that match $\delta^{18}O$ records extracted from Greenland ice-cores. The work presents the non-standard feature of emulating over infinite-dimensional objects, such as ice shapes. The problem tackled here is of primary interest to the climate community, given that the current melting from the Greenland ice sheet represents the greatest contributor to the sea-level rise at global level.

5.1. Introduction

The work detailed in this chapter is the result of a collaboration started in September 2016, following an informal conversation with Dr. Louise Sime at a conference organised by the Past Earth Network. Louise Sime is a paleoclimate modeller at the British Antarctic Survey (BAS, Cambridge). The collaboration has involved myself (the author) and my supervisor Jochen Voss on the one side, and Louise Sime alongside her PhD student Irene Malmierca-Vallet on the other.

The problem presented by Louise Sime is a central one in paleoclimate: reconstruct how the Greenland ice sheet looked like during the Last Interglacial period (115–129 kya). To this aim, the idea was to use both climate simulations and ice-core records to approach the problem. It seemed apparent that the existing climate literature on the topic could benefit from the statistical contribution of GP emulation. At the same time, the collaboration would allow to develop an emulator on "ice shapes" (infinite dimensional objects), rather than on a standard finite number of inputs that are directly tuned in the simulations.

In the remainder of this section, we introduce in more detail the problem and its relevance. Subsection 5.1.1 reviews the literature information which allows to put the problem in context. Subsection 5.1.2 provides a simple illustration of key climate terms and concepts. Finally, Subsection 5.1.3 specifically illustrates our problem and the way it is tackled, providing as well an overview of the chapter structure.

5.1.1. The Issue of Current Sea-Level Rise

In Chapter 4 we discussed the interest of the climate community to understand the dynamics and nature of past warm climates, to shed light on future scenarios. One of the main sources of concern is the magnitude of future sea-level rise, brought about by polar ice melting. According to a recent estimate (Kopp et al. [2017]), under a high CO_2 emission scenario, land currently home to more than 150 million people may be submerged by the end of this century if no protective measures are adopted.

More than 99% of the ice present on Earth can be found within only two regions of

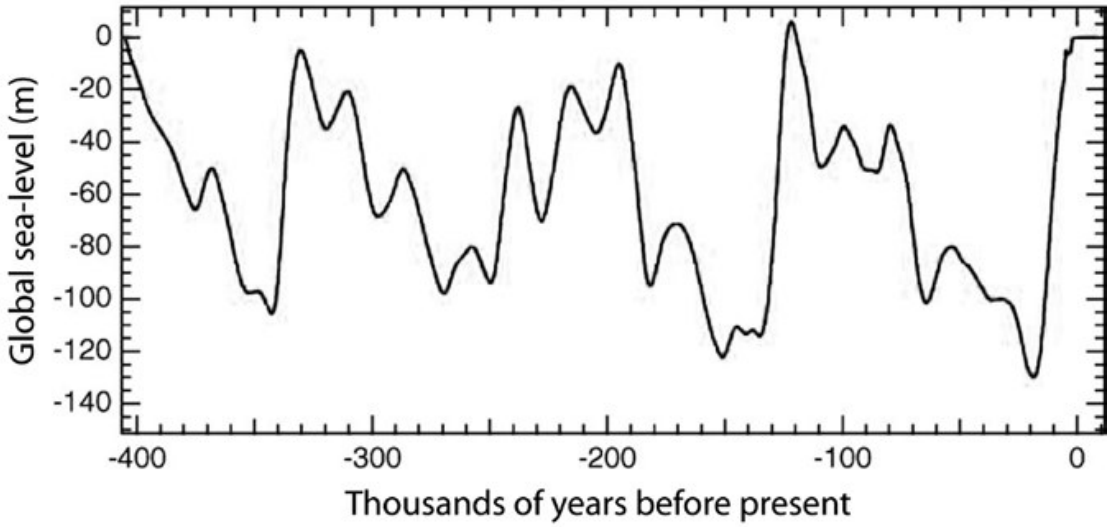


Figure 5.1: Reconstruction of global sea levels during the last 400 thousand years, from Williams and Gutierrez [2009].

our planet: Greenland and Antarctica. Antarctica holds the majority of Earth's ice: if the Antarctic ice melted completely, it is estimated that the sea level would rise of about 58 meters, on a global average (Fretwell et al. [2013]). Greenland contains a lower amount of ice, corresponding to approximately 7 meters of global sea-level equivalent (Robinson et al. [2011]). However, the Arctic island is undergoing a more dramatic ice loss than the Antarctic continent. Although all estimates come with significant uncertainties, most studies suggest that Greenland's current melting rate is about twice that of Antarctica, equating approximately 1 to 2 $\times 10^{14}$ kilograms of ice loss per year (Shepherd et al. [2012], IPCC [2013], Van den Broeke et al. [2016]). According to Brunnabend et al. [2012], this translates into about 0.3-0.6 mm of global sea-level rise per year. On this basis, in 2013 the IPCC recognised Greenland as the largest single contributor to the current sea-level rise.

The last time in history with higher-than-present sea levels was the Eemian Interglacial, also known as Last Interglacial (LIG; 115–129 kya); see Figure 5.1. Temperatures were at the time 2–3°C warmer than today on a global average; Arctic temperatures may have been up to even 4–5°C warmer (CAPE Members [2006]). The difference in sea levels with respect to present day is thought to have been of several meters (Kopp et al. [2009] estimates it to be greater than 6.6 meters with 95% probability). However, the contribution of the Greenland Ice Sheet (GrIS) to the latter

is highly uncertain, with estimated contributions ranging from as little as 0.3 meters to over 5 meters (Robinson et al. [2011]).

The uncertainty on the GrIS contribution to the LIG sea level stems directly from the uncertainty on the morphology itself of the GrIS during the LIG, and on the locations of predominant melting. Some studies suggest that strong melting happened in the south (*e.g.*, Otto-Bliesner et al. [2006]), some suggest it happened in the north (*e.g.*, Quiquet et al. [2013]), while some others in both (*e.g.*, Born and Nisancioglu [2012]). A common effort has therefore been undertaken to try to reduce the uncertainties regarding these reconstructions. To the aim, both climate simulations and geological data in the form of ice-core records have been used (Robinson et al. [2011], Stone et al. [2013]). In Subsection 5.1.2 we provide more details about ice cores and the information that they contain.

5.1.2. Ice Sheets as Frozen Archives of Earth's History

Ice sheets form at latitudes where the annual snowfall rate exceeds the annual snow melt: snow accumulates, and soon turns into ice under the above pressure. Through this process, a "frozen archive" of the Earth's climatic and atmospheric history builds up. Specific events may be recorded within the ice: for example, ashes may reveal an exceptional volcanic event happening hundreds or even thousands or kilometres away from the site. But more importantly, bubbles of air are trapped within the falling snow crystals, together with physical and chemical properties of the water forming the snow itself. Figure 5.2 shows a NASA 3D reconstruction of the GrIS near the Camp Century site (details in caption), which illustrates the "layered" structure of an ice sheet. By drilling down from its surface, scientists and engineers are able to extract deep cylinders of ice, whose physical and chemical properties can be studied to recover information from the past. These cylinders are known as ice cores: they may reach three or four kilometres in length, and often take several years to be extracted. Figure 5.3 shows a photograph of a core containing ice more than 16 thousand years old. The annual layer structure can be appreciated.

One of the most important pieces of information that ice cores contain comes from oxygen isotopes. Oxygen atoms always have eight protons in their nucleus. The

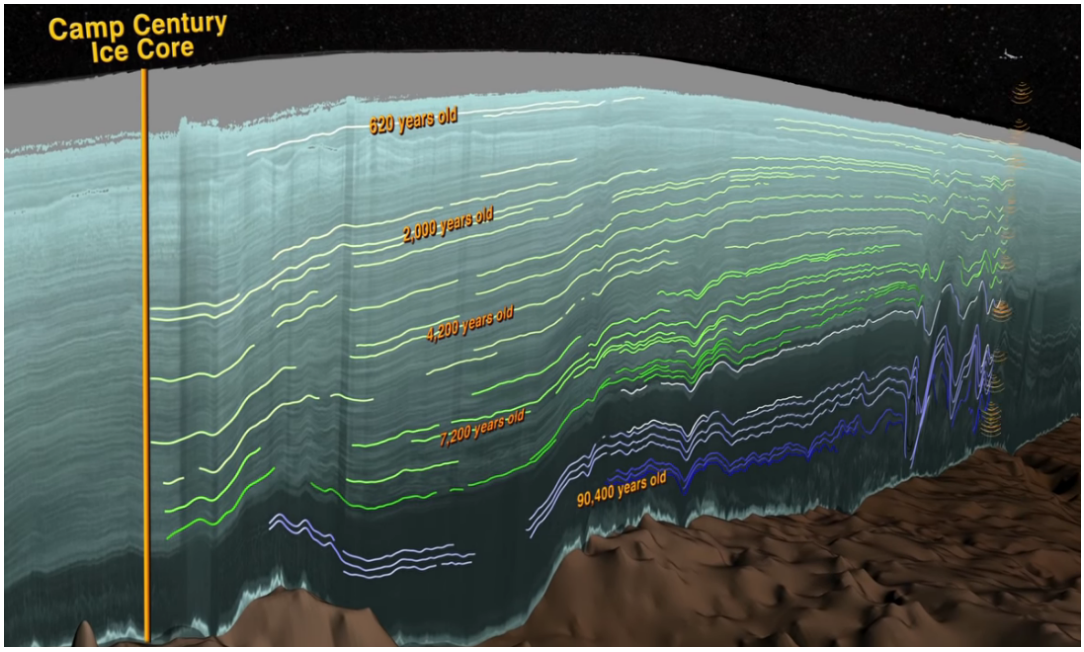


Figure 5.2: NASA 3D reconstruction of the Greenland ice layers near the Camp Century site (in the north west of Greenland: latitude 77.2°N, longitude 61.1°W). Video available at https://www.nasa.gov/content/goddard/nasa-data-peers-into-greenlands-ice-sheet.

number of neutrons in the nucleus can however vary: most atoms present eight, but a small percentage presents ten. Such variants of the same element are called isotopes: the additional neutrons do not affect the chemical properties of the element (oxygen in this case), but increase the atom weight. The two oxygen isotopes mentioned above are denoted by the symbols ^{16}O and ^{18}O, which reflect the total number of protons and neutrons in the nucleus. Lighter water molecules, formed by two hydrogen atoms and one ^{16}O atom, require less energy (or, equivalently, lower temperatures) than heavier water molecules to evaporate; they are therefore the first ones to escape water and be transformed into vapour in any naturally occurring evaporation process. As a consequence, the proportion of light and heavy oxygen isotopes present in precipitation water is informative of the temperature at which the water formed. Higher proportions of heavy isotopes are indicative of higher temperatures.

In a given layer of an ice core, the proportion of light and heavy oxygen isotopes can be measured and used to draw information about the ice sheet and the local climate, at the time the layer formed. Throughout geochemistry and paleoclimatology, the measure used is the so-called $\delta^{18}O$. It compares the heavy-to-light ratio of oxygen

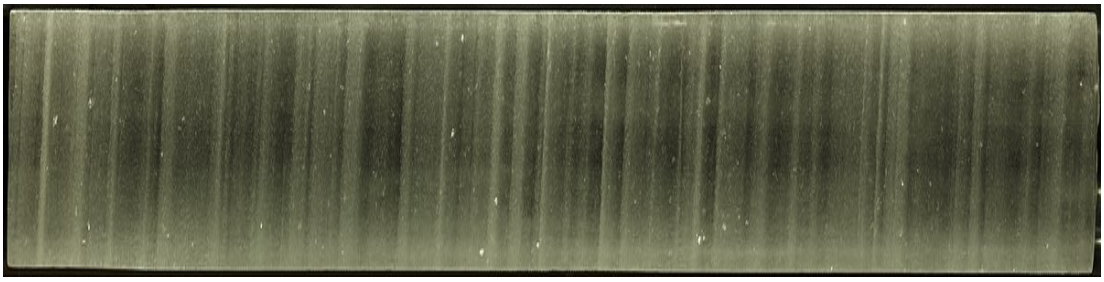


Figure 5.3: Photograph showing a section of an ice core, extracted from the GrIS at the GISP2 location (72.6°N, 38.5°W). Annual layers are clearly visible. The section shows ice from around 16,250 years ago, and was drilled at a depth of 1837 meters. Picture credit: Department of the Interior, U.S. Geological Survey.

isotopes in a water sample of interest, to the same ratio of a reference water sample. More precisely, we have:

$$\delta^{18}O = \frac{(^{18}O/^{16}O)_{\text{sample}}}{(^{18}O/^{16}O)_{\text{reference}}} - 1\,. \tag{5.1}$$

For completeness, we mention that the reference sample commonly used is the Vienna Standard Mean Ocean Water sample. Despite the name, it consists of pure distilled water, with no salt.

5.1.3. Overview of the Chapter

We have mentioned earlier that the LIG morphology of the GrIS is a matter of some controversy. The various reconstructions differ substantially from each other and consequently yield very different estimates of the GrIS contribution to the LIG sea-level rise. The aim of this chapter is to join information from climate simulations and from ice cores (in the form of $\delta^{18}O$ records), to provide constraints on the shape and extent of the LIG GrIS. We schematically summarise here the information that these two sources provide, and the way the two are merged in this work. More details are of course provided in the following sections.

- Ice-Core Records: Estimated LIG $\delta^{18}O$ values, with error bands, are provided at six Greenland locations, depicted in Figure 5.4.

- Climate Simulations: The HadCM3 climate model is used to simulate, at each of the six locations, $\delta^{18}O$ values corresponding to different GrIS morphologies. This allows to see the simulator as a map

$$f_{_L} : M \longmapsto \delta^{18}O\,, \tag{5.2}$$

  which associates, to a given morphology M, a $\delta^{18}O$ value at location L.

By comparing the simulated $\delta^{18}O$ values to the ice-core records, one can in principle identify the morphologies that are compatible with the records, and study their properties. An exhaustive search within the space of GrIS morphologies, however, is made unfeasible by the expensiveness of the simulator. Hence, we build an emulator of each of the six maps $f_{_L}$ in (5.2), and carry out a statistical comparison between the emulator predictions and the ice-core records. This allows to inspect the properties of the GrIS morphologies that are compatible with the available $\delta^{18}O$ records.

The chapter is structured as follows. In Section 5.2, we discuss the available ice-core records. In Section 5.3, we conveniently represent the simulator as a collection of functions, within an appropriate mathematical setting for our problem. The domain of these functions is a space of ice morphologies, infinite dimensional objects, fact that marks a difference from standard emulation settings. We therefore devote Section 5.4 to its parameterisation, and describe how to generate input morphologies for the simulator, starting from elements of the parameterised space. We are then ready to discuss the emulators. Section 5.5 describes the experimental design, built in two waves; Section 5.6 details the construction and validation of our emulators. Hence, in Section 5.7, we illustrate how the emulators are used to identify morphologies that are compatible with records. The results of our approach are discussed in Section 5.8, under both a more mathematical and a more applied point of view. Finally, in Section 5.9, we conclude the chapter with some remarks, and an overview of future directions of investigation in the field.

The data produced as a result of the work detailed in this chapter can be accessed at the following repository: https://ramadda.data.bas.ac.uk/repository/entry/show?entryid=35aed839-1634-4692-b6d6-4d6312953eb5, or via the following link: https://data.bas.ac.uk/full-record.php?id=GB/NERC/BAS/PDC/01283.

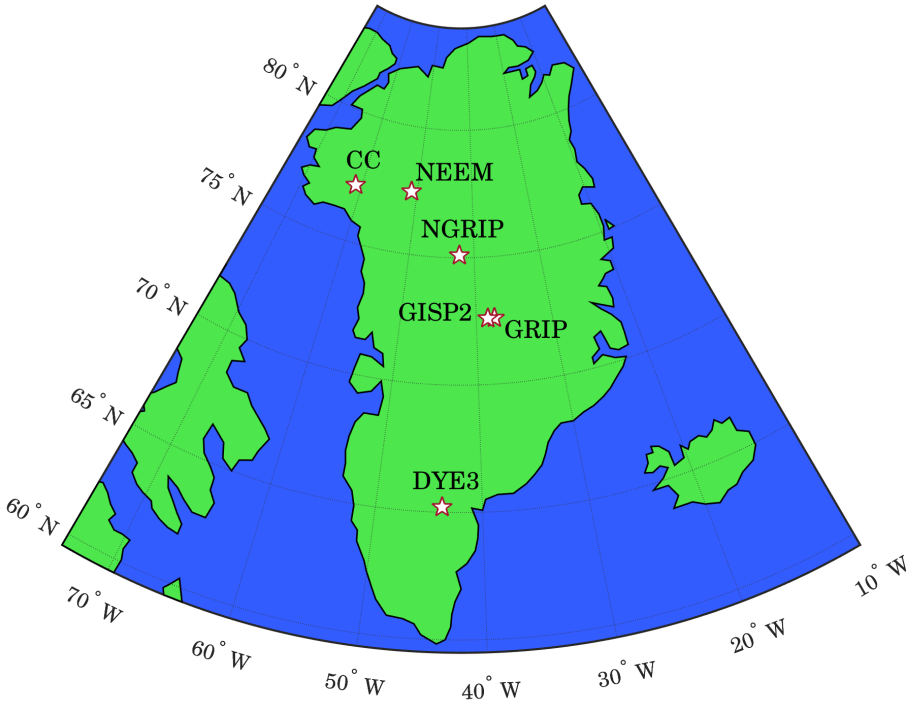


Figure 5.4: Position of the six drilling sites in Greenland where the ice-core records ($\delta^{18}O$) used in this work come from. Geographical coordinates: Camp Century, abbreviated CC (77.2° N, 61.1° W); NEEM (77.45° N, 51.06° W); NGRIP (75.1° N, 42.3° W); GISP2 (72.6° N, 38.5° W); GRIP (72.6° N, 37.6° W); DYE3 (65.2° N, 43.8° W).

5.2. Available Ice-Core Records

The $\delta^{18}O$ records that we use, alongside climate simulations and GP emulation to constrain the LIG GrIS morphology, come from ice cores extracted at six different sites. These are depicted in Figure 5.4. We briefly provide below an account of the six sites, and the associated ice-core drilling projects.

- **NEEM**. Site in North Greenland. The ice core was extracted within the North Greenland Eemian Ice Drilling project, NEEM Community Members [2013]. The drilling started in June 2009 and ended in July 2010, after hitting the bedrock at a depth of 2537 metres.

- **GRIP**. Site around the centre of Greenland. The acronym stands for GReenland Ice core Project, GRIP Members [1993]. Drilling started in 1990 and ended in

summer 1992, extracting a core 3028 metres long.

- **NGRIP**. Site around 320 km North of GRIP (from which the name), NGRIP Members [2004]. The NGRIP ice core is 3085 metres long. It took almost four years to be extracted (1999–2003).

- **GISP2**. Greenland Ice Sheet Project 2, Johnsen and Vinther [2007]. Site located 28 km to the west of GRIP. The drilling was completed in five years (1988–1993), and recovered a 3053-metre-long ice core.

- **Camp Century** (CC). The site is situated in the North West of Greenland. The core was drilled during the 1960's, and is 1390 metres long.

- **DYE3**. The Southern most site of the six considered. The drilling was carried out during the 1970's, within the Greenland Ice Sheet Project (GISP). It was completed in 1981, extracting a 2038-metre-long ice core. The core is not intact: together with the Camp Century one, it represents the least well-preserved core.

Historical Note

The Camp Century site was born during the cold war, and it was, in reality, a military site. It was the base of a top-secret US program (the Iceworm project), whose aim was to build around 600 ballistic missiles trained towards the Soviet Union. The aim was, fortunately, not accomplished. The site was conveniently placed between the US and the USSR, and was at the same time isolated from the rest of the world, therefore providing a "perfect" location in the eyes of the US, to carry out the project. In 1960, a scientific drilling project was started at the site. The project, mainly supposed to act as a "cover" to the secret military program, was in the end extremely successful and drilled the 1390-metre-long ice core mentioned above. By revealing the great potential of ice cores, it opened up a whole new scientific era in the study of past climate, and pioneered a number of future drilling projects which we have referenced above (NEEM, GRIP, GISP). For a more detailed account of the history behind the Camp Century site, we refer the interested reader to the online article available here (reference Gertner [2019] in bibliography).

	Anomaly values (LIG minus present day) at each ice-core site					
	NEEM $\delta^{18}O$ ‰	**NGRIP** $\delta^{18}O$ ‰	**GRIP** $\delta^{18}O$ ‰	**GISP2** $\delta^{18}O$ ‰	**Camp Cent.** $\delta^{18}O$ ‰	**DYE3** $\delta^{18}O$ ‰
Most likely	+3.6	+3.1	+3.2	+2.7	+2.5	+4.7
Minimum	+2.7	+2.1	+2.2	+1.7	+0.0	+0.0
Maximum	+4.0	+3.8	+3.5	+3.4	+4.0	+5.2

Table 5.1: Stable water isotopic ($\delta^{18}O$) information from the six Greenland ice cores used in this work. Most likely, maximum, and minimum values are provided, see text for full details.

Due to the difficulties arising in dating ice-core layers, and due to the possibility of missing layers, it is not possible to obtain, from the six ice cores, a precise estimate of what the $\delta^{18}O$ at the location was during a specific time of the LIG. Compiling the available information therefore requires to account for these uncertainties. Here, at each of the six sites, we consider a $\delta^{18}O$ central estimate for the time 125 kya, alongside a maximum and minimum value. These have been extracted on the basis of the information reviewed in Johnsen and Vinther [2007] or available from the members of the community in charge of the ice core drilling. Table 5.1 summarises the information. All reported values are anomalies with respect to present-day measurements at the same sites.

The poor preservation of the oldest ice cores (DYE3 and Camp Century) makes it difficult to have lower $\delta^{18}O$ estimate at these sites. In Table 5.1 we report the present-day value as minimum $\delta^{18}O$. In Section 5.8, however, we consider three different scenarios, which reflect three different minimum values for the anomalies at these sites.

5.3. Climate Simulations: Inputs and Outputs

The climate simulator that we employ is HadCM3, whose main features were already introduced in Chapter 4 (Section 4.3). Here, we use the simulator to reproduce the

$\delta^{18}O$ response to changes in the shape and extent of the GrIS. We provide more details below.

A "shape" of the GrIS is best referred to as morphology. For instance, a morphology may be characterised by massive ice presence in the north, or in the south, and so on. An intuitive way to identify a morphology would then be to specify the ice thickness at any place in Greenland. For simulation purposes, it is more convenient to refer to surface elevation rather than ice thickness. If we denote by $G \subseteq S^2$ the two-dimensional subset of the sphere (thought of as the Earth) identifying Greenland, a GrIS morphology can then be viewed as a map

$$M : G \longrightarrow \mathbb{R}^+, \tag{5.3}$$

where the quantity $M(l)$ represents the surface elevation of the morphology at location $l \in G$. Information about the GrIS morphology can be supplied to the climate simulator as one of the boundary conditions, essentially in the form (5.3) above: the natural adjustment consists in providing the surface height information at a finite number of grid cells only, rather than an infinite number of locations.

In light of the above, we can use the model to simulate the climate response associated with different morphologies. Here, in particular, we look at the $\delta^{18}O$ output field of the simulator. As all other output fields, this is provided on a number of grid cells. Since the aim is to compare the simulated outputs to the available ice-core records, we extract the outputs at the six sites of interest by interpolating over nearby cells. This way, for each site L introduced in Section 5.2, the simulator can be represented as the following map ($\mathcal{M}$ denotes the space of all morphologies):

$$\begin{array}{rccc} f_{_L}: & \mathcal{M} & \longrightarrow & \mathbb{R} \\ & M & \mapsto & \delta^{18}O \end{array} . \tag{5.4}$$

The map $f_{_L}$ associates, to a given morphology M, the simulated $\delta^{18}O$ output at location L. By running the simulator on a selected number of design morphologies, we can build emulators of the six maps $f_{_L}$ of interest (one for each of the sites of Section 5.2), and compare the outputs to the available ice-core records.

Our input space $\mathcal{M}$, however, differs from the ones encountered in classical emulation

Table of Constants

Name	Meaning	Value
m	Number of morphologies from previous studies, used as starting point for PCA	14
s	Dimension of morphologies from previous studies, used as starting point for PCA	122×314 (38,808)
r	Number of PCs used to generate new morphologies (equation (5.10))	8
p	Dimension of emulators' input space (by construction, $p = r$)	8
n	Number of emulators' design points	69
N	Number of morphologies generated to perform DMC (Section 5.7)	10^7

Table 5.2: Table reporting the meaning and value of the main constants used in the chapter. While most of these will be introduced in later pages, they are all reported here to provide a compact reference.

settings. Although represented in the simulator via a finite number of values (one surface elevation value for each grid cell), it is in reality infinite-dimensional: specifically, a space of functions representing ice shapes. We therefore need to first parameterise the set $\mathcal{M}$, in order to be able to identify a morphology by a small number of independent parameters only. In the next section we explain how we perform the task.

5.4. Parameterise and Generate New Morphologies

The idea behind our morphology parameterisation is to find an "interesting" finite-dimensional subspace $\mathcal{M}'$ of $\mathcal{M}$, alongside an appropriate basis which allows to identify a morphology via the basis coefficients. To identify the subspace, we gather from previous studies $m = 14$ reconstructions identifying a wide range of GrIS morphologies, and consider the affine subspace that these generate. In order to find an appropriate

basis for this, and to further lower its dimensionality, we use the PC approach presented in Chapter 3. Details of the overall procedure, comprising an ice-land mask generation, are given in Sections 5.4.1–5.4.3.

5.4.1. Regridding the Original Morphologies

Figure 5.5 shows the surface elevation of the N original morphologies that we choose. Details of the corresponding studies are provided in the caption of the same figure. It can be appreciated that the chosen studies cover a wide range of GrIS morphologies. Each morphology is represented by a matrix, whose elements report the surface elevation at the different grid cells. The grid used by the different studies varies. In order to apply the PC procedure detailed in Chapter 3 and find a basis of the space that the morphologies generate, we firstly need to represent each morphology as a vector of the same length. To retain the detail provided by the reconstructions, we regrid each morphology into a (longitude-latitude) rectangular grid with resolution of 0.2° in both directions: latitude values range from 59.7° N to 83.9° N; longitudes values range from 73.6° W to 11° W. This yields a total of $s = 122 \times 314$ grid cells for each regridded morphology. The plots in Figure 5.5 show the regridded morphologies.

The aforementioned regridding has not been performed on the original latitude and longitude coordinates. Such a procedure would indeed generate heavily distorted regridded morphologies, especially at high latitudes. This is due to the non-uniform local behaviour of the transformation mapping latitude-longitude coordinate into points on the sphere. To circumvent the issue, we carry out the regridding on the coordinates obtained by projecting a point on the sphere orthogonally onto the plane π tangent to the sphere at the point with latitude $\varphi^* = 72°$ N and longitude $\theta^* = 40°$ E. The point with coordinates (φ^*, θ^*) essentially lies in the centre of Greenland. We derive the details of the transformation below.

A point on the sphere with latitude φ and longitude θ has the following coordinates:

$$\mathbf{\Phi}(\varphi, \theta) = \big(\cos\varphi\cos\theta,\, \cos\varphi\sin\theta,\, \sin\varphi\big)^T \in S^2 \subseteq \mathbb{R}^3. \tag{5.5}$$

An (orthogonal) basis for the plane tangent to the sphere at the point (φ^*, θ^*) is given by the following two vectors:

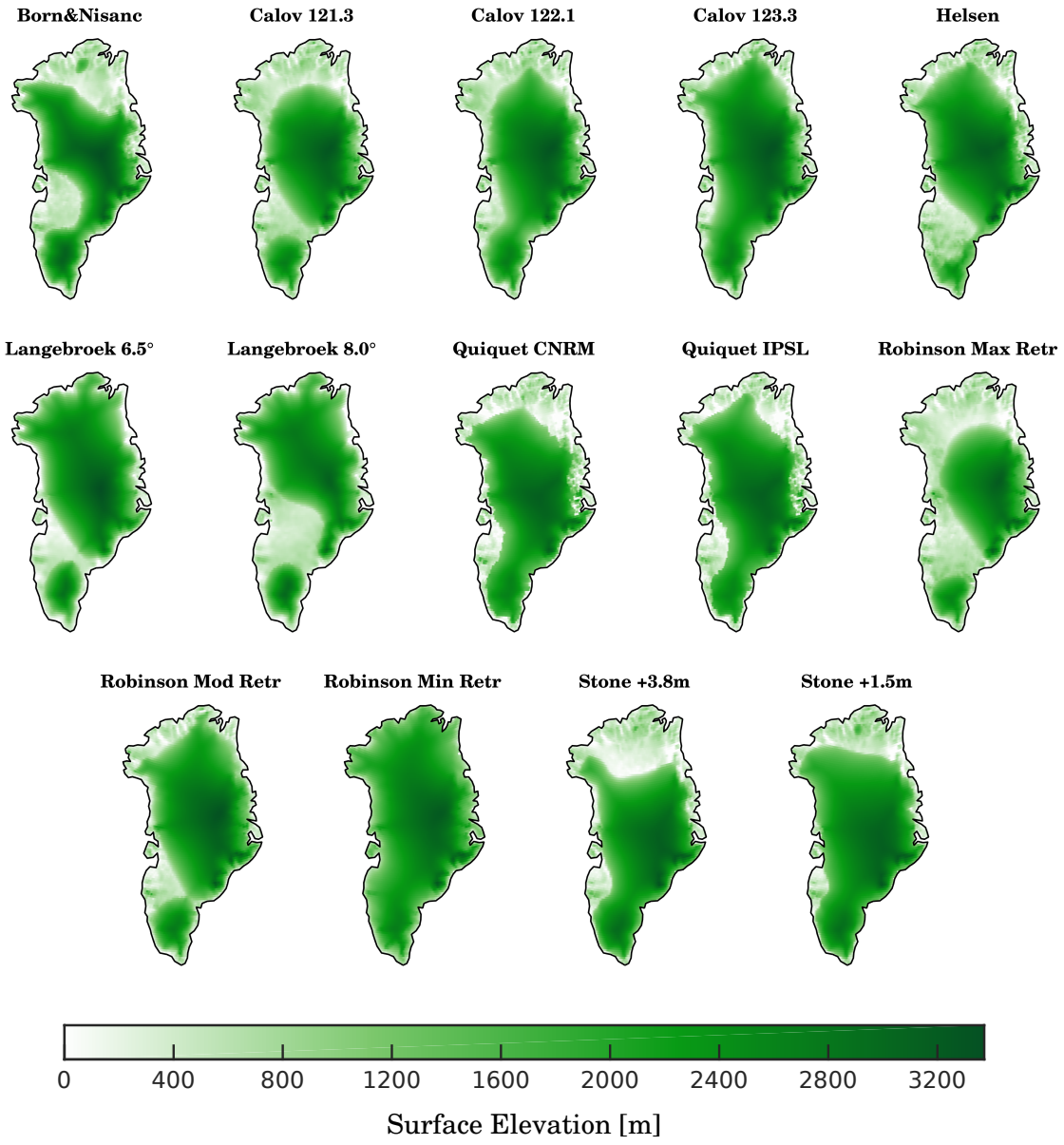


Figure 5.5: Surface elevation of the initial GrIS morphologies used in this work to generate new synthetic morphologies. From top to bottom, left to right, reconstructions from: Born and Nisancioglu [2012]; Calov et al. [2015] (reconstructions at 121.3, 122.1 and 123.3 kya respectively); Helsen et al. [2013]; Langebroek and Nisancioglu [2016] (reconstructions corresponding to simulated temperature lapse rate of 6.5°C/km and 8°C/km); Quiquet et al. [2013] (CNRM and IPSL anomaly experiments); Robinson et al. [2011] (reconstructions corresponding to strong, moderate and weak GrIS retreat); Stone et al. [2013] (reconstructions for maximum contribution (+3.8 m, at 121 kya) and most likely contribution (+1.5 m, at 123.5 kya) to sea-level rise).

$$\boldsymbol{v_1} = \frac{\partial \boldsymbol{\Phi}}{\partial \varphi}(\varphi^*, \theta^*)\,, \quad \boldsymbol{v_2} = \frac{\partial \boldsymbol{\Phi}}{\partial \theta}(\varphi^*, \theta^*)\,. \tag{5.6}$$

We consider the normalised versions of the two vectors, so that $\|\boldsymbol{v_1}\| = \|\boldsymbol{v_2}\| = 1$.

Given any point $\boldsymbol{\Phi}(\varphi, \theta) \in S^2$, its orthogonal projection onto the plane π can be written as linear combination of $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. Since the basis is orthonormal, the coefficients of the linear combination are as follows:

$$\big\langle \boldsymbol{\Phi}(\varphi, \theta), \boldsymbol{v_1} \big\rangle \quad \text{and} \quad \big\langle \boldsymbol{\Phi}(\varphi, \theta), \boldsymbol{v_2} \big\rangle\,, \tag{5.7}$$

where the symbol $\langle \cdot\,, \cdot \rangle$ denotes the Euclidean scalar product of $\mathbb{R}^3$. For convenience of notation, let us denote by $\boldsymbol{V} \in \mathbb{R}^{3\times 2}$ the matrix with columns $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. This way, we have defined a transformation of coordinates:

$$\begin{aligned} \boldsymbol{\Psi}\colon \ \Big[-\frac{\pi}{2}, \frac{\pi}{2}\Big] \times S^1 \ &\longrightarrow \quad \mathbb{R}^2 \\ \big(\varphi, \theta\big) \ &\mapsto \quad \boldsymbol{V}^T \boldsymbol{\Phi}(\varphi, \theta)\,. \end{aligned} \tag{5.8}$$

The coordinates $\boldsymbol{\Psi}(\varphi, \theta)$ are used in place of (φ, θ) to interpolate the surface height values in the original grid of the N starting morphologies, at the lattice points identified by the uniform 0.2°-wide grid introduced at the beginning of this section.

The interpolation is performed via triangulation-based cubic splines. To this aim, we use the MATLAB function `griddata`. The code implementing our procedure can be found in the Matlab Appendix F.1.

Figure 5.6 provides an illustration of the space deformation induced by (5.8) in each of the two directions, latitude and longitude. In particular, notice how, especially at high latitudes, squares are transformed into tall and thin rectangles. Interpolation carried out in the new coordinates yields therefore very different results than interpolation in the old coordinates. As an elementary example of this, consider a square Q in the left panel of Figure 5.6, the square $Q_{2\mathrm{R}}$ two "steps" on its right, and the square Q_{T} just on top of Q. In the old coordinates, the square $Q_{2\mathrm{R}}$ lies farther from Q than Q_{T} does. In the new coordinates, especially at high latitudes, the role of the transformed "squares" is inverted, with obvious consequences on interpolation.

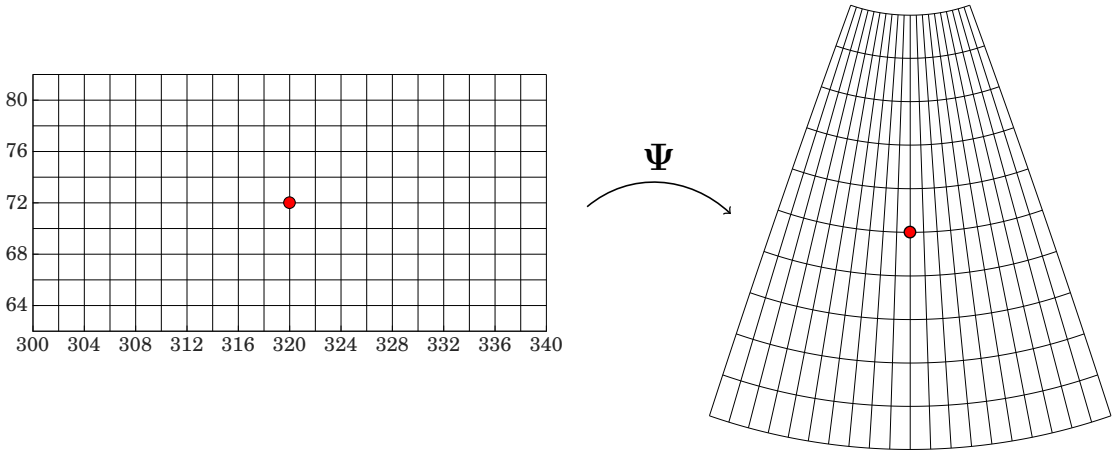


Figure 5.6: Illustration of how the transformation $\boldsymbol{\Psi}$ in (5.8) deforms the space. In the left part, a uniform grid in latitude and longitude is shown, with a step-size of 2°. In the right part, its image under $\boldsymbol{\Psi}$ is shown. The red dot on the left represents the point of tangency between the sphere and the plane onto which points of the sphere are projected through $\boldsymbol{\Psi}$: the point is mapped into $(0,0) \in \mathbb{R}^2$ after the transformation. Units along the x and y axes are not shown in the right plot, however the same scale is used on both axes to appreciate the deformation induced by $\boldsymbol{\Psi}$.

5.4.2. Principal Components and Synthetic Morphologies

Once the m initial morphologies have been regridded, we perform on these the PC procedure described in Chapter 3. This allows to find a basis of the space that they generate, by identifying directions (*i.e.*, morphologies) that sequentially explain most-to-least of the data set variance. Figure 5.7 shows the elements of this basis as obtained through our procedure. These are the PCs, which we denote in the following by $\boldsymbol{V_j}$.

Any of the m original morphologies $\boldsymbol{M}^{(k)}_{\text{orig}} \in \mathbb{R}^s$ can be recovered as linear combination of the PCs, plus the fixed term $\overline{\boldsymbol{M}}$ obtained as average of the morphologies themselves (average computed grid cell by grid cell). That is:

$$\boldsymbol{M}^{(k)}_{\text{orig}} = \overline{\boldsymbol{M}} + \sum_{j=1}^{m-1} \alpha_j^{(k)} \boldsymbol{V_j} \in \mathbb{R}^s, \qquad k = 1, \dots, m, \tag{5.9}$$

for some coefficients $\alpha_i^{(k)} \in \mathbb{R}$. By generalising to any set of coefficients, we can represent any element belonging to the space generated by the PCs. Our specific

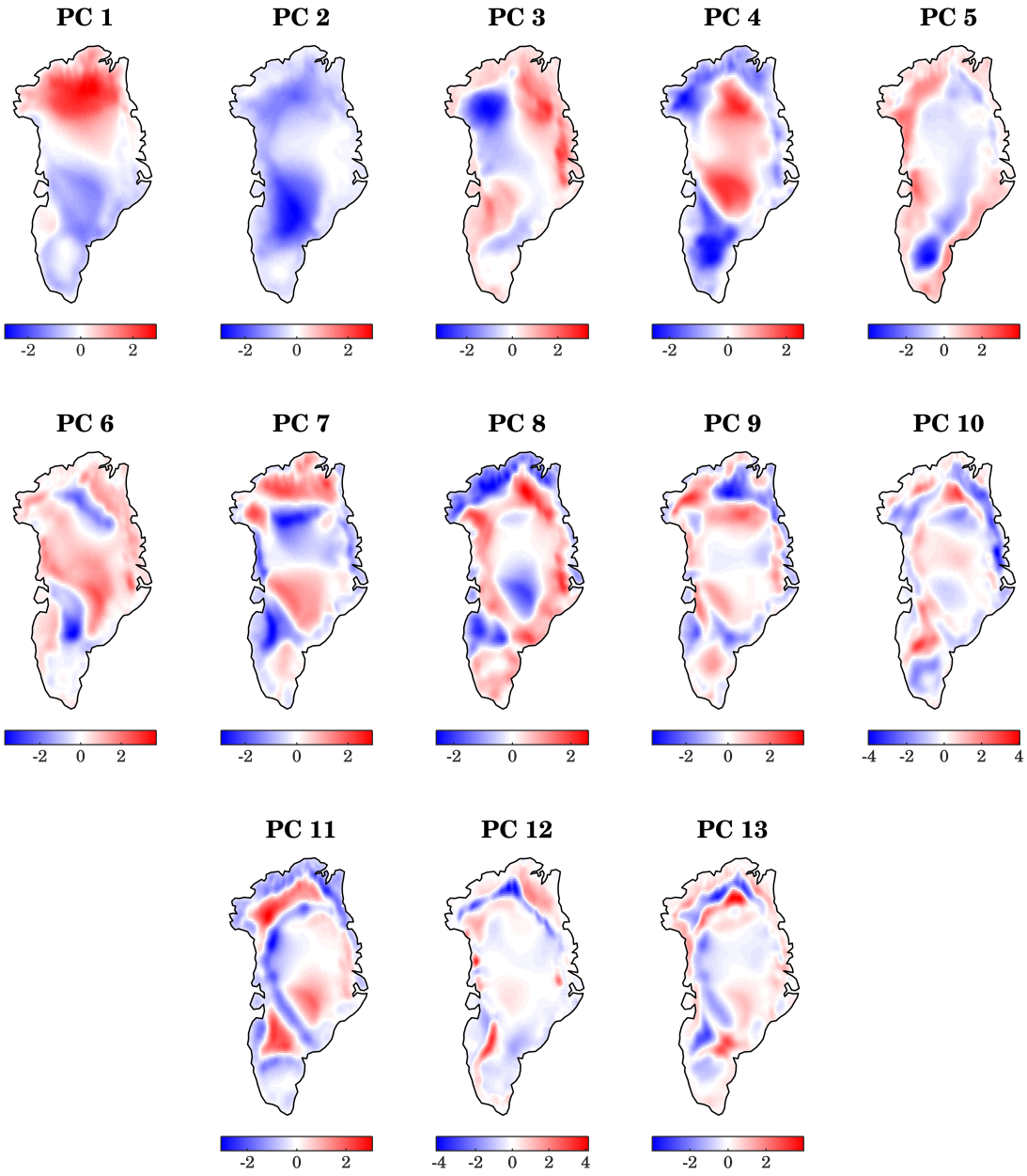


Figure 5.7: PCs used to generate new synthetic morphologies via equation (5.10). Within each PC, areas of opposite sign yield opposite contributions to the surface height of the generated morphology. Values in the colour bars do not have a physical meaning: the PCs have norm equal to one with respect to the scalar product used to carry out the PCA.

choice, motivated in the following, is to consider the space spanned by the first $r = 8$ PCs only. In other words, any morphology considered in this work is of the following

form:

$$\boldsymbol{M}(\boldsymbol{\alpha}) := \overline{\boldsymbol{M}} + \sum_{j=1}^{r} \alpha_j \boldsymbol{V_j} \in \mathbb{R}^s, \tag{5.10}$$

for some set of real coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_r) \in \mathbb{R}^r$.

The choice of employing fewer than available PCs is motivated by the computational limitations associated with the running of climate simulations. Indeed, as we shall see in Section 5.5, the vectors $\boldsymbol{\alpha} \in \mathbb{R}^r$ which identify a morphology via equation (5.10) will represent inputs to the emulators built in this chapter. Time and computational constraints imposed a design formed by less than 70 simulations (details in Section 5.5), hence it seemed appropriate to lower the dimension of the input space to suit this constraint[16]. The specific choice $r = 8$ reflects the information summarised in Table 5.3. The table reports the percentage of variance explained by the different PCs, and shows that the first eight PCs account for more than 95% of this. Let us recall that the variance associated with each PC is the sample variance

$$\sigma_j{}^2 = \mathrm{Var}\left\{\alpha_j^{(1)}, \ldots, \alpha_j^{(m)}\right\}. \tag{5.11}$$

Table 5.3 also reports the values of the standard deviations σ_j.

For our analysis, it is convenient to consider a prior distribution on the set of morphologies or, equivalently, on the set $\mathbb{R}^r = \mathbb{R}^8$ of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_8)$ associated with a morphology via equation (5.10). Since the prior distribution should be concentrated in a region of the parameter space which also contains the m original morphologies, we consider

$$\alpha_j \sim N\left(0, \sigma_j{}^2\right), \tag{5.12}$$

independently. That is, the prior is multivariate normal, with centred independent components of variance σ_j^2. We will use this prior distribution when designing the simulation runs in Section 5.5, and when the compatibility with the ice-core records is explored in Section 5.7.

We conclude with a note. Through equation (5.10), there is the possibility that an

[16] A commonly-used rule of thumb suggests around 10 design simulations per emulator input dimension.

	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7	PC 8
σ_j	271.12	228.56	145.45	141.41	91.37	86.16	75.40	65.21
% of Explained Variance	36.37	25.85	10.47	9.90	4.13	3.67	2.81	2.10
Cumulative % of Expl. Var.	36.37	62.22	72.69	82.59	86.72	90.40	93.21	95.31

	PC 9	PC 10	PC 11	PC 12	PC 13
σ_j	56.62	47.80	44.45	36.72	25.71
% of Explained Variance	1.59	1.13	0.98	0.67	0.33
Cumulative % of Expl. Var.	96.90	98.03	99.01	99.67	100.00

Table 5.3: In the first row of each table, values of the standard deviations σ_i associated with the PCs used in equation (5.10). In the second and third rows, corresponding percentages of explained variance (single and cumulative).

element $\boldsymbol{\alpha} \in \mathbb{R}^8$ generates a morphology with unrealistically low, or even negative, surface heights. To avoid that this be the case, at each grid cell we consider the maximum between the surface height from equation (5.10), and the bedrock height as provided with the morphology of Stone et al. [2013] corresponding to +1.5 m of sea-level rise. The possibility of unrealistically high surface elevation is dealt with during the generation of the design morphologies; a more comprehensive criterion of physical plausibility of a GrIS morphology is introduced in Section 5.7, when the compatibility between simulated $\delta^{18}O$ and ice-core records is explored.

5.4.3. Mask Generation of Synthetic Morphologies

Subsection 5.4.2 establishes a bijection between the space of synthetic morphologies that we consider in this work and $\mathbb{R}^8$. Some morphologies are used as input to actual

$\delta^{18}O$ simulations. In this case, a corresponding land-ice mask must also be provided to the simulator: at each grid cell, this specifies whether the morphology is covered by ice or not. We explain below how we generate masks. Being able to associate a mask to a given morphology will also be useful during the discussion of our results, in Section 5.8.

For the $m = 14$ original morphologies, the mask is provided. For a new morphology $\boldsymbol{M}$, the main idea behind the mask generation is as follows: first, associate ice with a cell if the height of $\boldsymbol{M}$ at the cell is greater than a given threshold, and associate land otherwise; hence, smooth the mask. The threshold is computed on the basis of the value of the masks and heights of the m original morphologies at the cell in question. More precisely, at a grid cell, we proceed as follows:

1. Let $I_1, \ldots, I_q$ and $L_1, \ldots, L_s$ $(q + s = m)$ be the surface heights of the original morphologies: the letter I or L serves to distinguish between morphologies having ice or land at the grid cell, respectively.

2. Let a be the minimum of $I_1, \ldots, I_q$, and b be maximum of $L_1, \ldots, L_s$. Define $c = (a + b)/2$.

3. Associate ice to the grid cell in question if the corresponding surface height of $\boldsymbol{M}$ is greater than c, and land otherwise.

Finally, in order to smooth possible irregular patterns of the mask obtained through steps 1–3, we generate a new mask where ice (land) is associated with each grid cell, according to whether a majority of ice (land) cells are present in the original mask, within a circle of radius 35 km around the grid cell centre. This last process is repeated ten times, at which point changes become hardly detectable.

Subsection F.2 of the appendix shows the code implementing the procedure described above. The first routine (`ice_mask_generator.m`) implements steps 1–3 above, while the following two routines deal with the smoothing process and the computation of ice proportion in disks of given radius around the centre of each grid cell.

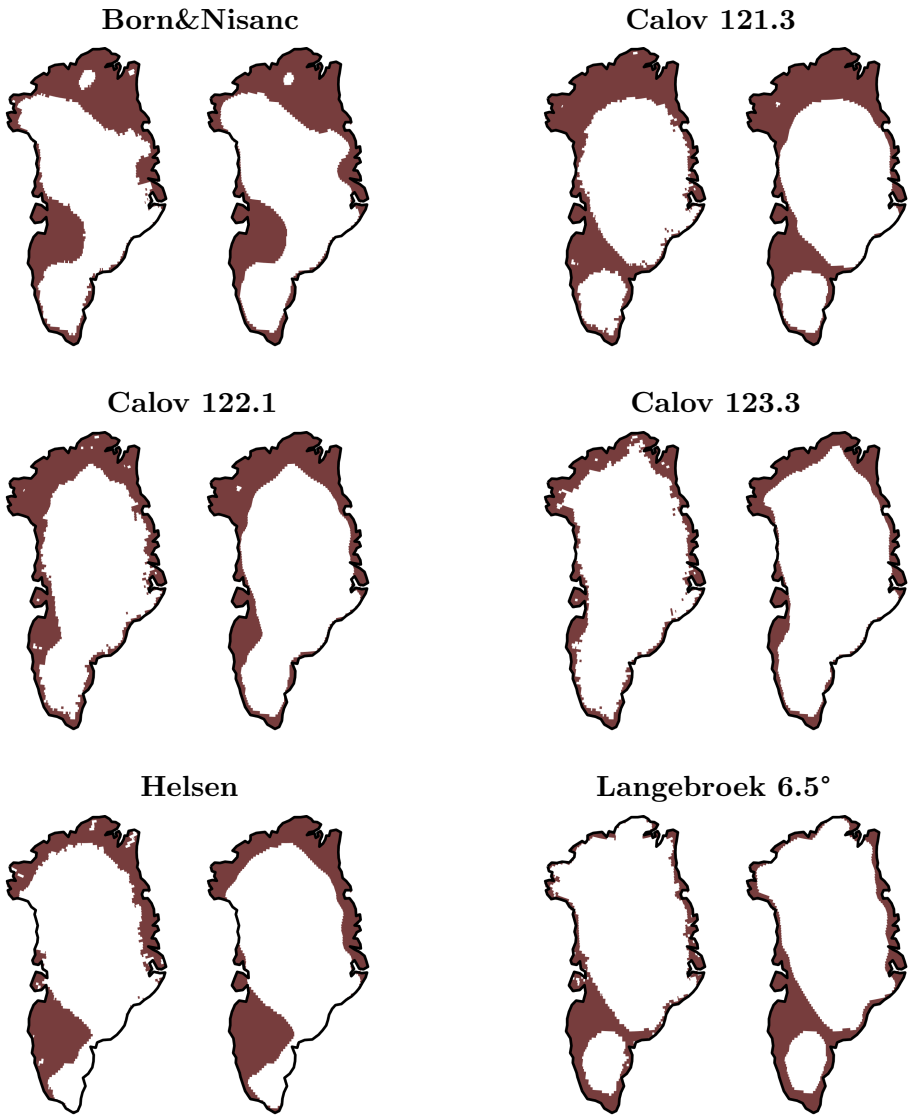


Figure 5.8: For each of the first six morphologies used as starting point to our PCA (see Figure 5.5 for their plots), we compare here the mask originally associated with the morphology (left) to the synthetic mask generated via our procedure (right). It can be appreciated that differences are mostly negligible. The same comparison for the remaining eight morphologies is shown in Figure 5.9.

Figure 5.8 and Figure 5.9 allow to compare the provided masks of the m original morphologies with the synthetic masks generated for these by the procedure detailed above. The similarities within each pair of masks are remarkable, supporting the validity of our procedure.

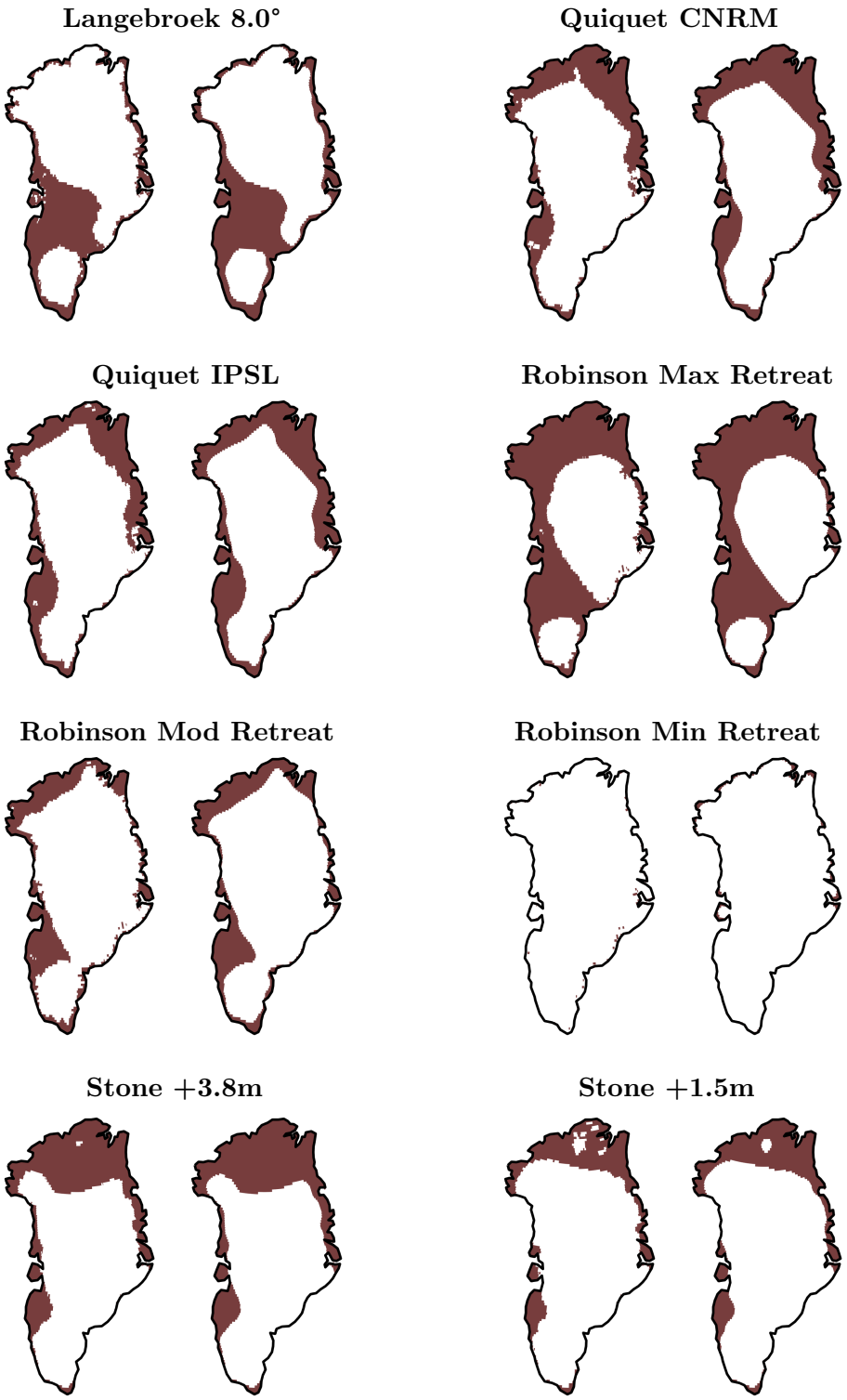


Figure 5.9: Same mask comparison as in Figure 5.9, here shown for the last eight morphologies used as starting point to our PCA (details of these in Figure 5.5).

5.5. Experimental Design

As specified in equation (5.4), for our purposes it is convenient to see the simulator as a collection of maps $\{f_L\}$, where f_L models the $\delta^{18}O$ response at location L to changes in morphologies. Having parameterised the set of morphologies by $\mathbb{R}^r$ via equation (5.10), $r = 8$, for each location L we can represent the map $f_L(\cdot)$ as follows:

$$\begin{array}{rccc} f_L: & \mathbb{R}^8 & \longrightarrow & \mathbb{R} \\ & \boldsymbol{\alpha} & \mapsto & \delta^{18}O \end{array} . \tag{5.13}$$

This way, we can see each map $f_L(\cdot)$ as a simulator defined on a low-dimensional input space and characterised by a univariate output: this is precisely the setting within which GP emulation has been presented in Chapter 2. We can therefore emulate the six maps $f_L(\cdot)$ of interest to us, *i.e.*, the ones corresponding to the six ice-core sites introduced in Section 5.2, and compare the emulated $\delta^{18}O$ outputs to the ice-core records at these locations.

The aim of this section is to provide details of the experimental design associated with the six emulators. Before this is done, we recall some notation and terminology from Chapter 2, and highlight their counterparts in this chapter.

Recall and Forthcoming Terminology/Notation

In Chapter 2, we have denoted by $\mathcal{P} \subseteq \mathbb{R}^p$ the simulator input space. Equation (5.13) shows that, for any of the six locations L of interest, we have here $\mathcal{P} = \mathbb{R}^8$. The simulator input space has therefore dimension $p = 8$, by construction equal to the number r of PCs used to generate morphologies via (5.10). Using the same notation of Chapter 2, we denote a general input of the simulator by $\boldsymbol{x} \in \mathbb{R}^p$.

In order to build an emulator of each map $f_L(\cdot)$, we need the actual simulated response on a small number n of inputs. We call these design points and denote them by $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \in \mathbb{R}^p$. The index $i = 1, \ldots, n$ will refer to them or to associated quantities. The morphologies associated with the design points will be called design morphologies, and denoted by $\boldsymbol{M_1}, \ldots, \boldsymbol{M_n}$ (*i.e.*, $\boldsymbol{M_i} := \boldsymbol{M}(\boldsymbol{x_i})$ with the notation of equation (5.10)). The index $j = 1, \ldots, p$ will be used to refer to quantities associated with the PCs (*e.g.*, the components of an input $\boldsymbol{x} \in \mathbb{R}^p$).

Our design is built in two waves. Each wave aims at assessing different properties of the maps $f_L(\cdot)$, as we explain below in Subsection 5.5.1 and Subsection 5.5.2. Before doing that, however, we provide a brief account of the simulation set-up.

All climate simulations are forced with greenhouse-gas values and orbital forcing which are appropriate for the time 125 kya, peak of the LIG. A 125 kya control simulation is initially run, with modern-day GrIS configuration, for 400 model years: this ensures that quasi-equilibrium conditions between the atmosphere and the upper ocean are reached. The simulations of interest for this work, with modified GrIS configuration, are then run on from the end of the previous "spin-up" simulation, for additional 70 model years. The average of the last 50 years is computed. As last step, $\delta^{18}O$ simulated values from a PI control simulation, set-up with atmospheric gas composition appropriate for the year 1850, are subtracted from the LIG-simulated outputs obtained above. The final values used represent therefore LIG minus PI anomalies.

5.5.1. Wave 1

The aim of the first wave of simulations is to generate design morphologies that are well-scattered within the region of space that the emulator is required to explore. To accomplish the task, we use a quasi-random sample from the prior distribution (5.12). This is obtained in the following way:

1. We generate a Halton sequence $\{\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}\} \subset [0,1]^p$ of well-scattered points in the p-dimensional unit cube ($p = 8$);

2. We then consider the sequence $\{\boldsymbol{x}_{\boldsymbol{i}}\} \subset \mathbb{R}^p$ obtained by applying the inverse of a $N\big(0, {\sigma_j}^2\big)$ cumulative distribution function to the j^{th} component of each vector $\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}$, for $j = 1, \ldots, p$ (values of σ_j in Table 5.3).

Halton sequences (Kocis and Whiten [1997]) are particular low-discrepancy sequences: these have been briefly introduced in Chapter 4, Section 4.5, and serve the purpose of systematically scattering points within the unit cube, ensuring that no region of space is left uncovered, and that no pair of points are too close to each other. More on low-discrepancy sequences can be found in Santner et al. [2003, Chap. 5]. The transformation used in point 2 above on the original Halton sequence yields design points

$\boldsymbol{x_i}$ with the correct variance along each PC, and whose corresponding morphologies systematically cover different GrIS scenarios for 125 ka. We observe that, essentially, the design points form a sample of a multivariate normal distribution.

Before a point $\boldsymbol{x_i} \in \mathbb{R}^p$ is accepted as design point, we carry out a basic plausibility test on the maximum height of the associated morphology. Specifically, we first compute the sample mean $\hat{m}$ and sample standard deviation $\hat{s}$ of the set of maximum heights of the 14 original morphologies. Hence, we accept the parameter $\boldsymbol{x_i}$ if and only if the maximum height of the corresponding morphology is less than $\hat{m} + 4\hat{s}$. From the sequence $\{\boldsymbol{x_i}\}$, we extract the first 64 elements that satisfy this criterion, and run the corresponding climate simulations[17]. Two of the climate simulations crashed for unknown reasons and were consequently omitted from the analysis. The experimental design associated with the first wave of simulations thus consists of 62 elements.

5.5.2. Wave 2

The aim of the second wave of simulations is to test how the simulator $\delta^{18}O$ response is affected by small changes in the morphologies. For this, we undertake seven additional simulations, corresponding to morphologies specially designed for the task. Hence, a total of $n = 62 + 7 = 69$ design points $\boldsymbol{x_i} \in \mathbb{R}^p$ form the full experimental design. One of the additional design morphologies is chosen on the basis of the results provided by the six emulators calibrated on the first wave of simulations only. While details on the emulator calibration will be provided in the next Section 5.6, we illustrate here how the emulator results from the first wave inform the choice of the additional design points/morphologies, in order to give now a full overview of the complete experimental design used in the rest of this work.

Details of the seven additional design points $\boldsymbol{x_i}$ and morphologies $\boldsymbol{M_i}$ associated to the second wave of simulations ($i = 62, \ldots, n = 69$) are given below. Reasons for the choices are provided immediately after.

[17] The process of running the simulations was not carried out by myself, the author of this work. I would like to renew here my thanks to Irene Malmierca, PhD student at the British Antarctic Survey, for carrying out the simulations and extracting the simulated $\delta^{18}O$ values at the six locations of interest.

1. The design point $\boldsymbol{x_{63}}$ maximises the first-wave emulator probability of hitting the data intervals provided in Table 5.1, where the lower bounds for Camp Century and DYE3 are set to 1‰ and 2‰, respectively. The design point $\boldsymbol{x_{63}}$ happens to be significantly closer to $\boldsymbol{x_{60}}$ than to any of the other design points of the first wave ($\boldsymbol{x_i}$ for $i = 1, \ldots, 62$).

2. The design point $\boldsymbol{x_{64}}$ is chosen on the straight line between $\boldsymbol{x_{60}}$ and $\boldsymbol{x_{63}}$, with distance from $\boldsymbol{x_{63}}$ equal to twice the distance from $\boldsymbol{x_{60}}$.

3. The design point $\boldsymbol{x_{65}}$ is obtained as small perturbation of $\boldsymbol{x_3}$: the j^{th} component of $\boldsymbol{x_3}$ is perturbed by an instance of a uniform random number between $-\sigma_j/10$ and $\sigma_j/10$ (values of the PC standard deviations σ_j shown in Table 5.3).

4. The design points $\boldsymbol{x_{66}}$ and $\boldsymbol{x_{67}}$ are obtained, respectively, from $\boldsymbol{x_{26}}$ and $\boldsymbol{x_{50}}$, through the same procedure used to obtain $\boldsymbol{x_{65}}$ from $\boldsymbol{x_3}$.

5. The design point $\boldsymbol{x_{68}}$ is essentially identical to $\boldsymbol{x_{60}}$ (height difference between $\boldsymbol{M_{68}}$ and $\boldsymbol{M_{60}}$ always smaller than 4 cm).

6. The design point $\boldsymbol{x_{69}}$ is essentially identical to $\boldsymbol{x_{63}}$ (height difference between $\boldsymbol{M_{69}}$ and $\boldsymbol{M_{63}}$ always smaller than 0.3 mm).

Choice 2 aims at assessing the changes in the simulator response to inputs that are one, two, and three units apart from each other. Choices 3 and 4 intend, more generally, to assess changes in the simulator response to relatively small perturbation of inputs which belong to different regions of the input space. Finally, choices 5 and 6 are adopted to test the chaoticity of the system. That is, we want to test whether morphologies that are completely equivalent from the physical point of view, but not identical, yield essentially indistinguishable outputs via the climate model. The values in Table 5.4 show, unequivocally, that this is not the case. On this basis, we introduce observational variance during the emulator calibration phase, as the following section explains.

In Figure 5.10, we show the results of the n design simulations, in per-mille anomalies with respect to PI. Colour bands identify the ranges of $\delta^{18}O$ anomalies from records, as provided in Table 5.1. It can be appreciated that, for each pair of locations, there

	NEEM $\delta^{18}O$ ‰	**NGRIP** $\delta^{18}O$ ‰	**GRIP** $\delta^{18}O$ ‰	**GISP2** $\delta^{18}O$ ‰	**CC** $\delta^{18}O$ ‰	**DYE3** $\delta^{18}O$ ‰
$\boldsymbol{x_{60}}$ **Output**	3.51	3.17	2.08	2.72	2.61	4.60
$\boldsymbol{x_{68}}$ **Output**	2.92	2.65	1.87	2.46	2.37	4.14
$\boldsymbol{x_{63}}$ **Output**	2.94	2.43	1.94	2.59	2.42	4.75
$\boldsymbol{x_{69}}$ **Output**	3.00	2.60	2.17	2.83	1.63	4.62

Table 5.4: Per-mille $\delta^{18}O$ outputs of two pairs of simulations, performed to investigate potentially chaotic behaviour of the simulator. Each pair is run on physically identical input morphologies (surface height smaller than 4 cm for the top pair, and 0.3 mm for the bottom pair). Results differ significantly (compare to overall ranges of simulated $\delta^{18}O$ anomalies in Figure 5.10), confirming the chaotic nature of the simulations.

are always design simulations matching the records at both sites. Further inspection reveals that there are eight simulations which match the records on five of the six sites simultaneously, while there is none that matches the records at all six sites.

5.6. Calibration of the Six Emulators

On the basis of the results provided by the n design simulations, we fit one emulator at each of the six sites of interest. We follow the same procedure, independently, to fit each emulator: below we explain and justify the choices we make. As done so far, we denote by $\boldsymbol{x_i} \in \mathbb{R}^p$ the n design points, and further denote by $y_i \in \mathbb{R}$ the corresponding simulated $\delta^{18}O$ anomalies, $i = 1, \ldots, n$.

5.6.1. Mean and Covariance Functions

To specify the prior emulator mean, we need to choose a set of basis functions of the inputs $\boldsymbol{x} \in \mathbb{R}^p$ (recall equation (2.4) in Chapter 2). To decide on the form of the basis functions, we perform, independently at each site, a multiple linear regression which explains y_i as linear combination of the p components of the vector $\boldsymbol{x_i}$. In all six cases,

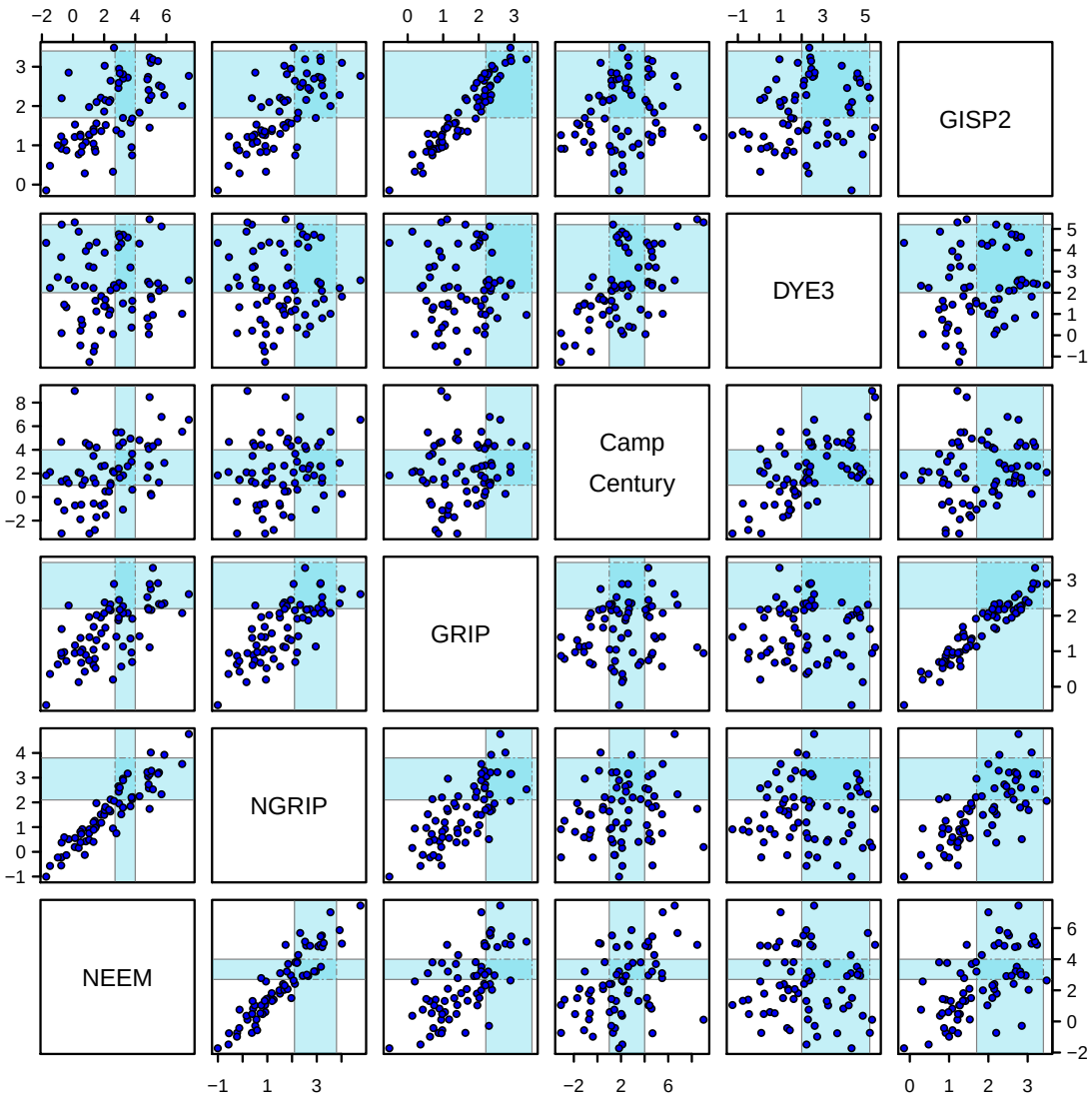


Figure 5.10: Simulated per-mille $\delta^{18}O$ anomalies for the $n = 69$ design morphologies. Results are shown for pairs of locations. Shaded light blue bands correspond to the ranges reconstructed from ice-core records, using the minimum/maximum values shown in Table 5.1.

the linear model returns unstructured residuals and adjusted-R^2 always greater than 0.89. In light of the fit, we choose a linear basis, $\boldsymbol{h}(\boldsymbol{x}) = (1, \boldsymbol{x}^T)^T \in \mathbb{R}^q$, with therefore $q = p + 1 = 9$. Equivalently, we specify the prior mean as follows:

$$m(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p, \quad \boldsymbol{x} \in \mathbb{R}^p. \tag{5.14}$$

The reader may recall that the idea of first exploring a simple linear regression was as well used in the development of the emulators of Chapter 4. In that case, however,

the fit was poor, and highlighted the need to include quadratic components. In the case of this chapter, a linear fit is on the contrary highly satisfactory.

As prior covariance function $c(\cdot, \cdot)$ (recall equation (2.5) in Chapter 2), we use the squared exponential (reasons for the choice given in Remark 5.1 below), with additional observational variance – the so-called "nugget" term, Section 2.8. That is, we consider:

$$c(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left(-(\boldsymbol{x} - \boldsymbol{x'})^T \boldsymbol{D}^{-1} (\boldsymbol{x} - \boldsymbol{x'})\right) + \nu\, \delta_{\boldsymbol{x},\boldsymbol{x'}}, \quad \boldsymbol{x}, \boldsymbol{x'} \in \mathbb{R}^p, \tag{5.15}$$

where $\boldsymbol{D}$ is the $p \times p$ diagonal matrix whose diagonal elements are the squared correlation lengths ${d_1}^2, \ldots, {d_p}^2$, and $\delta_{\cdot,\cdot}$ is the Kronecker delta function in (2.109). The procedure to estimate simultaneously the correlation lengths d_j and the nugget ν is discussed in Subsection 5.6.2. Before detailing this, we make two remarks about choice (5.15).

Remark 5.1. Correlation functions other than the squared exponential were tested during the calibration phase of the emulator. Specifically, we carried out experiments using the absolute exponential correlation function (expression (1.59)), and the Matérn correlation functions with parameter $\nu = 3/2$ and $\nu = 5/2$ (expressions (1.57) and (1.58); notice that the parameter ν is here clearly distinct from the nugget term used in (5.15)). In all cases, the mean and variance of the emulator predictions, as well as cross-validated estimates of the emulation error, were very similar. Given the apparent linear response highlighted at the beginning of this section, we chose the smoothest covariance function, *i.e.*, the squared exponential.

Remark 5.2. The choice of adding a nugget term was made on the basis of the second wave of simulation results, which highlighted a chaotic simulated response of the climate model: compare to the values in Table 5.4 and the discussion is Subsection 5.5.2. The emulator fitted after the first wave of simulations, and whose results were used to identify the design point $\boldsymbol{x_{63}}$ as explained in Subsection 5.5.2-point 1, was built following the same procedure presented here, with the only constraint of zero observational variance ($\nu = 0$).

5.6.2. Estimation of Correlation Lengths and Nugget Term

To complete the emulator calibration, the correlation lengths $d_j > 0$ $(j = 1, \dots, p)$ and the nugget term $\nu > 0$ appearing in equation (5.15) must be estimated. We detail below the procedure we use.

First, we observe from Figure 5.7 that all PCs have comparable size, as a consequence of being normalised with respect to the same scalar product. This implies that changes of similar magnitude in any two components of an input $\boldsymbol{x} \in \mathbb{R}^p$ yield changes of similar magnitude in the corresponding ice morphologies, built via the linear combination in equation (5.10). Following this reasoning, we assume the same correlation length for all the eight dimensions of the input space. That is, we impose:

$$d_1 = \cdots = d_p =: d\,. \tag{5.16}$$

This reduces the total number of parameters to estimate from nine to two.

At this point, to estimate the pair (d, ν), we employ a similar procedure to the one used to estimate the nugget and correlation lengths of the emulators of Chapter 4: maximise a cross-validated estimate of the emulator density of the data $(\boldsymbol{x_i}, y_i)$. This time, however, we do not use any prior distribution on the pair (d, ν). Specifically, for each $i = 1, \dots, n$, we consider $\rho^{(i)}_{d,\nu}(\cdot)$, the posterior density function at $\boldsymbol{x_i}$ of the emulator built on the data set where the pair $(\boldsymbol{x_i}, y_i)$ has been left out. Hence, we evaluate this density on the known output y_i, and consider the log-likelihood:

$$g(d, \mu) = \log\left(\prod_{i=1}^{n} \rho^{(i)}_{d,\mu}(y_i)\right) = \sum_{i=1}^{n} \log\left(\rho^{(i)}_{d,\mu}(y_i)\right). \tag{5.17}$$

The function $g(\cdot, \cdot)$ represents a cross-validated measure of the goodness of the emulator fit, when correlation lengths d and nugget term ν are used. We therefore choose the parameters d and ν which maximise the function g.

The maximisation is carried out using the Matlab nonlinear solver `fminunc`. The `fminunc` solver performs unconstrained optimisation, hence the maximisation is carried out on the logarithm of the variables d and ν. We perform the maximisation multiple times, from different starting points, in order to minimise the risk of only identifying

	NEEM	NGRIP	GRIP	GISP2	CC	DYE3
Corr. Length (d)	208.14	309.40	397.58	322.93	3,117.3	513.40
Nugget Term (ν)	0.45	0.35	0.28	0.53	2.54×10^{-4}	0.09

Table 5.5: Location by location, the values of the correlation length (d) and the nugget term (ν) which are used to build the corresponding emulator. The estimation is carried out through the LOOCV procedure explained in Subsection 5.6.2.

a local extremum.

We report in Table 5.5 the estimated values of the correlation length d and nugget ν, for the six locations of interest. With the only exception of Camp Century, the estimated d are of comparable magnitude to the standard deviations σ_j of the first PCs, as shown Table 5.3. In the case of Camp Century, the estimation returns instead a relatively large d. After considerations, we decided to retain the value: as we have discussed in Subsection 2.8.2, the large d denotes that the emulator model approaches a linear model, which would still represent an appropriate limit case given the considerations, in Subsection 5.6.1, on the apparent underlying linear response of the simulator. The emulators at all six sites are validated in the next section.

5.6.3. Emulator Validation

Before using the emulators as stochastic surrogates of the simulator, it is important to validate them. To assess the capability of each emulator to make correct predictions, we appeal again to the idea of LOOCV, using, however, a different measure than the one in (5.17): we consider standardised cross-validated residuals for each site.

For each $i = 1, \ldots, n$, we remove the i^{th} observation $(\boldsymbol{x_i}, y_i)$ from the data set, and fit an emulator with nugget and correlation length values given by Table 5.5. Let $\widehat{y}_i$ be the emulator prediction for the left out element of the data set, and $\widehat{s}_i$ the emulator standard deviation associated with the prediction. We then define the standardised residual as follows:

$$\widetilde{\varepsilon}_i = \frac{\widehat{y}_i - y_i}{\widehat{s}_i}\,, \qquad i = 1, \ldots, n\,. \tag{5.18}$$

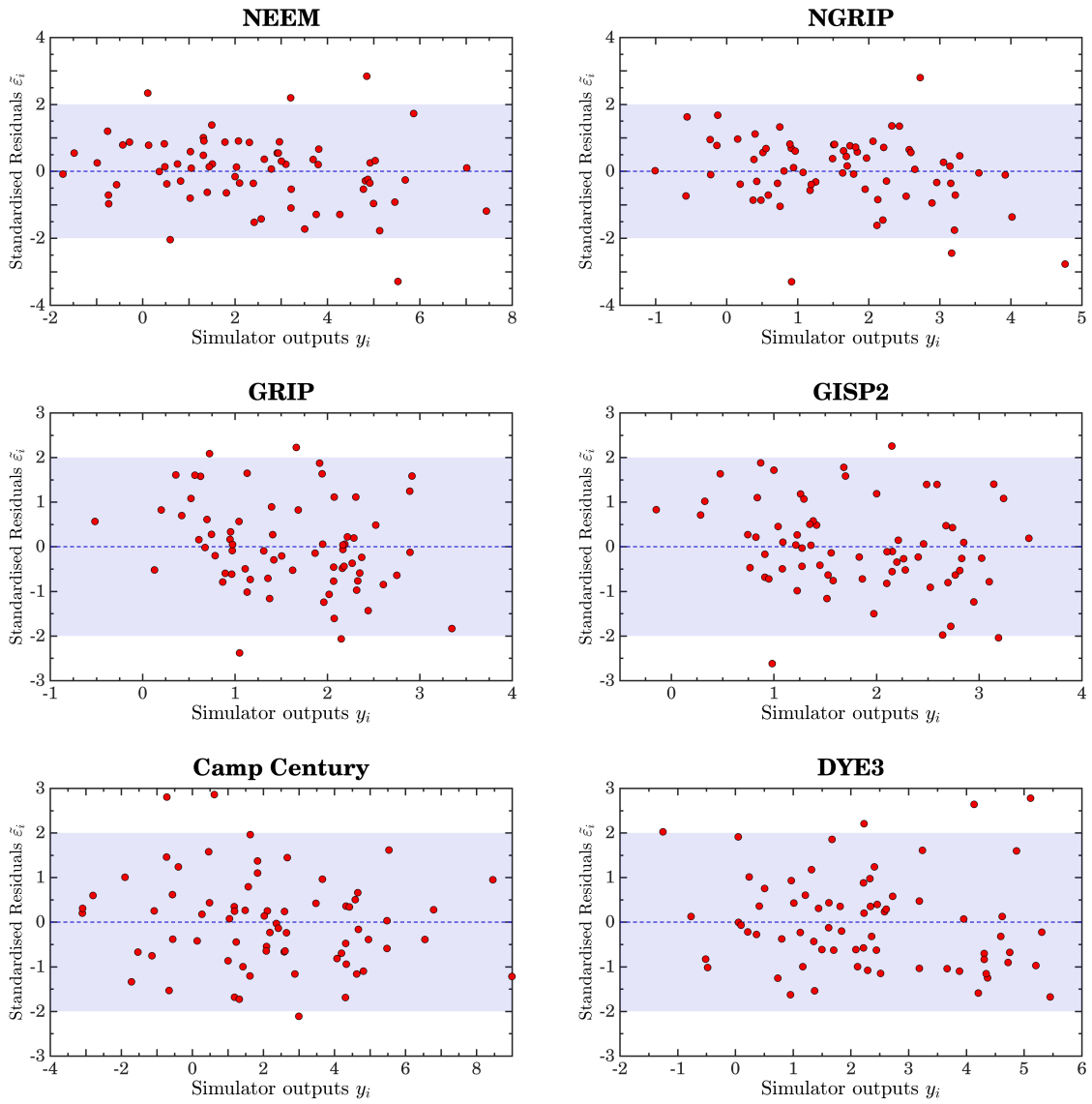


Figure 5.11: Cross-validated standardised residuals of the emulators, as defined in equation (5.18), as function of the corresponding simulator outputs. Shaded bands highlight values between -2 and 2, where around 95% of the standardised residuals are expected to lie.

In the terminology of Bastos and O'Hagan [2009], $\widetilde{\varepsilon}_i$ would be called (cross-validated) individual prediction error. Given the emulator distribution (t-Student with $n - q = 69 - 9 = 60$ degrees of freedom), we expect the standardised residuals of a good emulator to be distributed, roughly, as an independent sample of a standard normal.

In Figure 5.11 we show the plot of the standardised residuals as function of the simulator outputs, for all six emulators. The result is satisfactory. The shaded bands

highlight the interval $[-2, 2]$, where we expect to find around 95% of the standardised residuals. In a sample of size 69, this translates into approximately 3-4 elements outside the band, as the case of all the panels in Figure 5.11 is (only NEEM displays five elements outside the band, with one element very close to the border). Moreover, no particular structure can be identified in the plots. The only exception may be represented by the GRIP and GISP2 plots, two very close sites, where a trace of decreasing relationship may be identified in the lower part of the plots. The sign is however weak, and the relatively small number of design points does not exclude that the behaviour may be ascribed to intrinsic randomness.

Overall, our validation suggest that all the six emulators built represent reliable statistical models, which can not only make accurate predictions, but also accurately assess the uncertainties around them.

5.7. Identifying Record-Compatible Morphologies

The emulators built and validated in Section 5.6 allow to efficiently predict, at the six locations where ice-core records are available, the $\delta^{18}O$ simulated response to any ice sheet morphology. In this section we illustrate how we compare these predictions to the records, in order to identify morphologies that are compatible with them. Such a procedure is called history matching, a term, originally, coming from the oil industry (compare to Subsection 2.1.1 in Chapter 2). Clearly, to accomplish the aim, we need to take into account both the uncertainty affecting the emulator predictions and the one affecting the $\delta^{18}O$ records.

To carry out the comparison, we use the so-called implausibility measure (details below), which quantifies the extent to which a given morphology matches the records. To the best of the author's knowledge, the idea of an implausibility measure was first introduced in Craig et al. [1997], in the context of hydrocarbon reservoirs. Since then, it has been employed in a variety of other areas, such as galaxy formation (Vernon et al. [2010]) or epidemiology (Andrianakis et al. [2017]).

Albeit applied to different contexts, the previous works all share the same framework: a computer simulator and historical observations are available, and the aim is to locate

a region of the simulator input space whose corresponding outputs match the observations. This aim is usually achieved through several "waves" of simulations, as explained in the following. After a first wave, an emulator is built and an implausibility measure is employed to rule out regions of the input space where the emulator predictions and the observed data are unlikely to be compatible. Hence, additional simulations are run within the "not-ruled-out-yet" (NROY) region, and a new emulator is trained which will give more precise predictions in the region. This procedure (known as *refocusing*, Craig et al. [2001]) is iterated a number of times, till a sufficiently small region is identified.

In the present work, carrying out the refocusing procedure described above is unfeasible, due to the notable amount of time that each simulation requires (15-20 days). Hence, after a single wave, we use the implausibility measure to directly select morphologies which are likely to be compatible with the records, rather than to discard incompatible ones and later refocus. We provide the details of the measure and of our procedure below.

Let $\hat{m}_{L}(\boldsymbol{x})$ and $\hat{v}_{L}(\boldsymbol{x},\boldsymbol{x})$ be the mean and variance of the $\delta^{18}O$ emulator prediction at the ice-core location L, for the morphology associated with input $\boldsymbol{x} \in \mathbb{R}^p$. Further, denote by R_{L} the most likely ice-core record $\delta^{18}O$ anomaly at the location, and by R_{L}^{-} and R_{L}^{+} the associated lower and upper bounds: for all locations, values of $R_{L}^{-} < R_{L} < R_{L}^{+}$ are provided in Table 5.1. The idea behind the implausibility measure is to quantify the mismatch between the emulator mean prediction $\hat{m}_{L}(\boldsymbol{x})$ for input $\boldsymbol{x} \in \mathcal{P}$ and the ice-core record R_{L}, relative to the uncertainty affecting both sources. Computing this as ratio between the mean and the standard deviation of the "random variable" emulator minus record, independence of the two sources reasonably leads to the following definition:

$$I_L(\boldsymbol{x}) = \frac{\left|\hat{m}_{L}(\boldsymbol{x}) - R_{L}\right|}{\sqrt{v_{L}(\boldsymbol{x},\boldsymbol{x}) + \mathrm{Var}(\mathrm{Rec}_{L})}}\,. \tag{5.19}$$

The quantity $\mathrm{Var}(\mathrm{Rec}_{L})$ is meant to provide a measure of the "variance" associated with the ice-core record. Clearly, information about the record does not come in the form of an abstract random variable. However, based on the analogy that the variance

of a uniform random variable on an interval of length l is $l^2/12$, we compute the term as follows:

$$\mathrm{Var}(\mathrm{Rec}_{L}) = \begin{cases} \left(R_{L}^{+} - R_{L}\right)^2/3 & \text{if } \hat{m}_{L}(\boldsymbol{x}) > R_{L} \\ \left(R_{L} - R_{L}^{-}\right)^2/3 & \text{if } \hat{m}_{L}(\boldsymbol{x}) < R_{L} \end{cases} . \tag{5.20}$$

The factor 1/3 rather than 1/12 is a consequence of the fact that, in equation (5.20), we deal with the two subintervals rather than the whole interval. In particular, if the interval is symmetric ($R_{L}^{+} - R_{L} = R_{L} - R_{L}^{-}$), then we recover the variance of the uniform distribution. We note that an additional term is sometimes included in the denominator of equation (5.19), accounting for discrepancy between the simulator and the physical process being simulated (*e.g.*, Vernon et al. [2010], Williamson et al. [2013]).

Equation (5.19) provides a measure of the emulator-record mismatch at location L. In order to provide a comprehensive measure of the mismatch at all locations, we define

$$I(\boldsymbol{x}) = \max\{I_{L_1}(\boldsymbol{x}), \ldots, I_{L_6}(\boldsymbol{x})\}, \tag{5.21}$$

where $L_1, \ldots, L_6$ are the six ice-core sites in question. We then classify a morphology represented by $\boldsymbol{x} \in \mathbb{R}^8$ as record-compatible (RC), if the following two conditions hold:

1. $I(\boldsymbol{x}) < 2$;

2. At least 95% of the morphology grid cells, weighted by their respective areas, have an height between $\widetilde{m} - 2\tilde{s}$ and $\widetilde{m} + 2\tilde{s}$, where $\widetilde{m}$ and $\tilde{s}$ represent the sample mean and standard deviation of the set of heights of the N original morphologies at the cell.

The combination of both criteria ensures that RC morphologies pass a general physical plausibility test, in addition to being compatible with the available ice-core records. In the Matlab Appendix F.3, we report the two functions separately implementing each of the two previous conditions.

5.8. Results

In this section we illustrate the results of our comparison between emulator predictions and ice-core records, in terms of plausible GrIS morphologies which match the records. We carry out the comparison in three different scenarios, as explained in the following subsection.

5.8.1. A Scenarios-Based Approach

In order to identify RC morphologies via the implausibility measure $I(\cdot)$ in (5.21), values of $R_L^- < R_L < R_L^+$ must be specified at all six ice-core sites. As explained in Section 5.2, however, lower bounds for the $\delta^{18}O$ anomalies at Camp Century (CC) and DYE3 cannot be easily inferred from the records. While Table 5.1 reports the present-day value (*i.e.*, a zero anomaly) as lower bound at these two sites, in this section we acknowledge the aforementioned uncertainty by considering three different scenarios. We name these according to how close the CC and DYE3 lower bounds are set to the central estimates at the two sites ($R_{\text{CC}} = +2.5‰$ and $R_{\text{DYE3}} = +4.7‰$).

1. Loose scenario (only imposing non-negative anomalies, as Table 5.1):

$$R_{\text{CC}}^- = 0‰, \quad R_{\text{DYE3}}^- = 0‰. \tag{5.22}$$

2. Middle scenario:

$$R_{\text{CC}}^- = 1‰, \quad R_{\text{DYE3}}^- = 2‰. \tag{5.23}$$

3. Tight scenario ($R_L^- = R_L - 1‰$):

$$R_{\text{CC}}^- = 1.5‰, \quad R_{\text{DYE3}}^- = 3.7‰. \tag{5.24}$$

In Subsection 5.8.2 and Subsection 5.8.3, we investigate and compare properties of the RC morphologies in the three scenarios.

5.8.2. Posterior Densities (Record-Compatible Morphologies)

In Subsection 5.4.2, we introduced a prior distribution on the set of all morphologies, here identified with $\mathbb{R}^p$, $p = 8$: the prior is multivariate normal, as specified in (5.12). In our analysis, we represent the prior distribution by a sample of $N = 10^7$ morphologies, drawn randomly from (5.12). Each of these N morphologies is classified as either being RC or not, according to the criterion described in Section 5.7. The resulting RC morphologies hence form a sample from the posterior distribution, which incorporates the constraints from the data (ice-core records). In this section we analyse the posterior in the three scenarios introduced above, and compare it to the prior distribution.

In the loose scenario, where the imposition of the record compatibility at Camp Century and DYE3 has little effect, the RC morphologies represent 7.37% of the morphologies sampled from the prior distribution. The percentage reduces when tighter constraints are imposed: it approximately halves (3.74%) in the middle scenario, and further decreases to 0.95% in the tight scenario.

Figure 5.12 illustrates how the posterior distribution compares to the prior, in the loose and tight scenario cases. While all these distributions are eight-dimensional, we plot two-dimensional sections of the subspace generated by the first three PCs to ease the interpretation. The grey shaded background shows the prior Gaussian density. The coloured lines instead represent contours of the posterior densities (red for tight scenario, blue for loose scenario), with labels indicating the percentage of RC morphologies that are selected from the original prior sample. As a general pattern, it can be appreciated that the posterior distribution in the loose scenario is wider and closer to the prior than the posterior distribution in the tight scenario. This is not surprising. However, the particular directions in which the shifts from prior to posterior happen, alongside the PC shapes, are informative of the main patterns that RC morphologies may display.

Particularly in the tight scenario case, we see that areas characterised by a higher density of RC morphologies tend to have a positive first PC score (this is clearly evident in the subplot of PC1-PC3, but can also be observed in the PC1-PC2 subplot). In light of the "North-South" pattern shown by the first PC, this seems to suggest the following: imposing the compatibility with records leads to ice morphologies with

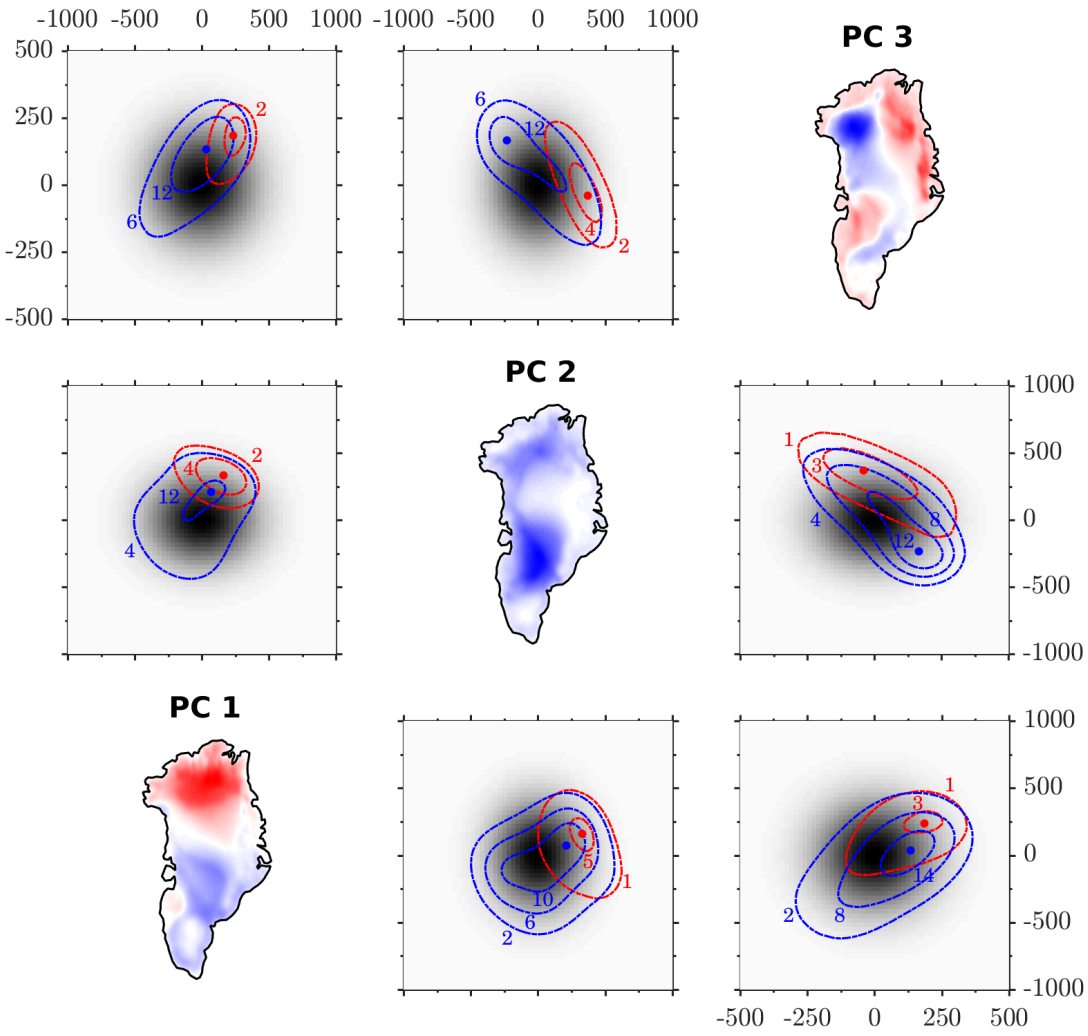


Figure 5.12: Comparison between prior and posterior densitites in the morphology space, represented by $\mathbb{R}^8$. For convenience, two-dimensional projections are shown, in the subspace generated by the first three PCs. Along the diagonal: illustration of the PCs. Off-diagonal plots: Gaussian prior illustrated by shaded grey background; contours of the posterior density shown in the case of the loose (blue) and tight (red) scenarios. Labels indicate the percentage of RC morphologies, along the specified contour. Different contour levels are shown in symmetric plots.

lower surface heights in the south and higher surface heights in the north, than the average morphology generated through the prior.

Similarly, the pair of PC2-PC3 subplots in Figure 5.12 reveals critical information. We see that PC2 scores of RC morphologies are for the vast majority positive in the tight scenario (red), and seem to be generally negative in the loose scenario (blue). To

interpret this information, notice that the second PC has almost everywhere negative values, particularly in the west-southern block of Greenland. Hence, the previous consideration suggests that the strong constraint on DYE3 and Camp Century imposed in the tight scenario induces a remarkable loss of ice in the west-southern block. However, when almost no constraint at DYE3 and Camp Century is imposed (loose scenario), the typical surface elevation in the south increases, reflecting the presence of more ice.

We hypothesise here that the cause of significantly lower south elevation in the tight scenario is to be mainly ascribed to the constraint provided by the DYE3 record, rather than to the one provided by Camp Century. Notice, indeed, the geographical position of the two sites (Camp Century, north; DYE3, south), and that the record at the DYE3 location is remarkably high (4.7‰ anomaly). In a scenario where the uncertainty around this last value is significantly reduced, as the case of our tight scenario is, it would not be surprising that morphologies compatible with the record were characterised by remarkable ice loss near the site. This hypothesis could be tested by running a separate analysis where only the uncertainty concerning the DYE3 record is reduced, as opposed to the analysis we have presented here which simultaneously tightens the uncertainty of the Camp Century and DYE3 records.

5.8.3. Shape and Uncertainty of RC Morphologies

In this final part of Section 5.8, we investigate directly physical characteristics of RC morphologies, and how these compare to general characteristics of prior morphologies. We mainly refer to Figure 5.13. This shows both mean and standard deviation, computed grid cell by grid cell, of the two sets of prior and RC morphologies, in the three scenarios.

We start by looking at the top row, which illustrates the variability characterising the different sets of morphologies. In the prior case (left), most of the variability, and thus of the uncertainty, is displayed within two regions: one in the north, one in the centre-south. We can appreciate how the variability in both regions is significantly reduced once the constraints from the data are taken into account. However, whilst little difference is highlighted in the north between the three scenarios, the imposition

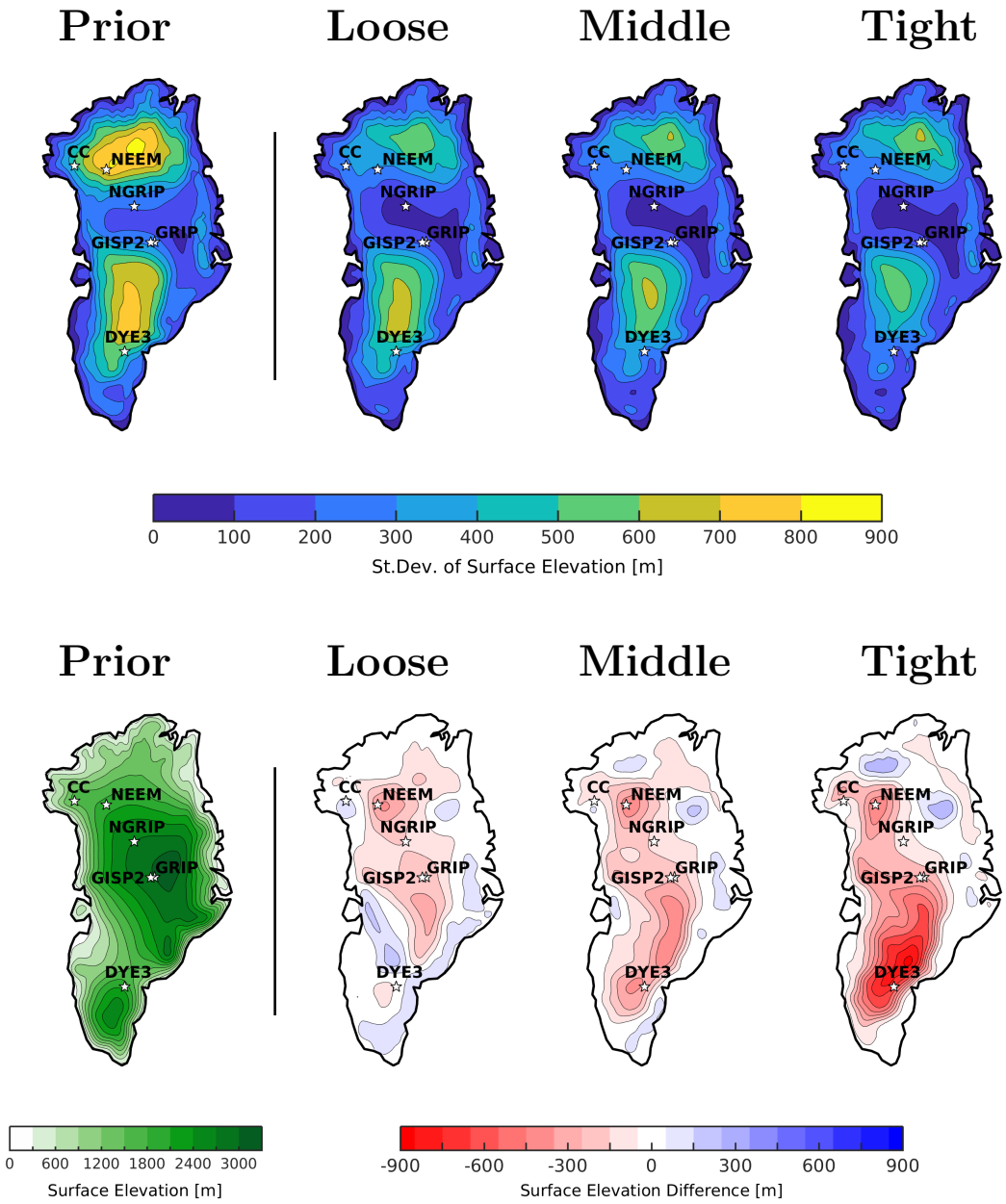


Figure 5.13: Plots represent the cell-by-cell standard deviation (top row) and average (bottom row) of the surface elevation of different sets of morphologies. $N = 10^7$ morphologies sampled from the prior distribution are used in the plots on the left of the vertical lines; only the subset of these which are RC are used for the posterior, in the three scenarios on the right. To ease visual interpretation, in the bottom-right plots (concerning means) we show the difference between posterior and prior. The loose scenario corresponds to the best current data, the middle and tight scenarios illustrate results which could be achieved, if better constraints on DYE3 and Camp Century were available.

of tighter record constraints clearly reduces the uncertainty on the southern surface elevation of RC morphologies.

To investigate the directions in which a higher degree of confidence in reconstructed surface elevation is attained, when going from looser to tighter scenarios, we can look at the prior and posterior average morphologies, in the bottom row. To ease the interpretation, we replace the raw posterior means with their difference from the prior. In the loose scenario, the compatibility with records imposes lower-than-prior average surface height in the central and northern parts of Greenland. It is interesting to notice that, in this case, the region characterised by lower elevations goes through the four sites where a non-negligible constraint is imposed (NEEM, NGRIP, GRIP, GISP2), but does not include DYE3 and Camp Century. This changes in the other two scenarios. What mainly characterises the transition from the loose, to the middle, to the tight scenario is a progressive and distinct decrease of the southern surface elevation. In the last case, remarkable ice loss with respect to the prior is displayed in the area around and just north of the DYE3 site.

In Subsection 5.4.3 we have explained how to associate a land-ice mask to a given morphology. Visual inspection of the masks associated with morphologies compatible with the tight scenario reveals that the majority of them presents a two-dome structure, with a large ice-dome in the north and a smaller one covering DYE3 in the south. A similar structure has been conjectured by other studies (Calov et al. [2015], Langebroek and Nisancioglu [2016]). Our emulator-based approach shows that such a pattern for the LIG GrIS may be recovered, if better constraints on the Camp Century and (especially) DYE3 records were available.

5.9. Conclusions

In this chapter, we have tackled a problem of central importance in paleoclimate: the one of reconstructing the past Greenland ice sheet (GrIS) morphology, specifically during the Last Interglacial period. The raising interest in the problem is motivated by the uncertainties surrounding the estimates of future sea-level rise, and the contribution to the latter stemming from the melting of the GrIS.

To approach the problem, we have merged information from ice-core records and climate simulations, using the statistical setting of GP emulation to compensate for the simulator expensiveness. Specifically, we have independently built emulators, predicting the $\delta^{18}O$ response to different GrIS morphologies, at the six sites where ice-core records are available. We have then compared the emulator predictions, on a large sample of prior morphologies, to the available records, and identified record-compatible morphologies. Our scenario-based approach, performed to face the large uncertainty of some records, has allowed to demonstrate the importance of well-constrained records. It suggests that more certain records at the DYE3 and Camp Century sites are compatible with morphologies characterised by low surface elevation in the south; also, such records would remarkably reduce the elevation uncertainty in this area. In particular, there are glimpses that a better dated DYE3 record would be important to make more certain inference about the past Arctic ice sheet and climate.

We remark that, in this chapter, we have not directly touched upon the issue of translating our results concerning the past GrIS morphology into results on sea-level rise contribution. This is a complex issue, as the extensive literature in the field (partly reviewed in Subsection 5.1.1) reveals. In fact, there is no simple way to estimate the ice volume corresponding to a given morphology. The ice pressure on the underlying bedrock is enormous, and causes substantial decreases in the bedrock elevation. Estimating the extent of the compression for a general morphology is a challenging physics task: it depends on the elasticity of the rock at different locations, and of course on the amount of above ice in large areas around the location.

Results about approximate ice-volume losses corresponding to our three scenarios have been investigated by the author fairly in depth, under the simplistic assumption of fixed bedrock. The description of the methods and the corresponding results have been reluctantly omitted from this chapter due to the urgency of terminating the present work, as well as to limit the overall length of the latter. Nonetheless, we believe that the work illustrated in the chapter, accounting for various uncertainties via the use of emulation methods, may trigger further collaborations with rock modellers to obtain precise estimates of ice volumes corresponding to our scenarios. These, in turn, may be translated into estimates of the GrIS contribution to the last interglacial sea-level rise, both globally and locally. This still represents a highly debated topic, which

is recently benefiting of the contributions of researchers from an increasing number of disciplines, called together to face the drastic climate changes that our planet is experiencing.

Concluding Remarks

Contributions of This Work

In this thesis we have offered an overview of Gaussian process (GP) emulation: we have analysed its framework and properties in Part I and have provided two examples of its relevance in tackling climate problems in Part II.

During the last three decades, a large amount of works have been published in the fields of GP emulation and more generally Uncertainty Quantification (UQ). These have been reviewed at the beginning of Chapter 2. Due to obvious length and time constraints, not all such aspects could be presented in detail in this work. However, in Part I of this thesis, we have offered a thorough presentation of the theory behind the construction of GP emulators. To this aim, we have used a formal probabilistic setting as solid ground to justify all the steps involved in the construction of a GP emulator. The case where observational variance is added to the emulator has also been investigated. Departing from the literature, we have provided an interpretation of the relevant formulas as being consequence of a component split of the emulator, derived via continuity arguments.

Motivated by the need to reduce the dimension of the simulator output space before carrying out emulation, in Chapter 3 we have proposed a dimension-reduction technique which is appropriate to the nature of most climate simulator outputs. While operating on elements of $\mathbb{R}^s$, our procedure interprets these as elements of the Hilbert space of real functions defined on the sphere $S^2 \subseteq \mathbb{R}^3$, and therefore adapts classical PCA ideas to a geometry of relevance to the problem.

In Part II of this thesis, the stress of this work has been on applications of the

GP emulation framework to climate reconstruction problems. We have in particular tackled two problems, both of central importance for the climate community. In Chapter 4, we have focussed our attention on the mismatch between simulated ocean temperatures and ocean temperatures reconstructed from geological records, during the mid-Pliocene (around three million years ago). As opposed to previous works on the same topic, the use of the emulation framework has allowed us to incorporate in the analysis the effects of the varying orbital forcing characterising the mid-Pliocene. The nature of the emulator as stochastic process has allowed us to sample trajectories from its distribution, at random input times, and thus to synthetically replicate the sequence of steps employed by geologists to estimate the mid-Pliocene temperature from time series of geological records. This way, we have shown that part of the currently observed data-model mismatch can be ascribed to the orbital variability characterising the mid-Pliocene, by showing that the mismatch is indeed reduced when this is taken into account.

In a second example, in Chapter 5, a problem directly linked to the current sea-level rise issue has been undertaken: the one of reconstructing the morphology of the Greenland ice sheet, during the Last Interglacial. Thanks to the use of emulation techniques, this has been treated as an inverse problem: the employed simulator has been used to predict $\delta^{18}O$ anomalies, at six sites, corresponding to various morphologies; hence, the morphologies whose simulator outputs matched ice-core records have been examined. Our contribution here is twofold. On the statistical side, our approach shows an example where GP emulation is performed on a space of functions (representing ice morphologies), rather than on a finite-dimensional space of few, independent parameters which are tuned as simulator inputs. On the applied side, our approach has allowed to combine, for the first time, ice-core records and climate simulations in a comprehensive way, widely exploring the space of ice morphologies. We find that, especially in the South of Greenland, the records would suggest a remarkable reduction of ice with respect to previous reconstructions. Such a conclusion cannot however be reached with a sufficiently high level of confidence, till better constrained ice-core records at the DYE3 and Camp Century sites are available.

Future Directions of Investigation

At the end of each of the two chapters in Part II, we have highlighted future directions of investigation that this work opens up. Within the setting of Chapter 4, these consist in the possibility of using the stochastic nature of the emulator in conjunction with recent data sets of marine records, characterised by the fact that each record is associated with a past time, with relatively low uncertainty. Our sampling procedure could straightforwardly be adapted to this case, and would be able to naturally incorporate the unavoidable time uncertainties associated with the data. Such research could enable a better understanding of the climate models employed and allow the community to better identify potential biases in these, before they are used for forecasting.

Within the setting of Chapter 5, one line of great climatological relevance that our work opens up is the translation of our scenario-based compatible morphologies into estimates of scenario-based Greenland ice sheet contribution to the LIG sea-level rise. As mentioned at the end of the chapter, this calls for the collaboration with physicists and rock modellers, to account for the elastic behaviour of the bedrock below the ice sheets.

Various lines of further investigations may also be pursued under the purely statistical point of view. We notice, for example, that the classical choices of prior covariance functions used in emulation, introduced in Subsection 1.4.4, only allow positive correlations. Covariance functions allowing negative correlations are used in other contexts, such as in kriging: examples of trigonometric correlation functions, or of wave correlation functions displaying a damping and oscillating behaviour, are for example reported in Diggle and Ribeiro [2006]. However, to the best of the author's knowledge, the use of prior covariances allowing negative values has not been a topic of investigation within the GP emulation field.

A way to investigate this may be to define a measure of the goodness of GP emulator predictions on a given set of test functions, when different prior correlations are used. If $f(\boldsymbol{x})$ is a function, $\boldsymbol{x} \in \mathcal{P}$ where $\mathcal{P}$ is a bounded domain of $\mathbb{R}^s$, and $\eta_f(\boldsymbol{x})$ is an emulator of $f(\cdot)$ built on a specified covariance function, then a normalised measure

to test how well $\eta_f(\cdot)$ approximates $f(\cdot)$ may be the following:

$$\mu(f, \eta_f) = \frac{1}{|\mathcal{P}|}\,\mathbb{E}\left[\int_{\mathcal{P}} \Big(\eta_f(\boldsymbol{x}) - f(\boldsymbol{x})\Big)^2 d\boldsymbol{x}\right].$$

Weights may also be introduced, for example, to lower the mismatch impact at the boundaries of the space, if needed. Hence, the measure $\mu(\cdot,\cdot)$, or otherwise, can be used to evaluate whether one covariance function systematically outperforms another, when $f(\cdot)$ varies in a given space S of test functions. If this is a vector space, then the linearity of the map

$$T\colon f \mapsto \eta_f$$

and the good properties of $\mu(\cdot,\cdot)$ may also allow to work on a basis only. Notice, indeed, that

$$\mu(cf, \eta_{cf}) = c^2\mu(f, \eta_f) \quad \forall\, c \in \mathbb{R}\,,$$

and $\mu(f+g, \eta_{f+g})$ can be easily estimated in terms of $\mu(f, \eta_f)$ and $\mu(g, \eta_g)$ through Cauchy-Schwarz inequality.

Before concluding, we mention another topic which may be object of further theoretical investigation and which, at the same time, is of relevance in practical contexts. This concerns the potential rise of identifiability issues in models with particularly large correlation lengths. At the end of Chapter 2, we have only briefly touched upon the issue, but we believe that the problem needs further investigation. This should also aim at providing guidance in recognising and tackling the issue in practical contexts.

The previous point is only one of several examples which highlight the interplay between sound theoretical investigations and practical applications, and it is surely far from being the most relevant. This work in its entirety has aimed to stress the importance of both components when tackling real problems in research, with examples borrowed from my, the author's, direct and necessarily limited experience. Even more after the PhD, it is a strong belief of mine that the formalism and rigour that mathematics and statistics provide are *not only* beautiful and elegant "per se", but they are as well of crucial importance to provide the correct guidance to practitioners outside the maths community.

Appendix

A. Results from Probability

In the following we state two basic results from probability theory, which have been used in Chapter 2.

Lemma A.1. *Let I, $J \subseteq \mathbb{R}$ be two real intervals and let $g : I \to J$ be a diffeomorphism (i.e., $g \in C^1(I)$, g is invertible, and $g^{-1} \in C^1(J)$). Let X be a random variable on I with density $f_X : I \to [0, \infty)$, and consider the random variable $Y = g(X)$. Then, Y has density $f_Y : J \to [0, \infty)$ given by:*

$$f_Y(y) = \frac{f_X\big(g^{-1}(y)\big)}{\big|g'\big(g^{-1}(y)\big)\big|}\,. \tag{A.1}$$

Proof. We exploit the fact that a random variable Z with support S has density $f(z)$ if and only if

$$\mathbb{E}\big[h(Z)\big] = \int_S h(z)\, f(z)\, dz\,, \quad \forall\, h \in C(S, \mathbb{R})\,. \tag{A.2}$$

Hence, for $h \in C(J, \mathbb{R})$, we get:

$$\begin{aligned}
\mathbb{E}\big[\, h(Y)\,\big] &= \mathbb{E}\big[(h \circ g)(X)\big] \\
&= \int_I (h \circ g)(x)\, f_X(x)\, dx \\
&= \int_J h(y)\, f_X(g^{-1}(y))\, \frac{1}{\big|g'\big(g^{-1}(x)\big)\big|}\, dy\,,
\end{aligned}$$

where the last equality is obtained through the change of variables $y = g(x)$. Given the characterisation at the beginning, we see that (A.1) is indeed the density of Y. □

Lemma A.2. *Let* $(\widetilde{\boldsymbol{Y}}, \boldsymbol{Y}) \in \mathbb{R}^{k+n}$ *be a Gaussian vector, with mean and variance accordingly partitioned as*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}} \\ \boldsymbol{\mu}_{\boldsymbol{Y}} \end{pmatrix} \in \mathbb{R}^{k+n}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{Y}}} & \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\boldsymbol{Y}} \\ \boldsymbol{\Sigma}_{\boldsymbol{Y}\widetilde{\boldsymbol{Y}}} & \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Y}} \end{pmatrix} \in \mathbb{R}^{(k+n)\times(k+n)}. \tag{A.3}$$

Let $\boldsymbol{a} \in \mathbb{R}^n$ *be a fixed vector. Then, the conditional distribution of* $\widetilde{\boldsymbol{Y}}$ *given* $\boldsymbol{Y} = \boldsymbol{a}$ *is still Gaussian, with mean* $\boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}}^{cond}$ *and variance* $\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}}^{cond}$ *given by*

$$\boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}}^{cond} = \boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}} + \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\boldsymbol{Y}} \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Y}}{}^{-1} (\boldsymbol{a} - \boldsymbol{\mu}_{\boldsymbol{Y}}) \in \mathbb{R}^k, \tag{A.4.a}$$

$$\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}}^{cond} = \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{Y}}} - \boldsymbol{\Sigma}_{\widetilde{\boldsymbol{Y}}\boldsymbol{Y}} \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Y}}{}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{Y}\widetilde{\boldsymbol{Y}}} \in \mathbb{R}^{k\times k}. \tag{A.4.b}$$

Sketch of Proof. The proof only consists in applying the standard formula for conditional densities,

$$p(\widetilde{\boldsymbol{y}} \,|\, \boldsymbol{y}) = \frac{p_{\text{joint}}(\widetilde{\boldsymbol{y}}, \boldsymbol{y})}{p_{\text{marg}}(\boldsymbol{y})}\,, \tag{A.5}$$

to the case where:

- ▷ $p_{\text{joint}}(\cdot,\cdot)$ is the density of a Gaussian random vector with mean $\boldsymbol{\mu} \in \mathbb{R}^{k+n}$ and covariance matrix $\boldsymbol{\Sigma}$, as in (A.3);
- ▷ $p_{\text{marg}}(\cdot)$ is the density of a Gaussian random vector with mean $\boldsymbol{\mu}_{\boldsymbol{Y}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{Y}}$.

Hence, only (unpleasant) algebraic manipulations remain to be carried out, to recognise the exponent of the resulting ratio as a quadratic form in $\widetilde{\boldsymbol{y}} - \boldsymbol{\mu}_{\widetilde{\boldsymbol{Y}}}^{\text{cond}}$. To this aim, the formula to invert a symmetric block matrix $\boldsymbol{\Sigma}$ needs to be used. The reader interested in carrying out the calculations can find the formula in a number of undergraduate textbooks, for example Horn and Johnson [2012, §0.7.3]. □

B. Results from Linear Algebra

We denote by $GL_n(\mathbb{R})$ the General Linear group of order n on $\mathbb{R}$: that is, the set of all $n \times n$ real invertible matrices. The following Lemma has been used in Chapter 2, specifically in Proposition 2.4.2 and Section 2.6.

Lemma B.1. *Let $\boldsymbol{B} \in GL_q(\mathbb{R})$ and $\boldsymbol{A} \in GL_n(\mathbb{R})$. Let also $\boldsymbol{U}$ and $\boldsymbol{V}$ be two rectangular matrices,*

$$\boldsymbol{U} \in \mathbb{R}^{q\times n}, \quad \boldsymbol{V} \in \mathbb{R}^{n\times q}.$$

Then, the matrix $\boldsymbol{B} + \boldsymbol{UAV} \in \mathbb{R}^{q\times q}$ is invertible if and only if the matrix $\boldsymbol{A}^{-1} + \boldsymbol{VB}^{-1}\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is invertible. In this case, it holds:

$$(\boldsymbol{B}+\boldsymbol{UAV})^{-1} = \boldsymbol{B}^{-1} - \boldsymbol{B}^{-1}\boldsymbol{U}(\boldsymbol{A}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1}. \tag{B.1}$$

Proof. To prove the statement it is sufficient to check that the product between the matrix $\boldsymbol{B}+\boldsymbol{UAV}$ and the RHS of (B.1) returns the identity matrix of order q. Indeed, being both matrices of order q, also also their product in reverse order will have to give the identity matrix. For completeness, we report the computations below:

$$\begin{aligned}
&(\boldsymbol{B}+\boldsymbol{UAV})\left[\boldsymbol{B}^{-1} - \boldsymbol{B}^{-1}\boldsymbol{U}(\boldsymbol{A}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1}\right] \\
&\quad = \boldsymbol{I_q} + \boldsymbol{UAVB}^{-1} - \boldsymbol{U}(\boldsymbol{A}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1} \\
&\qquad - \boldsymbol{UAVB}^{-1}\boldsymbol{U}(\boldsymbol{A}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1} \\
&\quad = \boldsymbol{I_q} + \boldsymbol{UAVB}^{-1} \\
&\qquad - \boldsymbol{U}\left[\boldsymbol{I_n} + \boldsymbol{AVB}^{-1}\boldsymbol{U}\right](\boldsymbol{A}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1} \\
&\quad = \boldsymbol{I_q} + \boldsymbol{UAVB}^{-1} \\
&\qquad - \boldsymbol{U}\,\boldsymbol{A}(\boldsymbol{A}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})(\boldsymbol{A}^{-1}+\boldsymbol{VB}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VB}^{-1} \\
&\quad = \boldsymbol{I_q} + \boldsymbol{UAVB}^{-1} - \boldsymbol{UAVB}^{-1} = \boldsymbol{I_q}.
\end{aligned}$$

As explained at the beginning, this completes the proof. □

The following result is a classical one in linear algebra. In its simplest form, it ensures that a real symmetric matrix admits a basis of orthogonal eigenvectors. We provide

the following more general form, from Lang [1987, Chapter VIII, Theorem 4.3]. This has been used in Chapter 3, Theorem 3.3.1, when deriving PCA with respect to a general inner product on $\mathbb{R}^p$.

Theorem B.2 (Spectral Theorem)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{s\times s}$ be square matrix, and $\langle \cdot, \cdot \rangle$ be a positive definite inner product on $\mathbb{R}^s$. Suppose that $\boldsymbol{A}$, as linear operator from $\mathbb{R}^s$ to $\mathbb{R}^s$, is symmetric (self-adjoint) with respect to the inner product $\langle \cdot, \cdot \rangle$. That is, suppose that*

$$\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle \quad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^s\,. \tag{B.2}$$

Then, there exists a $\langle \cdot, \cdot \rangle$-orthonormal basis of $\mathbb{R}^s$, whose elements are eigenvectors of the matrix $\boldsymbol{A}$.

If $\mathcal{B} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_s}\}$ is the basis of the theorem, orthonormal means $\langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle = \delta_{ij}$.

C. Proof of Integrated Likelihood Formula

Below we provide the expression of the marginal integrated likelihood of a vector $\boldsymbol{y}$ with respect to the emulator posterior distribution, under the classical assumption of a non-informative prior discussed in Section 2.6. As mentioned in Section 4.8, this function is often maximised in the literature to estimate hyperparameters of the emulator model. Although we have not directly used it in this work, we provide here the expression and a proof of the formula, for technical reference.

The notation we use is the one of Chapter 2, which we briefly summarise as follows:

- $\boldsymbol{\beta} \in \mathbb{R}^q$, $\sigma^2 \geq 0$ are random;
- $\boldsymbol{H} \in \mathbb{R}^{n \times q}$ is a full rank matrix ($\text{rank}(\boldsymbol{H}) = q$ since we assume $q < n$);
- $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is a symmetric, positive definite matrix;
- $\boldsymbol{Y} \in \mathbb{R}^n$ is a Gaussian random vector, $\boldsymbol{Y} | \boldsymbol{\beta}, \sigma^2 \sim N(\boldsymbol{H}\boldsymbol{\beta}, \sigma^2 \boldsymbol{A})$.

In particular, the conditional density of $\boldsymbol{Y}$ given $\boldsymbol{\beta}, \sigma^2$ is as follows:

$$L^*\big(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2\big) = \frac{1}{(\sigma^2)^{n/2}\,|\boldsymbol{A}|^{1/2}} \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta})^T \boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\beta})\right]. \tag{C.1}$$

Proposition C.1. *Under the previous notation, assume an improper, non-informative prior for the pair* $(\boldsymbol{\beta}, \sigma^2)$*:*

$$\pi(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^2}\,. \tag{C.2}$$

Also assume that the vector $\boldsymbol{y} \in \mathbb{R}^n$ *has been observed as a single realisation of* $\boldsymbol{Y}$*. Then, the following formula for the integrated likelihood of* $\boldsymbol{Y}$ *holds true:*

$$\begin{aligned} L(\boldsymbol{y}) &= \int_{\mathbb{R}^q \times \mathbb{R}^+} \pi(\boldsymbol{y}, \boldsymbol{\beta}, \sigma^2)\, d\boldsymbol{\beta}\, d\sigma^2 = \int_{\mathbb{R}^q \times \mathbb{R}^+} L^*(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2)\, \pi(\boldsymbol{\beta}, \sigma^2)\, d\boldsymbol{\beta}\, d\sigma^2 \\ &\propto |\boldsymbol{A}|^{-1/2}\, |\boldsymbol{B}|^{-1/2} \big[(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})^T \boldsymbol{A}^{-1}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{b})\big]^{-\frac{n-q}{2}}\,, \end{aligned} \tag{C.3}$$

where $\boldsymbol{B} = \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{H}$ *and* $\boldsymbol{b} = \boldsymbol{B}^{-1} \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{y}$*, as in* (2.95.a) *and* (2.95.b)*.*

Proof. Given (C.1) and (C.2), the following holds:

$$L(\boldsymbol{y}) \propto \int_{\mathbb{R}^q \times \mathbb{R}^+} \frac{1}{(\sigma^2)^{n/2+1} |\boldsymbol{A}|^{1/2}} \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})^T \boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})\right] d\boldsymbol{\beta}\, d\sigma^2$$

$$= |\boldsymbol{A}|^{-1/2} \int_{\mathbb{R}^q} \left[\int_0^{+\infty} \frac{1}{(\sigma^2)^{\alpha}} \exp\left(-\frac{D(\boldsymbol{\beta})}{2\sigma^2}\right) d\sigma^2\right] d\boldsymbol{\beta}\,, \tag{C.4}$$

where $\alpha = n/2\ +1$ and $D(\boldsymbol{\beta}) = (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})^T \boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})$.

Now observe that, for any constant D, we have:

$$\int_0^{+\infty} \frac{1}{x^{\alpha}} \exp\left(-\frac{D}{2x}\right) dx \overset{z=\frac{x}{D}}{=} \frac{1}{D^{\alpha-1}} \int_0^{+\infty} \frac{1}{z^{\alpha}} \exp\left(-\frac{1}{2z}\right) dz = \frac{c(\alpha)}{D^{\alpha-1}}\,,$$

as long as the condition $\alpha > 1$ holds, to guarantee the convergence of the integral. Taking into account this result, from (C.4) we get the following:

$$L(\boldsymbol{y}) \propto |\boldsymbol{A}|^{-1/2} \int_{\mathbb{R}^q} \frac{1}{D(\boldsymbol{\beta})^{\alpha-1}}\, d\boldsymbol{\beta} = |\boldsymbol{A}|^{-1/2} \int_{\mathbb{R}^q} \frac{1}{\left[(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})^T \boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})\right]^{\alpha-1}}\, d\boldsymbol{\beta}\,. \tag{C.5}$$

Notice that the condition $\alpha > 1$ is indeed satisfied, since $\alpha = n/2\ +1$.
Let us now expand the integrand of (C.5). We have

$$(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})^T \boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta}) = \boldsymbol{\beta}^T \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{H}\boldsymbol{\beta} - 2\boldsymbol{y}^T \boldsymbol{A}^{-1} \boldsymbol{H}\boldsymbol{\beta} + \boldsymbol{y}^T \boldsymbol{A}^{-1} \boldsymbol{y}\,. \tag{C.6}$$

The matrix $\boldsymbol{B} = \boldsymbol{H}^T \boldsymbol{A}^{-1} \boldsymbol{H} \in \mathbb{R}^{q \times q}$ is symmetric and positive definite (clearly, $\boldsymbol{x}^T \boldsymbol{B} \boldsymbol{x} \geq 0$ for any $\boldsymbol{x} \in \mathbb{R}^q$, since $\boldsymbol{A}$ is positive definite). We can therefore consider its Cholesky decomposition, and write:

$$\boldsymbol{B} = \boldsymbol{R}^T \boldsymbol{R}\,,$$

where $\boldsymbol{R} \in \mathbb{R}^{q \times q}$ is upper triangular. In particular, $|\boldsymbol{R}| = |\boldsymbol{B}|^{1/2}$.

The matrix $\boldsymbol{R}$ is of course invertible (since $\boldsymbol{B}$ is), hence we can consider the following change of variable:

$$\boldsymbol{z} = \boldsymbol{R}\boldsymbol{\beta} \in \mathbb{R}^q\,, \tag{C.7}$$

in terms of which we can write the following:

$$\begin{aligned}(\boldsymbol{y}-\boldsymbol{H\beta})^T\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H\beta}) &\overset{(C.6)+(C.7)}{=} \boldsymbol{z}^T\boldsymbol{z}-2\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{R}^{-1}\boldsymbol{z}+\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y}\\ &= \boldsymbol{z}^T\boldsymbol{z}-2\widetilde{\boldsymbol{w}}^T\boldsymbol{z}+\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y}\\ &= (z-\widetilde{\boldsymbol{w}})^T(\boldsymbol{z}-\widetilde{\boldsymbol{w}})\ -\widetilde{\boldsymbol{w}}^T\widetilde{\boldsymbol{w}}+\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y}\\ &= (z-\widetilde{\boldsymbol{w}})^T(\boldsymbol{z}-\widetilde{\boldsymbol{w}})+\ \boldsymbol{y}^T\left[\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{B}^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\right]\boldsymbol{y}\\ &= (\boldsymbol{z}-\widetilde{\boldsymbol{w}})^T(\boldsymbol{z}-\widetilde{\boldsymbol{w}})+\ S(\boldsymbol{y})^2 \end{aligned} \tag{C.8}$$

For the sake of simplicity, we have denoted by $\widetilde{\boldsymbol{w}}$ the constant vector $\boldsymbol{R}^{-T}\boldsymbol{H}^T\boldsymbol{A}^{-1}\boldsymbol{y}$, and by $S(\boldsymbol{y})^2$ the quantity $\boldsymbol{y}^T\left[\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{B}^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\right]\boldsymbol{y}$.

Substituting (C.8) back into (C.5), we get

$$\begin{aligned}L(\boldsymbol{y}) &\propto |\boldsymbol{A}|^{-1/2}\int_{\mathbb{R}^q}\left[(\boldsymbol{y}-\boldsymbol{H\beta})^T\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{H\beta})\right]^{-(\alpha-1)}d\boldsymbol{\beta}\\ &\overset{\boldsymbol{z}=\boldsymbol{R\beta}}{=} |\boldsymbol{A}|^{-1/2}|\boldsymbol{R}|^{-1}\int_{\mathbb{R}^q}\left[(\boldsymbol{z}-\widetilde{\boldsymbol{w}})^T(\boldsymbol{z}-\widetilde{\boldsymbol{w}})+\ S(\boldsymbol{y})^2\right]^{-(\alpha-1)}d\boldsymbol{z}\\ &= |\boldsymbol{A}|^{-1/2}|\boldsymbol{B}|^{-1/2}\int_{\mathbb{R}^q}\left[\boldsymbol{z}^T\boldsymbol{z}+\ S(\boldsymbol{y})^2\right]^{-(\alpha-1)}d\boldsymbol{z}\\ &\overset{\boldsymbol{x}=\boldsymbol{z}/S(\boldsymbol{y})}{=} |\boldsymbol{A}|^{-1/2}|\boldsymbol{B}|^{-1/2}\int_{\mathbb{R}^q}\left[S(\boldsymbol{y})^2(\boldsymbol{x}^T\boldsymbol{x}+1)\right]^{-(\alpha-1)}S(\boldsymbol{y})^q\,d\boldsymbol{x}\\ &\propto |\boldsymbol{A}|^{-1/2}|\boldsymbol{B}|^{-1/2}S(\boldsymbol{y})^{2-2\alpha+q}\\ &\overset{\alpha=\frac{n}{2}+1}{\propto} |\boldsymbol{A}|^{-1/2}|\boldsymbol{B}|^{-1/2}\left(S(\boldsymbol{y})^2\right)^{-\frac{n-q}{2}}\end{aligned} \tag{C.9}$$

The last step left to prove the claim is showing that $S(\boldsymbol{y})^2$ can be rewritten as follows:

$$S(\boldsymbol{y})^2=(\boldsymbol{y}-\boldsymbol{Hb})^T\boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{Hb})\,. \tag{C.10}$$

This is a trivial check, starting from $S(\boldsymbol{y})^2=\boldsymbol{y}^T\left[\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{H}\boldsymbol{B}^{-1}\boldsymbol{H}^T\boldsymbol{A}^{-1}\right]\boldsymbol{y}$, and remembering the definitions of $\boldsymbol{B}$ and $\boldsymbol{b}$. Hence the proof is complete. □

MATLAB Code

Most of the code used in this thesis can be found at https://github.com/dariod89. Part of it is reported in this appendix. For brevity, we omit initial checks on inputs from the body of the following functions, although these are implemented in the actual code.

Details of all inputs and outputs of the functions are found within each script.

D. General Routines

D.1. Covariance Functions

The following code evaluates the covariance functions of Subsection 1.4.4 on different pairs of inputs, under the convention of equation (1.54) and with norm as in (1.61). The string `fun` will specify which covariance function to use. A nugget term `nu` may also be specified (equation (2.108)). Each pair of inputs consists one row of the matrix $\texttt{X} \in \mathbb{R}^{n\times k}$ and one row of the matrix $\texttt{Y} \in \mathbb{R}^{m\times k}$. The outputs are returned in the form of a $n \times m$ matrix.

The routine `multiprod.m`, implementing "inner" product between tensors of any order, is used to increase speed, in place of two nested `for` loops over the rows of each input matrix. Downside: code interpretation not as straightforward as with `for` loops.

```
% INPUTS:
% X: nxk  contains n vectors of length k
% Y: mxk  contains m vectors of length k
% d:      vector of lenght k, containing the correlation lengths
% nu:     positive number, nugget term
% fun: one of the following strings, to specify correlation function:
```

```
%       'exp2'     (square exponential);    'abs_exp'  (absolute exponential);
%       'matern32' (matern 3/2);            'matern52' (matern 5/2).
%
% OUTPUT:
% C: nxm matrix, with C(i,j) = corr(X(i,:), Y(j,:))

function C = Corr_fun(X, Y, d, nu, fun)

n = size(X,1);
m = size(Y,1);
k = size(X,2);

%% STORE DIFFERENCES BETWEEN ALL PAIRS OF ROWS OF 'X' AND 'Y' (in 4D tensor)
X2 = repmat(X, [1 1 1 m]);          % size: n x k x 1 x m,   m copies of X
Y2 = repmat(Y, [1 1 1 n]);          % size: m x k x 1 x n,   n copies of Y
Z2 = X2 - permute(Y2, [4 2 3 1]); % n x k x 1 x m.
                                    % Z2(i,:,:,j) = X(i,:) and Y(j,:)
Z = permute(abs(Z2), [3 2 1 4]);  % 1 x k x n x m. Just reorder dimensions for
                                    %                following multiprods to work.

%% ESSENTIALLY, COMPUTE  Z*D*Z, 'D' DIAGONAL WITH CORR LENGTHS. Via two multiprods:
D=sparse(diag(d.^-2)); % k x k
if k==1
    D = d^(-2); % special case needs to be explicit for following multiprods to work
end
A = multiprod(Z, D); % 1 x k x n x m
A = multiprod(A, permute(Z, [2 1 3 4])); % equivalent to many z*D*z', z=X(i,:)-Y(j,:)
A = squeeze(A);  % convert from 1x1xnxm to nxm
if n==1 % if n=1, specify that n x m becomes 1 x m, not m x 1 (as by default)
    A = reshape(A, 1, m);
end

%% PERFORM FINAL COMPUTATION, ACCORDING TO SPECIFIED CORRELATION FUNCTION
if strcmp(fun, 'exp2') % Square exponential
    C = exp( -A ) + nu*(A==0);

elseif strcmp(fun, 'abs_exp') % Absolute exponential
    C = exp(-sqrt(A)) + nu*(A==0);

elseif strcmp(fun, 'matern32') % Matern 3/2
    A = sqrt(3*A);
    C = (1+A).*exp(-A) + nu*(A==0);

elseif strcmp(fun, 'matern52') % Matern 5/2
    A1 = sqrt(5*A);
    A2 = 5*A/3;
    C = (1 + A1 + A2).*exp(-A1) + nu*(A==0);
end

end
```

D.2. Principal Components

By default, this function computes the PCs of a set of m vectors in $\mathbb{R}^s$ (each provided in the form of a $N_1 \times N_2$ matrix, where $s = N_1 N_2$), and the coefficients of each vector with respect to the PCs. If `varargin` is present, then it should be a vector of s positive weights; in this case, the procedure presented in Subsection 3.3.3 is implemented. This is the way the function is used in Section E and Section F of this appendix.

```
% INPUTS
% - X: N1 x N2 x m matrix consisting of m starting matrices.
% - varargin: if present, a vector of weights of length s=N1*N2.

% OUTPUS
% - PC: N1 x N2 x (m-1) matrix with j-th PC at level PC(:,:,j).
% - Mn: N1 x N2 matrix, average of initial matrices in X.
% - Coeff: mx(m-1) matrix. Coeff(i,:) = coefficients of X(:,:,i) wrt the m-1 PCs.
% - Std: (m-1)x1 vector, with PC st.deviation from eigenvalue decomposition.

function [PC, Mn, Coeff, Std] = PCA(X, varargin)

%% GENERAL VARIABLES
N1 = size(X,1); N2 = size(X,2); s = N1*N2;
m = size(X,3);
Mn = mean(X,3); % N1 x N2, average of X(:,:,1), ..., X(:,:,m)
Xbar = X - Mn; %  N1 x N2 x m
Xbar = reshape(Xbar, [N1*N2, n])'; % m x s
Xbar(isnan(Xbar)) = 0; % replace NANs by 0s

%% PCA: TWO CASES, ACCORDING TO WHETHER WEIGHTS ARE PROVIDED OR NOT

if isempty(varargin)  % no weights ==> do standard PCA
    [U,S,PC] = svd(Xbar, 'econ'); % Xbar = U*S*PC'; U:m x m; S:m x m; PC:s x m;
    % Get rid of information corresponding to last PC:
    % U: mx(m-1); S: (m-1)x(m-1); PC: sx(m-1); still Xbar = U*S*PC'.
    U(:,end) = []; PC(:,end) = []; S(:,end)=[]; S(end,:)=[];

    Std = diag(S)/sqrt(m-1);
    Coeff = U*S;  % m x (m-1)

else      % weights has been provided; solve Xbar'*Xbar*W u = \lambda u
    weights=varargin{:};
  % Transform forward
    W = sparse(1:s, 1:s, weights, s, s); % sxs diagonal matrix of weights
    Y = Xbar * sqrt(W);
  % Compute SVD
    [U,S,V] = svd(Y, 'econ');
```

```
    U(:,end)=[]; V(:,end)=[]; S(:,end)=[]; S(end,:)=[];
  %Transform back
    W_minus1 = sparse(1:s, 1:s, 1./weights, s, s);
    PC = sqrt(W_minus1) * V;

    Coeff = U*S;
    Std = diag(S)/sqrt(m-1);
end

PC = reshape(PC, [N1, N2, m-1]);

end
```

E. Code Relating to Chapter 4

The following code has been specifically written and employed to tackle the problem of ocean temperature reconstruction described in Chapter 4.

E.1. Emulation of PC scores (functions $f_j(\cdot)$ of Section 4.6)

For each $j = 1, \ldots, n-1$, the n known values $\{f_j(\boldsymbol{x_i})\}_{i=1,\ldots,n}$ corresponding to the design points $\boldsymbol{x_i} \in \mathbb{R}^3$ are provided in the j^{th} column of the input matrix `Coeff`. Emulator predictions are made for the input parameters provided in `New_points`. The regressors to use are specified in `index_lr` (they are selected as explained in Subsection 4.7.1).

Correlation lengths and nugget can be provided; otherwise, they are estimated as explained in Section 4.8. In such a case, the function in Subsection E.2 of this appendix is used. As output, alongside the mean, either only the emulator variances or all pairs of emulator covariances are computed, according to what specified in string `var_cov`.

```
% INPUTS:
% - Design_points: nx3 matrix: in each row, triple the form:
%                  x=(ecc*cos(prec), ecc*sin(prec), obliq).
% - index_lr: rx3 matrix of integers between 1 and 5, corresponding to the five
%             regressors in eqn (4.11). index_lr(j,:) contains the indices of the
%             three regressors selected as basis for emulator mean of PC j.
% - Coeff: nxh matrix, with coefficients to emulate in columns. r≤h≤n-1.
```

```
% - r: number of (first) columns of Coeff to actually emulate.
% - New_points: Tx3 matrix: in each row, new parameters at which to perform emulation
% - cor_fun: one of the strings 'exp2', 'matern32', 'matern52', 'abs_exp'.
% - var_cov: a string, either 'var' or 'cov'.
% - varargin: optional. rx3 matrix, with corr lengths in columns 1&2, nugget in col3.
%
% OUTPUTS:
% - M: Txr matrix, M(:,j) = emulated means of jth PC, for inputs in New_points.
% - VarCov: size and content depends on 'var_cov' input. See below.
% - Dnu: rx3 matrix. Final (provided or estimated) corr lengths and nugget.
% - sigma_sq: rx1 vector, with s2 values of emulation. Needed if a nugget term is
%             present, but want to plot continuous trajectories. In this case, nu*s2
%             shall be subracted from the emulator variance at each point.
%
%   If var_cov='var', then:
% - VarCov: Txr matrix. As M, but with variances rather than means.
%   If var_cov='cov', then:
% - VarCov: TxTxr. Cov(:,:,k) = covar matrix of emulated kth PC.


function [M, VarCov, Dnu, sigma_sq] = emulation_PCscores(Design_points, index_lr,
                                    Coeff, r, New_points, cor_fun, var_cov, varargin)

n = size(Design_points,1);     % number of design  points
q = size(Design_points,2) +1; % q-1 regressors will be used
T = size(New_points,1);        % number of test points

%% PART 1: CHOOSE REGRESSORS TO USE, AND VALUES OF CORRELATION LENGTHS & NUGGET
%%          (IF NOT ALREADY SPECIFIED IN OPTIONAL ARGUMENT 'varargin')

Dnu = zeros(r,3);
H_full = cell(r,1); % H_full{c} nx3 matrix, will contain regressors for component c,
                    % at design points
h_full = cell(r,1); % h_full{c} Tx3 matrix, will contain regressors for component c,
                    % at test points
All_discr_regressors = [Design_points, Design_points(:,1:2).^2]; % eqn (4.11)
All_cont_regressors =  [New_points, New_points(:,1:2).^2];       % eqn (4.11)

for c = 1:r
    y = Coeff(:,c); % output values at design points

    %% Build matrices of regressors, for both design points and new parameters
    H = [ones(n,1), All_discr_regressors(:, index_lr(c,:))];
    H_full{c} = H;  % nxq: Matrix of covariates at design points, for component c
    h = [ones(T,1), All_cont_regressors(:, index_lr(c,:))];
    h_full{c} = h;  % Txq: Matrix of covariates at new parameters, for component c

    %%  Choose correlation lengths
    if isempty(varargin)  % ie, if no corr_lengths have been specified
        m1 = 0.02; m2 = 0.003; m_nu = 0.5; a = 4; % parameters for prior
```

```
        % next function finds MAP estimate (code provided later)
        [d, nu] = max_cross_val(Design_points, y, H, cor_fun, m1, m2, m_nu, a);
        Dnu(c,1:2) = d; Dnu(c,3)=nu;
    else  % ie, corr lengths and nugget were given in input, in varargin
        Dnu = varargin{:};
    end
end

%% PART 2: CARRY OUT ACTUAL EMULATION

% Initialise relevant variables to zeros
M = zeros(T,r);
if strcmp(var_cov, 'var')
    VarCov = zeros(T,r);
else
    VarCov = zeros(T,T,r);
end
sigma_sq = zeros(r,1);

for c = 1:r % perform emulation with 'Design_par' and 'Coeff(:,c)'

    d = [Dnu(c,1), Dnu(c,1), Dnu(c,2)]; % same corr length for e*cos and e*sin
    nu = Dnu(c,3);
    y = Coeff(:,c);
    A = Corr_fun(Design_points, Design_points, d, nu, cor_fun);  % nxn
    H = H_full{c};  h = h_full{c};
    K = H'/A;                 % qxn, K = H'*(A^-1)
    B = K*H;                  % qxq, B = H'*(A^-1)*H
    b = B\(K*y);              % qx1, b = (B^-1)*Ky
    f = y - H*b;              % nx1
    e = A\f;                  % nx1, e = (A^-1)*(y - Hb)
    s2 = (f'*e)/(n-q-2);      % scalar, posterior average of sigma^2
    sigma_sq(c) = s2;         % store for output

    t = Corr_fun(New_points, Design_points, d, 0, cor_fun);  % Txn
    M(:,c) = (h*b) + (t*e);
    p = h' - K*t';            % qxT, p = h(x) - H'*(A^-1)*t

    if strcmp(var_cov, 'var') % only return variances
        v1 = Corr_fun(Design_points(1,:), Design_points(1,:), d, nu, cor_fun);
        VarCov(:,c) = s2*(v1 - diag(t*(A\t')) + diag(p'*(B\p)) );
    else                      % otherwise return full covariances
        v1 = Corr_fun(New_points, New_points, d, nu, cor_fun);
        VarCov(:,:,c) = s2*(v1 - t*(A\t') + p'*(B\p) );
    end

end % end of (for c=1:r)
end
```

E.2. Maximum a Posteriori Estimate of d and ν

This function computes the correlation lengths and nugget, as explained in Section 4.8 (*i.e.*, by maximising (4.19)). Design points are in `Design_points`, response values in `y`, predictors in `H`. The function `cross_val.m` (not shown for brevity) is maximised, after this is component-wise multiplied with a product of Gamma densities whose parameters are specified as input. Maximisation carried out from M different starting points.

```
% INPUTS
% - Design_points: nx3 matrix with n design points (orbital parameters).
% - y:   nx1 vector of observed outputs.
% - H:   nxq matrix of predictors (usually, first column of 1s).
% - cor_fun: one of the following strings: 'exp2', 'matern32', 'matern52', 'abs_exp'.
% - m1, m2, m_nu: position of the modes of the gamma distributions used as prior
%                 in the maximisation (respectively for d1, d2, nu).
% - a: shape parameter of the Gamma prior. Smaller a <-> flatter density.
%
% OUTPUTS:
% - d:  final value of optimised correlation lengths (2D)
% - nu: final value of optimised nu (1D)

function [d, nu] = max_cross_val(Design_points, y, H, cor_fun, m1, m2, m_nu, a)

%% STORE IN X (size: Mx3) INITIAL STARTING POINTS FOR FUTURE MAXIMISATION
M = 10;
p = haltonset(3,'Skip',floor(1000*rand),'Leap',16);
X = net(p,M); % well-scattered points, in 3D unit cube
X = X*diag(2*[m1,m2,m_nu]); % rescales the columns within a plausible range

%% DEFINE THE FUNCTION h=f*g TO BE MAXIMISED
% f: Likelihood, approximated through cross validation.
% g: Prior (gamma, with shape parameter a>1 and mode equal to m1, m2, or m_nu).
% h: Main function to be maximised, product of likelihood and prior.
f = @(x) cross_val(exp(x(1:2)), exp(x(3)), Design_points, y, H, cor_fun, 'dens');
g = @(x,alpha,m) gampdf(exp(x), alpha, m/(alpha-1));
h = @(x) -sum(log(f(x))) - log( g(x(1),a,m1) * g(x(2),a,m2) * g(x(3),a,m_nu) );

%% CARRY OUT MAXIMISATION STARTING FROM M POINTS IN X
x0=log(X(1,:));
[xf,vf] = fminsearch(h, x0, options); % xf = maximising value, vf=h(xf)

for k = 2:M
    x0 = log(X(k,:));
    [x_temp,v_temp] = fminsearch(h, x0, options);
```

```
    if v_temp < vf
        xf=x_temp; vf=v_temp;
    end
end

%% RETURN MAXIMISING VALUES
d = exp(xf(1:2));    nu = exp(xf(3));

end
```

E.3. Recombine the PC scores

Given the output of `emulation_PCscores.m`, this function computes temperature emulated mean and variance/covariance for a set of locations, via linear combinations (see (4.29) and (4.31)). Values of the PCs at the N_{loc} locations are provided in `PC_Val`. The average of the original simulations at these locations is provided in `Mn_Val`. Both of these are computed via the procedure of Subsection 4.9.1, for which the main routine is provided in the following Subsection E.4.

```
% INPUTS (Starred inputs generally obtained as output of 'emulation_PCscores.m')
% - *M_Pc:   Txn matrix (n≥r). In jth column, emulated mean of jth PC coefficient.
% - *Cov_Pc: TxTxr tensor of r covar matrices, or Txr matrix of r variance vectors,
%            as from 'emulation_PCscores' output. Last dimens can also be >r,
%            it will be trimmed later.
% - PC_Val:  Nloc x n. In row i, values of the PCs at ith location.
% - Mn_Val:  Nloc x 1. Intercept of affine combination (for each location).
% - Std_PCA: Vector, length ≥r. St.Dev. of PCs, from eigenvalue decomposition.
% - r:       Number of PCs to use in affine combinations.
% - var_cov: a string, either 'var' or 'cov'. Accordingly, only variance or full
%            covariance will be computed as output Cov.
%
% OUTPUTS:
% - M: T x Nloc. M(i,j) = emulated mean temperature at time i, location j.
%    Second output changes according to value of input 'var_cov'.
%    If var_cov = 'var', then:
% - VarCov: T x Nloc. As M, but with variances rather than means.
%    If var_cov = 'cov', then:
% - VarCov: T x T x Nloc, with full covariances at levels (:,:,j).

function [M,VarCov]= emul_complete(M_Pc, Cov_Pc, PC_Val, Mn_Val, Std_PCA, r, var_cov)

%% GENERAL VARIABLES, RESHAPING AND "TRIMMING"
n = length(Std_PCA);   % total number of PCs
```

```
T = size(M_Pc,1);       % number of inputs at which compute emulator predictions
Nloc = size(PC_Val,1); % total number of locations
Std_PCA = reshape(Std_PCA, [1,n]);
M_Pc = M_Pc(:, 1:r);   % Txr, discard values of unused PCs


%% PART 1: ADD CONSTANT VARIANCE, OR IDENTITY MATRIX COVARIANCE, FOR UNUSED PCs
if strcmp(var_cov, 'var')
    Full_PC_Var = zeros(T,n);  % Txn
    Full_PC_Var(:, 1:r) = Cov_Pc(:, 1:r);  % Actual variance for first r PCs
    if r<n
        Full_PC_Var(:,r+1:n) = ones(T,1)*(Std_PCA(r+1:n).^2); % constant variance
    end
else
    Full_PC_Cov = zeros(T,T,n);  % TxTxn
    Full_PC_Cov(:, :, 1:r) = Cov_Pc(:,:, 1:r); % Actual covariance for first r PCs
    if r<n       % multiple of id. matrix for remaining components
        for k = r+1:n
            Full_PC_Cov(:, :, k) = (Std_PCA(k)^2)* sparse(eye(T));
        end
    end
end % end of if/else statement


%% PART 2: COMPUTE EMULATOR MEAN AND (CO)VARIANCES, AS LINEAR COMBINATIONS
M = ones(T,1)*Mn_Val' + (M_Pc*PC_Val(:, 1:r)'); % TxNloc + (Txr)x(rxNloc)
Squared_PC = (PC_Val').^2;      % n x Nloc
if strcmp(var_cov, 'var')
    VarCov = Full_PC_Var*Squared_PC; % T x Nloc, from (Txn)x(nxNloc)
else
    VarCov = zeros(T,T,Nloc);
    for loc=1:Nloc
        %% Take linear combination of the n levels of Full_PC_Cov, with ...
            coefficients Squared_PC(:,loc).
        for c=1:n
            VarCov(:,:,loc) = VarCov(:,:,loc) + Squared_PC(c,loc)*Full_PC_Cov(:,:,c);
        end
end  % end if/else

% NOTE: double for loop can be avoided, by running the following multiprod: ...
    TxTxNloc, from (TxTxn)x(nxNloc):
% VarCov = squeeze(multiprod(Full_PC_Cov, Squared_PC, [0 3], [1 0])).
% This however requires to store a matrix of dimension T x T x n x Nloc.

end
```

E.4. Interpolate Among Cells (Subsection 4.9.1)

The following function returns the indices of the vectors `lat_vec` and `lon_vec` whose values are closest to `lat_star` and `lon_star`, respectively. It also returns weights, inversely proportional to the geodesic distance between the point with coordinates (`lat_star`, `lon_star`) and the points with coordinates identified in `lat_vec` and `lon_vec`. Compare with (4.26). If requested, only indices corresponding to sea (or land) will be returned.

```
% INPUTS
% - lat_star: number in [-90, 90]
% - lon_star: any real number
% - lat_vec: row or col vector of latitudes in decreasing order (length Nlat)
% - lon_vec: row or col vector of longitudes in increasing order (length Nlon)
% - varargin: optional argument. If provided, it must consist of:
%             1) a land-sea mask M: Nlat x Nlon (0 sea, 1 land)
%             2) a string, either 'sea' or 'land'.
%             In this case, only cells corresponding to sea or land will be returned.
%
% OUTPUTS
% - ind_lat: 4x1 vector, with latitude indices of closest gridcells
% - ind_lon: 4x1 vector, with longitude indices of closest gridcells
% - linear_ind: 4x1 linear indices between 1 and Nlat*Nlon, each corresponding to ...
    the pair (ind_lat(i), ind_lon(i))
% - w: 4x1 vector with weights, summing up to 1.
%
% Note: Function to compute geodesic distance in next box.

function [ind_lat, ind_lon, linear_ind, w] = interpolate_latlon(lat_star, lon_star,
                                                    lat_vec, lon_vec, varargin)
%% PRELIMINARY CODE
Nlon = length(lon_vec);
Nlat = length(lat_vec);
lat_vec = reshape(lat_vec, [Nlat,1]);
lon_vec = reshape(lon_vec, [Nlon,1]);

% Bring lon_star in [-180,180]
lon_star = mod(lon_star, 360);
if lon_star ≥ 180
    lon_star = lon_star-360;
end

% If further inputs are provided, store them
if ¬isempty(varargin)
    M = varargin{1};
```

```
    sea_land_str = varargin{2};
end

%% IDENTIFY CONSECUTIVE INDICES ind_lat1 & ind_lat2, WHOSE ELEMENTS IN lat_vec
%  ARE RESPECTIVELY BIGGER AND SMALLER THAN lat_star
ind_lat1 = sum(lat_vec-lat_star≥0); % index in lat_vec whose latitude is closest to
                                    % (and bigger than) lat_star
if lat_vec(ind_lat1)==lat_star
    ind_lat2 = ind_lat1;        % take same index ...
else
    ind_lat2 = min(ind_lat1+1, Nlat); % ... or following one (if not last already)
end
ind_lat=[ind_lat1, ind_lat1, ind_lat2, ind_lat2]';

%% SAME AS ABOVE, FOR LONGITUDES
ind_lon1 = sum(lon_vec-lon_star≤0); % lon_vec[ind_lon1] ≤ lon_star
if lon_vec(ind_lon1)==lon_star
    ind_lon2 = ind_lon1;
else
    ind_lon2 = min(ind_lon1+1, Nlon);
end
ind_lon=[ind_lon1, ind_lon2, ind_lon1, ind_lon2]';

linear_ind=zeros(4,1); % linear indices corresponding to selected pairs
for i=1:4
    linear_ind(i) = sub2ind([Nlat,Nlon], ind_lat(i), ind_lon(i));
end

%% IF 'varargin' PROVIDED, SELECT ONLY INDICES CORRESPONDING TO SEA OR LAND
if ¬isempty(varargin)
    if strcmp(sea_land_str, 'sea')
        sea = M(linear_ind)<0.5; % M = mask stored at beginning
        linear_ind = linear_ind(sea);
    else
        land = M(linear_ind)>0.5;
        linear_ind = linear_ind(land);
    end
end

%% COMPUTE WEIGHTS, INVERSELY PROPORTIONAL TO GEODESIC DISTANCE
[ind_lat, ind_lon] = ind2sub([Nlat,Nlon], linear_ind');
A = [lat_vec(ind_lat), lon_vec(ind_lon)];
B = [lat_star, lon_star];
w = geodesic_dist(A, B) + 1.e-10; % correction ensures stricly positive values
w = 1./w;
w = w/sum(w(:));   % normalise sum to 1

end
```

The code of the function `geodesic_dist.m`, used above, is shown in the next box.

E.4.1 Compute Great Circle Distance (from equation (4.26))

```
% INPUTS
% - Coord1: nx2. Each row, one pair of the form (lat,lon), in degrees.
% - Coord2: px2. Each row, one pair of the form (lat,lon), in degrees.
%
% OUTPUTS
% - G: nxp. G(i,j) = geodesic dist between Coord1(i,:) and Coord2(j,:).

function G = geodesic_dist(Coord1, Coord2)

Coord1 = Coord1/180*pi;
Coord2 = Coord2/180*pi;
n = size(Coord1,1);
p = size(Coord2,1);

G = zeros(n,p);
for i = 1:n
    lat1 = Coord1(i,1);
    lon1 = Coord1(i,2);
    for j = 1:p
        lat2 = Coord2(j,1);
        lon2 = Coord2(j,2);
        G(i,j) = acos(sin(lat1)*sin(lat2) + cos(lat1)*cos(lat2)*cos(lon2-lon1));
    end
end
end
```

F. Code Relating to Chapter 5

The following code has been specifically written and employed to tackle the problem of the last Interglacial Greenland ice sheet reconstruction described in Chapter 5.

F.1. Regridding of Original Morphologies (Subsection 5.4.1)

This function regrids an image (a matrix H), usually around Greenland, from any grid into a regular latitude-longitude grid. Regridding happens in the coordinates obtained by projecting elements of the sphere onto the plane tangent to the sphere at the point (72°N, 40°W). See the coordinate transformation in (5.8).

```
% INPUTS
% - H: matrix, containing starting values, among which to interpolate.
% - lat_or & lon_or: *matrices*, containing the original lats and lons coordinates
%                    at which values in H are provided.
% - lat & lon: *vectors* (grid will be created as 'cartesian product') of lat and lon
%              coordinates where interpolating values need computing.
% - str: one of the strings 'linear', 'nearest', 'natural', 'cubic', specifying the
%        interpolation method to use.
%
% OUTPUTS
% - A: new regridded image, with size length(lat) x length(lon)

function A = regrid(lat_or, lon_or, H, lat, lon, str)

latstar = 72*pi/180;      % [point of tangency, ...
lonstar = 320*pi/180;     % ... in radians]

%% DEFINE TANGENT VECTORS AT (latstar,lonstar). These are *column* vectors.
V1 = [-cos(latstar)*sin(lonstar); ... % derivative wrt lon
       cos(latstar)*cos(lonstar); ...
       0]/abs(cos(latstar));
V2 = [-sin(latstar)*cos(lonstar); ... % derivative wrt lat
      -sin(latstar)*sin(lonstar); ...
       cos(latstar)];

%% TRANSFORM MATRICES 'lat_or' & 'lon_or' IN COLUMN VECTORS
n1 = size(lat_or,1);  n2 = size(lat_or,2);  N = n1*n2;
lat_or = reshape(lat_or*pi/180, [N,1]);
lon_or = reshape(lon_or*pi/180, [N,1]);

%% FIRST CREATE RECTANG GRID FROM 'lat' & 'lon', THEN TRANSFORM IN COLUMNS (as above)
```

```
[lat,lon] = ndgrid(lat,lon); % lat and lon now both matrices
n1 = size(lat,1); n2 = size(lon,2); N = n1*n2;
lat = reshape(lat*pi/180, [N,1]);
lon = reshape(lon*pi/180, [N,1]);

%% PROJECT ORIGINAL GRID ('lat_or' & 'lon_or') ONTO TANGENT PLANE. TWO STEPS:
% 1) Obtain points on the sphere, via spherical polar coordinates:
latlon_or_grid = [cos(lat_or).*cos(lon_or), cos(lat_or).*sin(lon_or), sin(lat_or)];
% each row of 'latlon_or_grid' is unit 3D vector
% 2) Project each row onto tangent vectors (scalar product):
x = reshape(latlon_grid*V1, [n1,n2]); % (Nx3)x(3x1) reshaped into n1xn2
y = reshape(latlon_grid*V2, [n1,n2]);

%% PROJECT NEW GRID ('lat' & 'lon') ONTO TANGENT PLANE (as above)
latlon_grid=[cos(lat).*cos(lon), cos(lat).*sin(lon), sin(lat)];
X = reshape(latlon_grid*V1, [n1,n2]);
Y = reshape(latlon_grid*V2, [n1,n2]);

%% CARRY OUT INTERPOLATION, VIA BUILT-IN FUNCTION 'griddata'
A = griddata(x, y, H, X, Y, str); % size: n1 x n2 = length(lat) x length(lon)

end
```

F.2. Generate Masks (Subsection 5.4.3)

Given a morphology in input, this function computes the first approximation of the corresponding mask (steps 1–3 in Subsection 5.4.3). The mask is subsequently smoothed through the function `denoising_mask_radius.m`, which I report later.

```
% INPUT
% Morph: Nlat x Nlon matrix of heights. (in the following, call p the number of
%                                        'non-sea' cells)
% OUTPUT:
% Mask: Nlat x Nlon matrix. Ice-land-sea mask: 0=ice, 1=land, 2=sea.

function Mask = ice_mask_generator(Morph)

%% 'Ht' CONTAINS THE SURFACE HEIGHTS OF THE ORIGINAL 14 MORPHOLOGIES
Ht = dlmread('Data/Dataset.txt', '');
Ht = Ht(2:end,:); % remove first row of area weights; size(Ht) = 14 x p

%% 'Ice_Mask' CONTAINS THE MASKS (sea excluded) OF THE ORIGINAL MORPHOLOGIES (14xp)
Ice_Mask = dlmread('Data/IceMask.txt');
land = Ice_Mask>0.5;  % logical index
ice  = Ice_Mask<0.5;  % logical index
```

```
%% FOR ALL LOCATIONS, COMPUTE THE THRESHOLD c ABOVE WHICH A GIVEN HEIGHT
%  IS TO BE ASSOCIATED WITH ICE
X = Ht; X(land) = Inf;
a = min(X); % 1xp; min(height | there is ice)
X = Ht; X(ice) = -Inf;
b = max(X); % 1xp; max(height | there is no ice)
c = (a+b)/2; % 1xp

%% BUILD THE MASK
str = 'Data/Mask.nc';
Mask = ncread(str, 'Mask'); % Land-sea mask for Greenland: 1=land, 2=sea.
GL = Mask<1.2; % logical index, identifies GreenLand, excludes sea
Morph = Morph(GL); % turn 'Morph' into a col vector, only with Greenland (land) cells
Morph = reshape(Morph, [1,numel(Morph)]); % reshape to row vector, as vector 'c'
ice_land_mask = double((Morph-c)<0); % assign 1 to land, 0 to ice
Mask(GL) = ice_land_mask;

end
```

F.2.1 Function `denoising_mask_radius.m`

This function takes an ice-land-sea mask and smoothens the ice and land parts. A disk of radius r km around each cell is considered, and its mask value is replaced by ice (land) according to whether the percentage of ice (land) in the disk is greater than p_1 (p_2). Note: The original code is structured so that variables of the function `denoising_mask_radius.m` are passed on to `ice_prop.m`. Here the two functions are shown in separate boxes for convenience, but variables in `denoising_mask_radius.m` are seen by `ice_prop.m` in the real code.

```
% INPUTS:
% - M:    Nlat x Nlon mask (matrix with 0=ice, 1=land, 2=sea).
% - lat:  vector of length Nlat, with lat coordinates for M (degrees).
% - lon:  vector of length Nlon, with lon coordinates for M (degrees).
% - p1:   number in [0,1]: percentage of ice below which an ice cell is replaced
%         by a land cell.
% - p2:   percentage of land below which a land cell is replaced by an ice cell.
% - r:    radius, measured in kilometers.
%
% OUTPUTS:
% - M_denoised: Nlat x Nlon denoised mask

% The values of 'lat' and 'lon' are used in the next function, 'ice_prop'.
```

```
function M_denoised = denoising_mask_radius(lat, lon, M, p1, p2, r)

%% STORE ICE AND LAND POSITIONS OF ORIGINAL MASK
orig_ice = find(M < 0.5);
orig_land = find( (M > 0.5) & (M < 1.5) );
M_denoised = M;

%% DENOISE, FIRST THROUGH ICE CELLS ...
for i = 1:length(orig_ice)
    ind = orig_ice(i);
    p_ice = ice_prop(M, ind, r); % approximates proportion of ice within r km
    if p_ice < p1
        M_denoised(ind) = 1; % put land
    end
end

%% ... THEN THROUGH LAND CELLS
for i = 1:length(orig_land)
    ind = orig_land(i);
    p_ice = ice_prop(M, ind, r);
    p_land = 1-p_ice;
    if p_land < p2
        M_denoised(ind) = 0; % put ice
    end
end

end
```

F.2.2 Function `ice_prop.m`

The function `ice_prop.m` computes the proportion of ice of the mask `M`, in (approximately) a circle of radius r around the location identified by the index `ind`. Other variables are passed from the previous script.

```
function ice_percentage = ice_prop(M, ind, r)

%% BASIC VARIABLES
Earth_rad = 6371; % km
lat = lat*pi/180;
lon = lon*pi/180;
Nlat = length(lat); Nlon = length(lon); % size(M) = Nlat x Nlon

%% COMPUTE STEPSIZE OF 'lat' AND 'lon' (IN RADIANS)
lat_angle = abs(lat(2)-lat(1));
lon_angle = abs(lon(2)-lon(1));
```

```
%% CREATE RECTANGULAR GRIDS FROM 'lat' AND 'lon' VECTORS
[latgrid, longrid] = ndgrid(lat, lon); % Two (Nlat x Nlon) matrices
Theta = latgrid(ind); % latitude of cell of interest
Phi   = longrid(ind); % longitude of cell of interest
[ihat,jhat] = ind2sub([Nlat, Nlon], ind); % converts linear index into subscripts

%% DEFINE THE TWO LATITUDE INDICES, BETWEEN WHICH ALL CELLS
%  AT MOST r KM APART FROM THE CELL AT LATITUDE 'Theta' LIE
max_lat_cells = ceil( r/(Earth_rad*lat_angle) ); % upper bound for number of cells,
                                                 % which cover less than r km in lat
i1 = max(ihat - max_lat_cells, 1);     % max and min needed to have ...
i2 = min(ihat + max_lat_cells, Nlat);  % ... indices between 1 and Nlat

%% DO THE SAME FOR LONGITUDE INDICES. NUMBER OF CELLS NOW DEPENDS ON LATITUDE
max_lon_cells = ceil( r/(Earth_rad*cos(Theta)*lon_angle) );
j1 = max(jhat - max_lon_cells, 1);
j2 = min(jhat + max_lon_cells, Nlon);

%% TRIM VARIABLES OF INTEREST, AROUND THE RELEVANT lats & lons FOUND ABOVE
New_Lat = latgrid(i1:i2, j1:j2); % n1 x n2
New_Lon = longrid(i1:i2, j1:j2); % n1 x n2
M_New   =       M(i1:i2, j1:j2); % n1 x n2

% NOW COMPUTE 'coslambda': COSINE OF GEODESIC ANGLE BETWEEN POINT (Theta, Phi)
% AND ALL OTHER POINTS IN 'New_Lat' and 'New_Lon' (say, nxm matrices)
coslambda = (sin(Theta)*sin(New_Lat)) + (cos(New_Lat).*cos(New_Lon-Phi)*cos(Theta));
dist = Earth_rad*acos(coslambda);  % n x m
neigh = M_New(dist<r); % only take mask cells, distant < than r km from (Theta, Phi)
L = length(neigh);
ice_percentage = sum(neigh<0.5)/L; % proportion of neighbours with ice

end
```

F.3. Identifying Plausible Morphologies (Section 5.7)

F.3.1 Compatibility With Ice-Core Records

For N input parameters $\boldsymbol{x} \in \mathbb{R}^8$, stored in `Input_par` $\in \mathbb{R}^{N\times 8}$, and for the locations corresponding to `index_loc`, the following function measures the compatibility between the emulators' predictions at the input parameters and the ice-core data at the locations. The condition $I(\boldsymbol{x}) < 2$ is implemented, $I(\cdot)$ defined as in (5.21). See Section 5.7 for more details, particularly condition 1 on page 191. The routine selecting morphologies which pass condition 2 is implemented in the next `ht_match.m`.

```
% INPUTS:
% - Design_par:  nx8 matrix of design points, to carry out emulation.
% - Sim_Outputs: nx6 matrix of simulator outputs for the 6 sites (list below).
% - cor_fun:     one of the strings 'exp2', 'matern32', 'matern52', 'abs_exp'.
% - d, nu:       6x1 vectors of corr lengths and nugget, respectively.
% - Input_par:   Nx8 matrix with N input parameters, at which assess compatibility.
% - range:   3x6 matrix. In each column, 'min, med, and max' d18O values for a site.
% - index_loc:   vector of integers of length L≤6, specifying at which locations
%                to carry out the comparison.
% - thrs: vector of length L, with thresholds used to classify an input parameter
%         compatible to the relevant record (thrs(i) used for location index_loc(i)).
%
% OUTPUTS
% - X: NxL. X(i,j) = compatibility measure (5.15) at input i, location index_loc(j)
% - index_compat: Nx1, logical. TRUE at position i iff input i compatible to records
%                 at all sites specified in index_loc.
%
% ORDER OF LOCATIONS
% 1: NEEM        2: NGRIP        3: GRIP
% 4: Camp        5: DYE3         6: GISP2

function [X, index_compat] = data_match(Design_par, Sim_Outputs, cor_fun, d, nu,
                                        Input_par, range, index_loc, thrs)
N = size(Input_par,1);
L = length(index_loc);

%% SELECT DATA ONLY FOR LOCATIONS SPECIFIED IN 'index_loc'.
range = range(:, index_loc); % 3xL
data_central = range(2,:);   % 1xL
std_top    = (range(3,:) - range(2,:))/sqrt(3); % compare to (5.20)
std_bottom = (range(2,:) - range(1,:))/sqrt(3); % compare to (5.20)
Sim_Outputs = Sim_Outputs(:, index_loc); % nxL
d = d(index_loc);   % vector of length L
nu = nu(index_loc); % vector of length L

% CARRY OUT EMULATION ON LOCATIONS OF INTEREST
M = zeros(N,L); % will store mean
S = zeros(N,L); % will store standard deviation
for loc = 1:L
    y = Sim_Outputs(:,loc);
    [M(:,loc), S(:,loc)] = emul(Design_par, y, Input_par, cor_fun, d(loc), nu(loc));
    % custom function, not shown here. Performs emulation in blocks of
    % at most 10,000 inputs, to avoid memory problems.
end

%% COMPUTE IMPLAUSIBILITY MEASURE (eqn (5.19) )
X = M - (ones(N,1)*data_central); % difference between emulator and records

% Create a matrix of 'record st.deviations', with std_top or std_bottom according to
% whether emulator prediction is > or < than data value)
```

```
Std_data = zeros(N,L);
for loc = 1:L
    Std_data(:,loc) = std_bottom(loc);
    higher_predictions = X(:,loc)>0;
    Std_data(higher_predictions, loc) = std_top(loc);
end
% Normalise difference in X by total standard deviation (emul + record)
Var_tot = S.^2 + Std_data.^2;
for loc = 1:L
   X(:,loc) = X(:,loc)./sqrt(Var_tot(:,loc));
end

%% DETECT INPUTS THAT HAVE IMPLAUS MEASURE AT ALL LOCATIONS < THAN THRESHOLD
index_compat = true(N,1);
for loc = 1:L
    index_compat = index_compat & (abs(X(:,loc)) < thrs(loc));
end

end
```

F.3.2 Physical Criterion of Plausibility

The following function implements condition 2 on page 191. It takes in input N parameters $\boldsymbol{x} \in \mathbb{R}^8$, and returns a logical vector of length N with `1`s corresponding to parameters satisfying the condition. Recall that this is satisfied if at least a fraction p of its cells, weighted by their surface area, have height within $\texttt{mean} \pm \texttt{n\_std} \times \texttt{std}$ of the 14 heigths of the original morphologies at that location (in practice, in Section 5.7 we choose $\texttt{n\_std} = 2$).

The code is structured to not suffer memory problems related to very large N.

```
% INPUTS:
% - Input_par: Nx8 matrix of input parameters.
% - H: Nlat x Nlon x n. H(:,:,i) contains the ith original morphology  (n=14).
% - n_sd: positive number. How many STDs from the mean of the H(:,:,i)s a given
%                          morphology is allowed to be, to be considered 'physical'.
% - p: percentage in [0,1] of 'good cells', above which to accept a morphology.
% - varargin: optional. If present, a logical vector of length N, specifying
%                       on which rows of Input_par to perform 'ht_match'.
%                       The output index will have zeros where 'varargin' had zeros.
%
% OUTPUT:
% - ht_index: a logical vector of length N with 1s where a morphology is 'physical'.
```

```
function ht_index = ht_match(Input_par, n_sd, p, varargin)

%% STARTING VARIABLES
N = size(Input_par, 1);
str = 'Data/Mask.nc';
lat = ncread(str, 'lat'); Nlat = length(lat);
lon = ncread(str, 'lon'); Nlon = length(lon);
mask = ncread(str, 'Physical_Mask'); % Nlat x Nlon
wei = ncread(str, 'Weights'); % px1, where p=Nlat*Nlon

% INITIALISE ht_index, PUTTING ZEROS IF SOME INPUTS NEED NOT BE CONSIDERED
if isempty(varargin)
    ht_index = true(N,1);
else
    ht_index = varargin{:};
end
Tot = sum(ht_index); % total number of shapes that have to be examined

% COMPUTE THE MORPHOLOGY OF MIN AND MAX ALLOWED HEIGHTS AT EACH LOCATIONS
Min_H = mean(H,3) - n_sd *std(H, [], 3); % mean and std along the 3rd dimension
Max_H = mean(H,3) + n_sd *std(H, [], 3);

% FFURTHER VARIABLES, NEEDED TO BUILT MORPHOLOGIES FROM 8D INPUT PARAMETERS
[PC, Mn, ~] = pca_greenland(); % extracts PCs and average of original morphologies

wei(mask>1.5) = 0;   % assign zero weights to the sea
wei = wei/sum(sum(wei)); % sum of weigths inside Greenland equals 1

% CODE IS DIVIDED INTO FOR LOOPS, TO AVOID MEMORY PROBLEMS  WHEN BUILDING SHAPES
N_block = 3000; % number of input parameters examined in each iteration
N_loop  = ceil(Tot/N_block); % length of for loop
important_indices = find(ht_index>0.5); % only indices where to assess physicality

for i=1:N_loop
    ind1 = (i-1)*N_block + 1;
    ind2 = min(i*N_block, Tot);
    small_block = important_indices(ind1:ind2); % N_block inputs to be examined
   % Next function builds the morphologies from the input parameters (not shown here)
    Shapes = build_shapes(PC, Mn, Input_par(small_block,:)); % Nlat x Nlon x N_block

    % Compare each morphology (shape) to min_H and max_H.
    I = (Shapes>Min_H) & (Shapes<Max_H); % Nlat x Nlon x N_block logical vector.
    I = I.*wei;                     % rescale each cell with corresponding weight.
    perc = (squeeze(sum(sum(I))));  % vector of length N_block: in component j,
                          % percentage of jth Shape that is between Min_H and Max_H.
    ht_index(small_block) = perc>p;
    clear Shapes;
end

end
```

Bibliography

Abramowitz, M. and Stegun, I. A. (1970). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, New York, ninth edition.

Adler, R. J. (1981). *The Geometry of Random Fields*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

Adler, R. J. and Taylor, J. E. (2007). *Random Fields and Geometry*. Springer Monographs in Mathematics. Springer.

Alden, K., Read, M., Timmis, J., Andrews, P. S., Veiga-Fernandes, H., and Coles, M. (2013). Spartan: a comprehensive tool for understanding uncertainty in simulations of biological systems. *PLoS computational biology*, 9(2):e1002916.

Andrianakis, I. and Challenor, P. G. (2012). The Effect of the Nugget on Gaussian Process Emulators of Computer Models. *Computational Statistics and Data Analytics*, pages 4215–4228.

Andrianakis, I., Vernon, I., McCreesh, N., McKinley, T., Oakley, J., Nsubuga, R., Goldstein, M., and White, R. (2017). History matching of a complex epidemiological model of human immunodeficiency virus transmission by using variance emulation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(4):717–740.

Araya-Melo, P. A., Crucifix, M., and Bounceur, N. (2015). Global sensitivity analysis of the Indian monsoon during the Pleistocene. *Climate of the Past*, 11(1):45–61.

Bastos, L. S. and O'Hagan, A. (2009). Diagnostic for Gaussian Process Emulators. *Technometrics*, 51:425–438.

Berger, A. and Loutre, M. (1991). Insolation values for the climate of the last 10 million years. *Quaternary Science Reviews*, 10(4):297–317.

Berger, J. O., de Oliveira, V., and Sanso, B. (2001). Bayesian Analysis of Spatially Correlated Data. *Journal of the American Statistical Association*, 26(456):1361–1374.

Bolstad, W. M. and Curran, J. M. (2016). *Introduction to Bayesian Statistics*. John Wiley & Sons, Third edition.

Bonceur, N., Crucifix, M., and Wilkinson, R. (2015). Global Sensitivity Analysis of the Climate-Vegetetion System to Astronomical Forcing: an Emulator-Based Approach. *Earth System Dynamics*, 6:205–224.

Born, A. and Nisancioglu, K. H. (2012). Melting of Northern Greenland during the last interglaciation. *The Cryosphere*, 6:1239–1250.

Brunnabend, S.-E., Schröter, J., Timmermann, R., Rietbroek, R., and Kusche, J. (2012). Modeled steric and mass-driven sea level change caused by Greenland Ice Sheet melting. *Journal of Geodynamics*, 59:219–225.

Calov, R., Robinson, A., Perrette, M., and Ganopolski, A. (2015). Simulating the Greenland ice sheet under present-day and palaeo constraints including a new discharge parameterization. *The Cryosphere*, 9(1):179–196.

CAPE Members, L. I. P. (2006). Last Interglacial Arctic warmth confirms polar amplification of climate change. *Quaternary Science Reviews*, 25(13-14):1383–1400.

Chandler, M., Dowsett, H., and Haywood, A. (2008). The PRISM Model/Data Cooperative: Mid-Pliocene data-model comparisons. *PAGES News*, 16(2):24–25.

Chang, K.-L. and Guillas, S. (2019). Computer model calibration with large non-stationary spatial outputs: application to the calibration of a climate model. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 68(1):51–78.

Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009). Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676.

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001). Bayesian Forecasting for Complex Systems Using Computer Simulators. *Journal of the American Statistical Association*, 96(454):717–729.

Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1997). Pressure matching for hydrocarbon reservoirs. In *Case studies in Bayesian statistics*, pages 37–93. Springer.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. John Wiley, New York. Revised Edition.

Cumming, J. A. and Goldstein, M. (2009). Small Sample Bayesian Designs for Complex High-Dimensional Models Based on Information Gained Using Fast Approximations. *Technometrics*, 51(4):377–388.

Cumming, J. A. and Goldstein, M. (2010). Bayes Linear Uncertainty Analysis for Oil Reservoirs Based on Multiscale Computer Experiments. *O'Hagan, West, AM (eds.) The Oxford Handbook of Applied Bayesian Analysis*, pages 241–270.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments. *Journal of the American Statistical Association*, 86(416):953–963.

Diggle, P. J. and Ribeiro, P. J. (2006). *Model-based Geostatistics*. Springer Series in Statistics. Springer.

Doll, J. C. and Jacquemin, S. J. (2018). Introduction to Bayesian Modeling and Inference for Fisheries Scientists. *Fisheries*, 43:152–161.

Dowsett, H., Dolan, A., Rowley, D., Moucha, R., Forte, A., Mitrovica, J., Pound, M., Salzmann, U., Robinson, M., Chandler, M., Foley, K., , and Haywood, A. (2016). The PRISM4 (mid-Piacenzian) paleoenvironmental reconstruction. *Climate of the Past*, 12:10.5194/cp–12–1519–2016.

Dowsett, H., Robinson, M., Haywood, A. M., Salzmann, U., Hill, D., Sohl, L., Chandler, M., Williams, M., Foley, K., and Stoll, D. (2010). The PRISM3D paleoenvironmental reconstruction. *Stratigraphy*, 7(2-3):123–139.

Dowsett, H., Thompson, R., Barron, J., Cronin, T., Fleming, F., Ishman, S., Poore, R., Willard, D., and Holtz, T. (1994). Joint investigations of the Middle Pliocene climate I: PRISM paleoenvironmental reconstructions. *Global and Planetary Change*, 9:169–195.

Dowsett, H. J., Barron, J. A., and Poore, R. Z. (1996). Middle Pliocene sea surface temperatures: a global reconstruction. *Marine Micropaleontology*, 27:13–25.

Dowsett, H. J., Foley, K. M., Stoll, D. K., Chandler, M. A., Sohl, L. E., Bentsen, M., Otto-Bliesner, B. L., Bragg, F. J., Chan, W.-L., Contoux, C., et al. (2013). Sea Surface Temperature of the mid-Piacenzian Ocean: a Data-Model Comparison. *Scientific reports*, 3.

Dowsett, H. J., Robinson, M. M., Foley, K. M., Herbert, T. D., Otto-Bliesner, B. L., and Spivey, W. (2019). The mid-Piacenzian of the North Atlantic Ocean. *Stratigraphy*, 16(3):119–144.

Fretwell, P., Pritchard, H., Vaughan, D., Bamber, J., Barrand, N., Bell, R., Bianchi, C., Bingham, R., Blankenship, D., Casassa, G., et al. (2013). Bedmap2: improved ice bed, surface and thickness datasets for Antarctica. *The Cryosphere*, 7:375–393.

Gertner, J. (2019). https://www.wired.com/story/the-top-secret-cold-war-project-that-pulled-climate-science-from-the-ice/. Excerpted from the book: "The Ice at the End of the World: An Epic Journey into Greenland's Buried Past and Our Perilous Future", by Jon Gertner. 2019.

Goldstein, M. and Rougier, J. (2004). Probabilistic Formulations for Transferring Inferences from Mathematical Models to Physical Systems. *SIAM journal on scientific computing*, 26(2):467–487.

Goldstein, M. and Rougier, J. (2006). Bayes Linear Calibrated Prediction for Complex Systems. *Journal of the American Statistical Association*, 101(475):1132–1143.

Goldstein, M. and Wooff, D. (2007). *Bayes Linear Statistics: Theory and Methods*. Wiley Series in Probability and Statistics. John Wiley & Sons.

Gordon, C., Cooper, C., Senior, C., Banks, H., Gregory, J., Johns, T., Mitchell, J., and Wood, R. (2000). The simulation of SST, sea ice extents and ocean heat transports

in a version of the Hadley Centre coupled model without flux adjustments. *Climate Dynamics*, 16:147–168.

GRIP Members (1993). Climate instability during the last interglacial period recorded in the GRIP ice core. *Nature*, 364(6434):203.

Haylock, R. and O'Hagan, A. (1996). On Inference for Outputs of Computationally Expensive Algorithms with Uncertainty on the Inputs. *Bayesian statistics*, 5:629–637.

Haywood, A., Dowsett, H., Dolan, A., Rowley, D., Abe-Ouchi, A., Otto-Bliesner, B., Chandler, M., Hunter, S., Lunt, D., Pound, M., and Salzmann, U. (2016a). The Pliocene Model Intercomparison Project (PlioMIP) Phase 2: scientific objectives and experimental design. *Climate of the Past*, 12(3):663–675.

Haywood, A. M., Dowsett, H. J., and Dolan, A. M. (2016b). Integrating geological archives and climate models for the mid-Pliocene warm period. *Nature Communications*, 7.

Haywood, A. M., Dowsett, H. J., Robinson, M. M., Stoll, D. K., Dolan, A. M., Lunt, D. J., Otto-Bliesner, B., and Chandler, M. A. (2011). Pliocene Model Intercomparison Project (PlioMIP): experimental design and boundary conditions (Experiment 2). *Geoscientific Model Development*, 4(3):571–577.

Helsen, M., Van De Berg, W., Van De Wal, R., Van Den Broeke, M., and Oerlemans, J. (2013). Coupled regional climate–ice-sheet simulation shows limited Greenland ice loss during the Eemian. *Climate of the Past*, 9(4):1773–1788.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer Model Calibration Using High-Dimensional Output. *Journal of the American Statistical Association*, 103(482):570–583.

Holden, P. B., Edwards, N. R., Rangel, T. F., Pereira, E. B., Tran, G. T., and Wilkinson, R. D. (2018). PALEO-PGEM v1.0: A statistical emulator of Pliocene-Pleistocene climate. *Geoscientific Model Development Discussions*, 2018:1–26.

Horn, R. A. and Johnson, C. R. (2012). *Matrix Analysis*. Cambridge University Press, Third edition.

IPCC (2007). *Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. IPCC, Geneva, Switzerland.

IPCC (2013). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Jacod, J. and Protter, P. (2000). *Probability Essentials*. Universitext. Springer.

Jahn, A. (2018). Reduced probability of ice-free summers for 1.5°C compared to 2°C warming. *Nature Climate Change*, 8:409–414.

Johnsen, S. J. and Vinther, B. M. (2007). Ice core records – Greenland stable isotopes. In Elias, S. A., editor, *Encyclopedia of Quarternary Sciences*, pages 1250–1258. Elsevier.

Johnson, J., Gosling, J., and Kennedy, M. (2011). Gaussian process emulation for second-order Monte Carlo simulations. *Journal of Statistical Planning and Inference*, 141(5):1838–1848.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer, second edition.

Kavetski, D. (2019). Parameter Estimation and Predictive Uncertainty Quantification in Hydrological Modelling. *Handbook of hydrometeorological ensemble forecasting*, pages 481–522.

Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society*, 63(3):425–464.

Kent, J. T. (1989). Continuity Properties for Random Fields. *Annals of Probability*, 17:1432–1440.

Kocis, L. and Whiten, W. J. (1997). Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 23(2):266–294.

Kopp, R. E., DeConto, R. M., Bader, D. A., Hay, C. C., Horton, R. M., Kulp, S., Oppenheimer, M., Pollard, D., and Strauss, B. H. (2017). Evolving understanding of Antarctic ice-sheet physics and ambiguity in probabilistic sea-level projections. *Earth's Future*, 5(12):1217–1233.

Kopp, R. E., Simons, F. J., Mitrovica, J. X., Maloof, A. C., and Oppenheimer, M. (2009). Probabilistic assessment of sea level during the last interglacial stage. *Nature*, 462(7275):863.

Lang, S. (1987). *Linear Algebra*. Undergraduate Texts in Mathematics. Springer, Third edition.

Langebroek, P. M. and Nisancioglu, K. H. (2016). Moderate Greenland ice sheet melt during the last interglacial constrained by present-day observations and paleo ice core reconstructions. *The Cryosphere Discussions*.

Laskar, J., Robutel, P., Joutel, F., Gastineau, M., Correia, A., and Levrard, B. (2004). A long-term numerical solution for the insolation quantities of the Earth. *Astronomy and Astrophysics*, 428(1):261–285.

Lee, L. A., Carslaw, K. S., Pringle, K. J., and Mann, G. W. (2012). Mapping the uncertainty in global CCN using emulation. *Atmospheric Chemistry and Physics*, 12(20):9739–9751.

Lee, L. A., Carslaw, K. S., Pringle, K. J., Mann, G. W., and Spracklen, D. V. (2011). Emulation of a complex global aerosol model to quantify sensitivity to uncertain parameters. *Atmospheric Chemistry and Physics*, 11(23):12253–12273.

Lee, P. M. (2012). *Bayesian Statistics: An Introduction*. Wiley Publishing, Fourth edition.

Levitus, S. (1982). Climatological Atlas of the World Ocean. *NOAA Profess. Pap.*, 13:1–173.

Lord, N. S., Crucifix, M., Lunt, D. J., Thorne, M. C., Bounceur, N., Dowsett, H., O'Brien, C. L., and Ridgwell, A. (2017). Emulation of long-term changes in global climate: application to the late Pliocene and future. *Climate of the Past*, 13(11):1539–1571.

Lunt, D. J., Haywood, A. M., Schmidt, G. A., Salzmann, U., Valdes, P. J., and Dowsett, H. J. (2010). Earth system sensitivity inferred from Pliocene modelling and data. *Nature Geoscience*, 3(1):60.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. Probability and Mathematical Statistics. Academic Press Inc.

Mattax, C. C. and Dalton, R. L. (1990). *Reservoir Simulation*. Monograph, Volume 13. Society of Petroleum Engineers.

Menemenlis, D., Hill, C., Adcrocft, A., Campin, J.-M., Cheng, B., Ciotti, B., Fukumori, I., Heimbach, P., Henze, C., Köhl, A., et al. (2005). NASA Supercomputer Improves Prospects for Ocean Climate Research. *Eos, Transactions American Geophysical Union*, 86(9):89–96.

Milanković, M. (1930). *Mathematische Klimalehre und Astronomische Theorie der Klimaschwankungen*. Handbuch der Klimatologie. Bornträger, Berlin.

NEEM Community Members (2013). Eemian interglacial reconstructed from a Greenland folded ice core. *Nature*, 493:489–494.

NGRIP Members (2004). High-resolution record of Northern Hemisphere climate extending into the last interglacial period. *Nature*, 431(7005):147–151.

Oakley, J. and O'Hagan, A. (2002). Bayesian Inference for the Uncertainty Distribution of Computer Model Outputs. *Biometrika*, 89(4):769–784.

Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769.

O'Hagan, A. (1978). Curve Fitting and Optimal Design for Prediction. *Journal of the Royal Statistical Society*, 40(1):1–42.

O'Hagan, A. (1992). Some Bayesian Numerical Analysis. In *Bayesian Statistics 4*, pages 345–363. Oxford University Press.

O'Hagan, A., Kennedy, M., and Oakley, J. (1998). Uncertainty Analysis and other Inference Tools for Complex Computer Codes. In *Bayesian Statistics 6*. Oxford University Press.

Øksendal, B. (1998). *Stochastic Differential Equations: An Introduction with Applications*. Springer, Fifth edition.

Otto-Bliesner, B. L., Marshall, S. J., Overpeck, J. T., Miller, G. H., Hu, A., and CAPE Members, L. I. P. (2006). Simulating Arctic climate warmth and icefield retreat in the last interglaciation. *science*, 311(5768):1751–1753.

Prescott, C. L. (2017). *Orbital forcing and its importance in understanding the warm Pliocene*. PhD thesis, University of Leeds.

Prescott, C. L., Haywood, A. M., Dolan, A. M., Hunter, S. J., Pope, J. O., and Pickering, S. J. (2014). Assessing orbitally-forced interglacial climate variability during the mid-Pliocene Warm Period. *Earth and Planetary Science Letters*, 400:261–271.

Quiquet, A., Ritz, C., Punge, H., and Salas y Mélia, D. (2013). Greenland ice sheet contribution to sea level rise during the last interglacial period: a modelling study driven and constrained by ice core data. *Climate of the Past*, 9(1):353–366.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. The MIT Press, Massachusetts Institute of Technology.

Reynolds, R. W. and Smith, T. M. (1995). A high-resolution global sea surface temperature climatology. *Journal of Climate*, 8(6):1571–1583.

Robinson, A., Calov, R., and Ganopolski, A. (2011). Greenland ice sheet model parameters constrained using simulations of the Eemian Interglacial. *Climate of the Past*, 7:381–396.

Rougier, J. and Goldstein, M. (2014). Climate Simulators and Climate Projections. *Annual Review of Statistics and Its Application*, 1(1):103–123.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical science*, 4(4):409–423.

Salter, J. M., Williamson, D. B., Scinocca, J., and Kharin, V. (2019). Uncertainty Quantification for Computer Models With Spatial Output Using Calibration-Optimal Bases. *Journal of the American Statistical Association*, 0:1–15.

Salzmann, U., Dolan, A. M., Haywood, A. M., Chan, W.-L., Voss, J., Hill, D. J., Abe-Ouchi, A., Otto-Bliesner, B., Bragg, F. J., Chandler, M. A., et al. (2013). Challenges in quantifying Pliocene terrestrial warming revealed by data–model discord. *Nature Climate Change*, 3(11):969.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. Springer.

Shepherd, A., Ivins, E. R., Geruo, A., Barletta, V. R., Bentley, M. J., Bettadpur, S., Briggs, K. H., Bromwich, D. H., Forsberg, R., Galin, N., et al. (2012). A reconciled estimate of ice-sheet mass balance. *Science*, 338(6111):1183–1189.

Shiryaev, A. N. (1996). *Probability*. Graduate Texts in Mathematics. Springer, Second edition.

Stone, E., Lunt, D., Annan, J., and Hargreaves, J. (2013). Quantification of the Greenland ice sheet contribution to Last Interglacial sea level rise. *Climate of the Past*, 9(2):621–639.

Tran, G. T., Oliver, K. I., Sóbester, A., Toal, D. J., Holden, P. B., Marsh, R., Challenor, P., and Edwards, N. R. (2016). Building a traceable climate model hierarchy with multi-level emulators. *Advances in Statistical Climatology, Meteorology and Oceanography*, 2(1):17–37.

United Nations (2015). FCCC/CP/2015/L.9/Rev.1. Adoption of the Paris Agreement.

Van den Broeke, M. R., Enderlin, E. M., Howat, I. M., Kuipers Munneke, P., Noël, B. P., Jan Van De Berg, W., Van Meijgaard, E., and Wouters, B. (2016). On the recent contribution of the Greenland ice sheet to sea level change. *The Cryosphere*, 10(5):1933–1946.

Vernon, I., Goldstein, M., and Bower, R. G. (2010). Galaxy formation: a Bayesian uncertainty analysis. *Bayesian analysis*, 5(4):619–669.

Williams, S. J. and Gutierrez, B. T. (2009). Sea-level rise and coastal change: Causes and implications for the future of coasts and low-lying regions. *Shore & beach*, 77(4):13–21.

Williamson, D., Goldstein, M., Allison, L., Blaker, A., Challenor, P., Jackson, L., and Yamazaki, K. (2013). History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble. *Climate Dynamics*, 41(7-8):1703–1729.

Wilson, A., Dent, C., and Goldstein, M. (2018). Quantifying uncertainty in wholesale electricity price projections using bayesian emulation of a generation investment model. *Sustainable Energy, Grids and Networks*, 13:42–55.

Zhang, W., Zhou, T., Zou, L., Zhang, L., and Chen, X. (2018). Reduced exposure to extreme precipitation from 0.5°C less warming in global land monsoon regions. *Nature Communications*, 9:1–8.

Zhou, S., Chen, D., Cai, W., Luo, L., Low, M. Y. H., Tian, F., Tay, V. S.-H., Ong, D. W. S., and Hamilton, B. D. (2010). Crowd Modeling and Simulation Technologies. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 20(4):20.