POLYTECH SORBONNE - LOCEAN-IPSL

Rapport de stage de fin d'études

History Matching for climate model tuning: experiments on the Lorenz 96 toy model

Author:
H. Durand
ID: 3672141

Supervisors:
R. Lguensat
J. Deshayes
V. Balaji

Septembre 2021

First of all, I would like to thank the whole team that supervised me, Redouane, Julie and Balaji, for the kindness they showed me and the precious advice they gave me. I would particularly like to thank Redouane who, in spite of the remote working, allowed me to work in good conditions, always listened to me and knew how to direct this work so that I would not go too far astray.

Abstract

This report presents the results obtained and the research avenues developed during my end-of-study internship at the LOCEAN-IPSL laboratory. The main problem it seeks to address is whether the tuning method known as History Matching or Iterative Refocussing is suitable for calibrating coupled ocean-atmosphere models. We show that in the framework of a toy model, Lorenz-96, which can, with some limitation, be assimilated to a simplified version of an Atmosphere Ocean General Circulation Model, the History Matching tuning method allows to significantly reduce the model's parameter search space. We also show that these results are valid for an AMIP- or OMIP-style experiment where, in the first case, a simplified atmosphere model is forced by ocean observations and in the second a simplified ocean model is forced by atmospheric observations. Finally, we propose two more general results on the History Matching method which we find interesting. First, we show that it is possible to significantly reduce the number of metrics used in History Matching by using a linear (Empirical Orthogonal Functions) or non-linear (Autoencoders) dimensionality reduction method. Furthermore, although further work is needed to validate this point, we propose two new emulators that seem to show some good properties to replace Gaussian Process Regressors in History Matching, Random Forest and Bayesian Neural Networks.

Contents

1 Context

This report presents the relevant and/or promising results obtained during my end-of-study internship in the LOCEAN-IPSL laboratory as part of the NEMO R&D team. The LOCEAN laboratory conducts studies on physical and biogeochemical processes in the ocean and their role in climate in interaction with marine ecosystems. It actively participates in international collaborations such as the World Climate Research Program (WCRP) which provides essential work for the IPCC and more generally for the projection of future climate change. The NEMO R&D team focuses on the three 'compartments' of the ocean, namely the physical ocean, the sea ice and the marine biogeochemistry.

This internship was supervised by three tutors, R. Lguensat, J. Deshayes and V. Balaji, leader of HRMES-MOPGA project (https://hrmes-mopga.github.io/), who collaborates within the laboratory on issues related to the study of the ocean and climate in a more general way with machine learning.

2 Introduction

Climate models or Earth System Models (ESMs) have become central to the study of climate evolution, both for the assessment of past climates and for projections of future climate. These models were among the first applications of numerical computation in the 1950s (see Platzman [1]) when the use of the "super-computers" of the time enabled the field of weather and climate prediction to experience a real boom. The structure of these models has become more complex over the last few decades, first including ocean circulation (see Manabe and Bryan [2]) in 1969, then the contribution of the radiation balance modified by human forcing linked to CO_2 emissions (see Manabe and Wetherald [3]) in 1975, leading finally to the creation of the Intergovernmental Panel for Climate Change (IPCC) in 1988, whose mission is "[...] to assess, in a systematic, clear and objective manner, the scientific, technical and socio-economic information needed to improve our understanding of the risks associated with human-induced global warming [...]".

The various components of the ESMs are generally modelled by systems of partial differential equations (PDEs) describing various processes such as fluid mechanics (described by the Navier-Stockes equations) or thermodynamics for modelling the ocean and atmosphere or biological and chemical processes describing marine and terrestrial ecosystems. These processes encompass spatial and temporal scales of different order, ranging from the collision between cloud particles of the order of a micron in size to the deep circulation of ocean, of the order of 1000 to 10000 km. The limited computing power of today's supercomputers does not allow the creation of models representing the entire Earth system at a sufficiently small scale to model small-scale processes such as cloud formation or the formation and circulation of plankton. Furthermore, human contributions to climate change are now widely accepted and their uncertain evolution complicates the modellers' projections.

The two issues raised above (scale and human forcing) are generally solved by using parameterization methods, where small-scale processes and human forcing are modelled by parameters that are considered unknown and that it is then necessary to estimate with regard to the different observations of the climate system at our disposal. Some physical processes also include unknown parameters in their structure, which similarly need to be estimated using field observations.

This work explores the application of a parameter estimation methodology, known as History Matching (HM), to coupled ocean-atmosphere models. This step in the evaluation of climate models is also referred to as 'tuning' or 'calibration' in the literature and refers to the search for the most likely parameters based on different observations of the climate system. The methodology generally used for all of these methods involves a step where a number of simulations are run with different parameters and then an evaluation step where the outputs of the simulation are compared with observations of the climate system. The parameters selected are then those that allow the numerical model to generate outputs that are closest to the observed state of the climate. We evaluate the application of the HM for tuning coupled ocean-atmosphere models using a toy model, the Lorenz-96, as a simplified version of the former.

One of the main problems in tuning these models is the high computational cost

of simulating climate models. The numerical model is therefore often replaced by a statistical model (called an emulator) whose computational cost is much lower, thus allowing a larger number of parameters sets to be tested. This is especially important when the number of unknown parameters is large and it is necessary to test a large combination of parameters sets to cover the parameters space satisfactorily.

We are also interested in exploring the application of different statistical models from the machine learning community, such as Random Forest (RFs) or Gaussian Process (GPs), as emulators of the numerical model for the tuning of atmosphere-ocean coupled general circulation model (AOGCMs) with History Matching.

3 Scope

The first part of this work is exploratory and seeks to find out to what extent HM tuning is applicable to coupled models (such as AOGCMs) by applying it to the calibration of Lorenz-96. An explanation of the History Matching method will be given in Subsect. 4.1 then a description of the Lorenz-96 will be given in Subsect. 4.2. We will therefore evaluate this methodology in a classical framework and then in the case of Atmosphere Model Intercomparison project (AMIP-style experiment) and Ocean Model Intercomparison Project (OMIP-style experiment) which seeks in the first case to tune an atmospheric model forced by oceanic observations and in the second case to tune an oceanic model forced by atmospheric observations. A more detailed description of those experiments is given in the Subsect. 4.4.

While linear regression models and Gaussian Process regressors have been widely studied and compared (see Salter and Williamson [4] and Williamson et al. [5]) as statistical emulators for climate model calibration, it seems that few recent models from the machine learning community have been studied for this task. Some of them however show very good performances in a large number of tasks and seems adapted as statistical emulators for history recalibration - in that they allow to estimate the mean of predictions as well as their uncertainties. A major limitation of Gaussian Processes is their computational complexity as the inversion of the covariance matrix required during the learning phase (see 4.3) is cubic which makes them hardly usuable for large datasets that can appears when a large number of parameters are tuned. On the other hand, linear regression models perform less well than GPRs for HM tuning (see Williamson et al. [5]) and it seems to us that some models could, with a lower learning cost than GPRs, show more interesting results than linear regressions. In particular, we believe that Bayesian neural networks and random forests have good properties and we will evaluate their performance in the context of Lorenz-96 calibration by History Matching. The Subsect. 4.3 will describe the different models evaluated in this work and how they will be tested.

Finally, in order to compare the outputs of the simulations and the field observations, it is necessary to have a certain number of metrics summarising the evolution of the system studied. Lorenz-96 is for example tuned using a set of 180 metrics (see Schneider et al. [6]), which can generate a high computational cost. Having observed that some of these metrics were highly correlated, we will finally consider the use of dimension reduction methods, namely Autoencoders (AE) and Empirical Orthogonal Functions (EOF), in order to reduce the number of metrics and thus reduce the

computational cost of training and predicting emulators. These two methods are detailed in Subsect. 4.5

4 Methodology

4.1 History Matching

The term History Matching first appeared in the oil engineering community (see Craig et al. [7]). In order to predict the future production of oil reservoirs, engineers have at their disposal complex numerical models (systems of partial differential equations) to measure the temporal evolution of water, gas and oil flows in reservoirs. However, those models must be adapted to the geological conditions and to the conditions of use of the reservoir studied and therefore have a set of adjustable parameters. In order to predict the evolution of production as reliably as possible, it is therefore necessary to find the set of parameters that best describe the reservoir conditions. To do this, the engineers use the model outputs (a set of oil, water and gas production measurements) which they will try to match with observed historical production (hence the term History Matching). To summarize, their problem is to find the set of parameters that will allow the model to best matches with observed historical production.

This leads to several problems. First of all, reservoir models are particularly expensive in terms of computation time and it is therefore only possible to test a reduced set of parameters. Secondly, the number of parameters is generally high, which combined with the low number of simulations, leads to a strong scattering of the parameter sets used for the simulation and it is therefore unlikely to have a correct representation of all possible sets of parameters.

History Matching is therefore a statistical method, which allows to answer this problem by iteratively rejecting the parameter sets considered to be the most implausible in view of the simulations they generate, the field observations and the various uncertainties expressed on the predictions of the emulator, the observations or on the structure of the numerical model itself.

History Matching is now a widely studied, published and established method and is used in many scientific and engineering fields such as galaxy formation modeling (Vernon et al. [8]), spread of infectious diseases and viruses (Andrianakis et al. [9]) and has been attracting the attention of the climate science community during the last decade.

We consider, retaining the notation of Williamson et al. [10], y the (imperfect) historical observation of the climate system such as $y = z + \epsilon_{obs}$ with z being the real historical state of the climate system and ϵ_{obs} the uncertainty about observations. Also we note by $f(x)$ the climate model for any set of parameters x in a d-dimensional space χ. As it is impossible to evaluate $f(x)$ for all $x \in \chi$ because of the continuity (at least per piece) of χ, we only have a set $F_{[n]} = f(x_1), ..., f(x_n)$ of n simulations (called perturbed physics ensemble - PPE) of the studied model corresponding to the simulation of the parameters $X_{[n]}$ (called Ensemble Design).

The choice of the Ensemble Design is called Space filling design and is detailed in section 4.1.1. The simulation of the PPE and the choice of metrics to represent it is

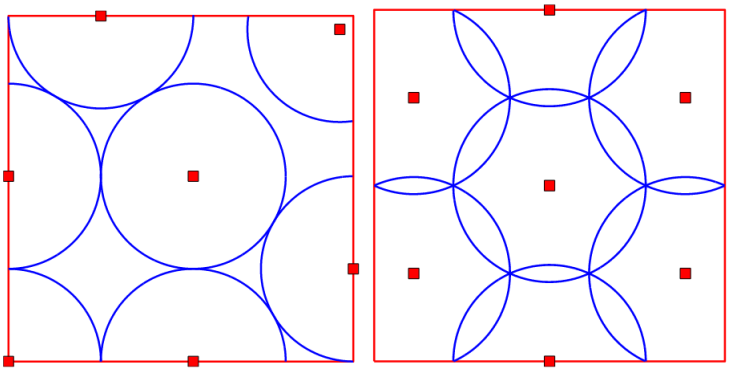

Figure 1: Maximin (left) and minimax (right) designs for 7 points in $[0,1]^2$. From Pronzato and Müller [12].

explained in section 4.1.2. We can then use the created ensembles (the metrics based on PPE and the Ensemble Design) to train a statistical model, also called emulator, so that it takes the place of the climate model to simulate the behavior of the latter with a greatly reduced computation time. This step is describe in section 4.1.3. We describe in section 4.1.4 how this emulator is used to generate the implausibility distribution over the entire parameter space. We can finally use the implausibility distribution to exclude the least plausible areas of the parameter space in order to reduce it. This being done we can reiterate all the steps previously described on the new space created as explained in the section 4.1.5.

4.1.1 Space filling design

As mentioned earlier, since our climate models are often expensive to run, it is impossible to have as many model runs as necessary to have an exhaustive sampling of the parameter space. It is therefore crucial to sample our parameter space judiciously. This step is often referred to as "Space Filling Design" in the literature and has been extensively studied for many different cases (Joseph [11], Pronzato and Müller [12]). As explained in (Pronzato and Müller [12]), the standard practice used for this is to select the parameters in such a way that they cover the χ parameter space in the most uniform way possible. This space being in general large, there are several methods for this.

In this section we will quickly detail the main methods used:

- **Geometric Sampling**

  If the considered space is one-dimensional, the space filling design seems obvious. Considering the space $\chi_1 = [0,1]$, a correct design could be $\zeta = \{\frac{i-1}{n-1} : \forall i \in 1..n\}$ or $\zeta = \{\frac{i-1}{2n-1} : \forall i \in 1..n\}$ depending on whether we consider the edges or not. The idea behind this simple example is the minimization of the distance. Let us now consider the general case with $\chi_d = [0,1]^d$. We want to sample as well as possible the set of points $\zeta = (x_1, x_2, ..., x_n)$ on χ_d. For this we have a norm (say the Euclidean norm) $< ., . >$ as a distance measure between two points $d_{ij} =< x_i, x_j >$. A simple idea could be to try to maximize the minimal distance between two points of the sample, we would have

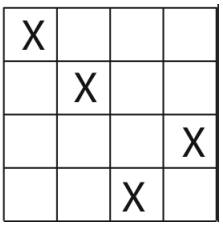

Figure 2: Latin Hypercube Sampling in 2 dimensions with 4 points.
Source : Wikipedia

$$\phi_{Mm}(\zeta) = \min_{i \neq j} d_{ij}$$

Maximizing $\phi_{Mm}(.)$ is called a *maximin-distance design* (see Johnson et al. [13]).

We can consider an other point of vue where we may attempt to minimize the maximum distance from all the points in χ_d to their closest point in ζ. This can be done by minimizing minimax-distance criterion

$$\phi_{mM}(\zeta) = \max_{x_i \in \chi_d} \min_{x_j \in \zeta} d_{ij}$$

We then speak of *minimax-distance design* of which a more complete description can also be found in Johnson et al. [13].

A comparison of the two methods is available Fig. [1].

- **Latin Hypercube Sampling**

  The Latin Hypercube Sampling (LHS) was developped by Mckay et al. [14] in 1979. The method performs sampling by ensuring that each sample is positioned in a d-dimensional space Ω as the only sample in each $(d-1)$-dimensional hyperplane aligned to the coordinates that define its position. Each sample is therefore positioned according to the position of previously positioned samples, to ensure that they do not have common coordinates in space Ω.

  The standard LHS can be taken as a starting design and then optimized according to some optimization criterion like maximin or minimax criterion describe earlier.

- **Monte-Carlo and Quasi-Monte-Carlo Sampling**

  Monte-Carlo Sampling (MCS) and Quasi-Monte-Carlo Sampling (QMCS) are pseudo random sampling methods. Unlike MCS, QMCS are designed to place sample points as uniformly as possible.

  The quasi-Monte-Carlo method is based on the same problem than the Monte-Carlo method. It approximates the integral of a squared-intergable function

f over the $n-$dimensional hypercube H^n by the average of the values of the function evaluated at a set of points $x_1, ..., x_N$:

$$\int_{H^n} f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{1}$$

The difference between Monte-Carlo method and quasi-Monte-Carlo method is that for the first the x_i are generated with pseudo-randomly sequences and for the second they are generated with some low discrepancy sequence like Halton sequence or Sobol Sequence.

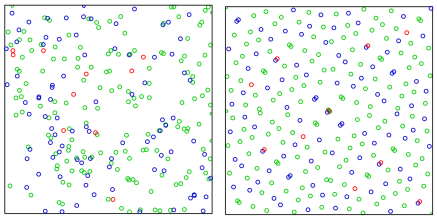

Figure 3: Monte Carlo Sampling (left) and Quasi Monte Carlo Sampling (right) for red=1,..,10, blue=11,..,100, green=101,..,256.
Source : Wikipedia

It is now widely recognized that Sobol Sampling (QMCS with with a Sobol sequence) is superior to other QMCS and MCS technics in many aspect (see Chen and Hong [15] and Kucherenko et al. [16]). For this reason, we will concentrate on this method.

The commonly used sampling method for HM is the maximin LHS (see Williamson et al. [10], Williamson et al. [5] or Vernon et al. [8]). As we have not found an explicit reason for this choice, we are interested in exploring how Sobol sampling (as represetent of QMCS methodology) will handle this step of History Matching.

The number of samples selected also plays a central role at this level. A high number of samples will allow a good modeling by the emulator but will lead to a high computation cost during the numerical simulation. A lower number will lead to a decrease in the quality of the modeling by the emulator but will reduce the computation time of the numerical simulation. It is therefore important to find a good compromise. The order of magnitude generally used for the number of samples is $10 \times p$ where p is the number of parameters (see Williamson et al. [10]).

4.1.2 Numerical simulation and metrics choice

A climate model, is generally a set of partial differential equations based on the equations of fluid mechanics (Navier-Stockes equations) and thermodynamics, but may also be based on equations describing biological or chemical phenomena. Its purpose may be to describe one of the actors in the climate (e.g. the ocean, in which case it is referred to as an Oceanic General Circulation Model, or the atmosphere, in which case it is referred to as an Atmospheric General Circulation Model), a region

of the climate (e.g. a region of the Earth) or the Earth as a whole (e.g. Earth System Models)

The numerical solution scheme of these equations may be of some importance at this stage. Indeed, an Euler scheme will have a larger integration error than a Runge-Kutta scheme (RK4 for example) and this error propagating as a structural error of the model (see 4.1.4) will lead to a different result when calculating implausibility.

Unlike typical calibration methods, which present the parameters search problem as an optimisation problem where the objective is to find the set of parameters that allows the model to be closest to a set of metrics, HM seeks to rule-out areas of the parameter space that are inconsistent in reproducing the chosen metrics.

In the climate science community, the term metrics refers to the measurements that the modeller chooses to report on the state of the climate system. They can be of different kinds (scalar, vector or tensor fields of different quantities, volume-integrated means and anomaly fields, heat and salt transport metrics, etc...). It is then important to clarify what is meant when two metrics are said to be consistent or inconsistent, especially when talking about vector or tensor fields.

Following Williamson et al. [5], there is three crucial ingredients when selecting metrics for model tuning :

- It is judged physically reasonable/desirable and important to use the proposed metric to constrain the model by the developers.
- We have a quantification of the uncertainty in the metrics. Without this, we do not know how close we are nor when we have succeeded.
- The metric actually provides sufficient constraint on the parameter space: certain metrics may be physically important, but do not vary sufficiently as the model parameters are varied to make them useful in tuning (McNeall et al. [17]).

4.1.3 Statistical emulators

One of the problems we quickly find ourselves confronted with the HM – and with calibration methods more generally – is that this method requires a very large number of simulations in order to be able to eliminate all the areas of the parameter space that do not correspond with the observations. However, the simulations in question are generally very costly in terms of computing time and it is therefore impractical in practice to generate the entire data set with the numerical model. This is why a statistical emulator is generally used in calibration methods, the aim of which is to replace the numerical simulator by generating the metrics from a certain set of parameters in a much shorter time.

For this purpose, we need to run a smaller ensemble of the model by using one of the sampling methods discussed above, and use that ensemble to train the statistical emulator which will take the place of numerical model when exploring the parameter space.

In the context of the HM, an emulator must be able to provide us with both a good estimate of the metrics and a measure of the uncertainty in that prediction. From a statistical point of view, our emulator must therefore be able to provide us

with the estimated expectation on the metrics for a given set of parameter x, noted $E[f(x)]$ and an estimate of the variance on them, noted $Var[f(x)]$.

A common choice for an emulator, following Williamson et al. [5], could be

$$f_i(x) = \sum_j \beta_{ij} g_j(x) + \epsilon_i(x) \tag{2}$$

$$\epsilon_i(x) \sim \mathrm{GP}(0, C_i(.,.;\phi_i i)) \tag{3}$$

where the vector $g(x)$ contains specified basis functions in x, the matrix β is a set of coefficients to be fitted. The GP stands for a Gaussian process, with C_i as pre-specified covariance functions, and with the ϕ_i being their parameters.

The search for new statistical models as emulators for the MH being one of the focal points of this report, we will develop in a more advanced way the different models considered in section 4.3.

4.1.4 Implausibilty and parameters space reduction

The simplest idea to find the set of parameters that allow the model to get as close as possible to the real state of the climate system seems to be to define a distance measure between the model output $f(x)$ and the real state of the system z. We could thus use our emulator to find the set of parameters that minimize the distance between the model output and the system state. Following Williamson et al. [5], we could then consider the following optimization problem

$$x^* = \underset{x}{\operatorname{argmin}} \, \|z - f(x)\|_f$$

Where $\|.\|_f$ is a norm taking into account the different uncertainties discussed previously. For example, we may consider the Mahalanobis distance

$$\|z - f(x)\|_f = (z - f(x))^T \, \mathrm{Var}[z - f(x)]^{-1} (z - f(x))$$

As stated in Williamson et al. [5], because we are using our emulator, we do not have access to entire distribution of our model $f(x)$ but only to the expectation $\mathrm{E}[f(x)]$ and to the variance $\mathrm{Var}[f(x)]$.

We can reformulate the distance, using the prediction of our emulator

$$\begin{aligned}
\|z - \mathrm{E}[f(x^*)]\|_f &= (z - \mathrm{E}[f(x^*)])^T \, \mathrm{Var}[z - \mathrm{E}[f(x^*)]]^{-1} (z - \mathrm{E}[f(x^*)]) \\
&= (z - m^*(x^*))^T \, \mathrm{Var}[(z - y) + (y - f(x^*)) + f(x^*) - \mathrm{E}[f(x^*)]^{-1} (z - m^*(x^*)) \\
&= (z - m^*(x^*))^T (V_e + V_\eta + \mathrm{Var}[f(x^*)])^{-1} (z - m^*(x^*))
\end{aligned}$$

We thus ensure that if our distance measure is large for a given set of parameters x^*, the outputs of our model are too far from the observations and those taking into account the different uncertainties that we have on the climate model, on the observations and on the predictions of the emulator. Thus the small values of $\|z - \mathrm{E}[f(x^*)]\|_f$ appear in two cases only: the distance between the prediction of the model and the real state of the system is small or one of the uncertainties is too high.

We will call this distance measure implausibility and notate it $I(x) = \|z - \mathrm{E}[f(x^*)]\|_f$. In order to rule out some region of the parameter space it is now necessary to decide the value from which the implausibility is too large.

4.1.5 Refocussing

We refer to the term refocussing by iteratively generating an EPP and a Design Ensemble on which to train a statistical emulator to then ruled-out part of the parameter space. The iterative aspect of the HM provides a certain flexibility that other approaches may not. After having significantly reduced the parameter space with a set of metrics describing well the general tendencies of the system we could indeed try to reduce it by using metrics describing some more local aspects of it over several iterations in order to reduce the parameter space even more.

However, There are still some methodological aspects on which there is no consensus. Firstly, the stopping criteria are not clearly defined and the approach therefore generally varies from one problem to another, it is usually pragmatic and limited by computational resources. The process will therefore most often be stopped when it is felt that performing one more wave would not reduce the parameter space sufficiently compared to the computational time that it would require. Also, as stated in Williamson et al. [5], "*when the emulator variance is largely smaller than the denominator in the implausibility calculation, then it is unlikely that further waves will change the implausibility very much*" and it may be unreasonable to perform a new wave. Secondly, a difficulty arises with multi-wave design after the first wave. In fact, it is no longer possible to use use LHS to sample the NROY space as it is in general not a hyperrectangle and may contain several disconnected regions.

Our approach to this work is to sample the entire parameter space with enough samples to leave approximately the desired number after rejecting those with an implausibility score greater than 3 for each emulator. Since the sampling is not perfectly uniform, we slightly overestimate the number of samples needed and then perform a random draw on the samples remaining after exclusion by History Matching. This is a simple and a non perfect strategy, but it was used for example in [8].

4.1.6 Not Ruled Out Yet Space

As the number of parameters to be tuned can be high, it can be difficult to visualise the space obtained after each wave of History Matching and to know if it has not excluded ground truth parameters. There is a type of display for this that we will use a lot in the results of this report and it is therefore important to understand how they are constructed and how to read them. After training the emulator, the implausibility score is calculated over a large number of echnatillons in the initial parameter space. The aim is then to simply visualise which areas of the parameter space have been rejected and which have not. Since the number of parameters can be greater than three (which will be the case in this report), it is impossible to simply display the parameter space directly with a single graph. We therefore represent the parameter space by displaying graphs of the parameters two by two.

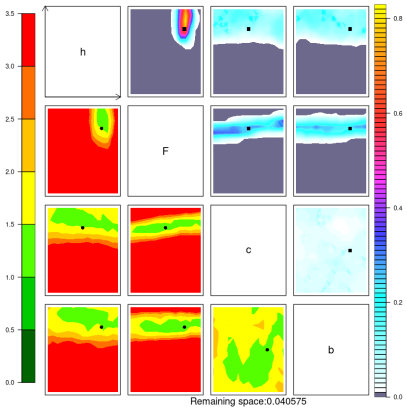

Figure 4: Not Ruled Out Space of Lorenz-96 after 1 wave of History Matching

For example, we can see Fig. 6 on the lower left-hand side a representation of a NROY space after a History Matching wave on a four parameter model (h, F, c and b). The red areas are the areas for which the implausibility is greater than 3 and the ground truth parameters are represented by the black points in the graphs. By numbering the graphs from top left to bottom right with $i = 1, 2, 3, 4$ for the lines and $j = 1, 2, 3, 4$ and noting them $I_{(i,j)}$ we can see on the graph $I_{(2,1)}$ the representation of the parameter h as a function of F. We can see on this graph that globally the parameter F is found (we can see that the graphs $I_{(2,1)}$, $\mathrm{I}_{(3,2)}$ and $I_{(4,2)}$ which represents the space F that the majority of the space has been rejected) whereas the parameters c and b are not very much (the space representing c as a function of b does not reject any zone).

4.2 Numerical model - Lorenz 96

As explained above, one of the objectives of this work is to evaluate the extent to which History Matching can be used for the parameter estimation in coupled models (for example, for a Ocean Atmosphere General Circulation Model). For all our experiments, we use a toy model, the two-layer Lorenz-96 which has been widely studied by the data assimilation community (see Lguensat et al. [18], Schneider et al. [6], Ott et al. [19], Lorenz [20], Anderson [21], Gagne et al. [22]). This choice is based on different considerations which will be detailed in subsection 4.2.2.

4.2.1 Model description and metrics

The two-layer Lorenz-96 is a dynamic system composed of two simple ODEs proposed by Edward Lorenz in 1996 in Lorenz [23] to study the pedictibility of weather and climate systems.

Using the notation of Schneider et al. [6], we can describe the model by the following ODEs

$$\frac{dX_k}{dt} = \underbrace{-X_{k-1}\left(X_{k-2} - X_{k+1}\right)}_{\text{Advection}} \underbrace{-X_k}_{\text{Diffusion}} \underbrace{+F}_{Forcing} \underbrace{-hc\bar{Y}_k}_{Coupling} \tag{4}$$

$$\frac{1}{c}\frac{dY_{j,k}}{dt} = \underbrace{-bY_{j+1,k}\left(Y_{j+2,k} - Y_{j-1,k}\right)}_{\text{Advection}} \underbrace{-Y_{j,k}}_{\text{Diffusion}} \underbrace{+\frac{h}{J}X_k}_{\text{Coupling}} \tag{5}$$

where $\bar{Y}_k = \frac{1}{J}\sum_{j=1}^{J} Y_{j,k}$. Following Lorenz [23] we let $K = 36$ et $J = 10$ so that there are 10 small sectors, each degree of longitude in length, in one large sector. So we have a set of 4 parameters that we will try to tune: h, F, b and c. Again, following Lorenz [23] we set the truth value of c and b to 10 implying that the convective scales tend to fluctuate 10 times as rapidly as the larger scales, while their typical amplitude is 1/10 as large. Also we let $h = 1$ and chose $F = 10$ as it is sufficent to make X and Y vary chaotically (Lorenz [23]). Note that contrary to what is proposed in Lorenz [23] we keep here the external forcing parameter F in addition to the forcing exerted by Y on X.

Following Rasp [24], this system is integrated using a Runge–Kutta fourth order scheme with a time step of 0.001. We used the L96 Python code accompanying the paper of Rasp [24] (see https://github.com/raspstephan/Lorenz-Online) in this work.

As stated in Schneider et al. [6], the quadratic nonlinearities in this dynamical system conserve the quadratic invariants ("energies") $\sum_k X_k^2$ and $\sum_j Y_{j,k}^2$. Also, the interaction between the slow and fast variables conserves the "total energy" $\sum_k (X_k^2 + \sum_j Y_{j,k}^2)$. Energies are prevented from decaying to zero by the external forcing F . After a certain number of iteration, the system approaches a statistically steady state (called attractor) in which driving by the external forcing F balances the linear damping.

Always following Schneider et al. [6], we will use the metrics

$$\boldsymbol{f}(X, Y) = \begin{pmatrix} X \\ \bar{Y} \\ X^2 \\ X\bar{Y} \\ \bar{Y^2} \end{pmatrix} \tag{6}$$

The priors on those parameters for this work will be the uniform distributions described by Tab. [1].

Table 1: Prior intervals for the parameters

heightParams	Prior	True
F	[-20,20]	10
h	[-2,2]	1
c	[0,20]	10
b	[-20,20]	10

Also we will discuss in the result to what extent those metrics are appropriate for this problem.

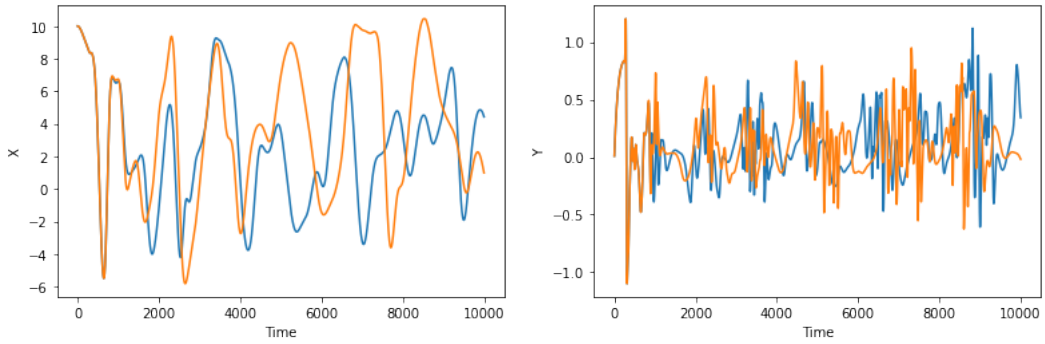

Figure 5: Evolution of $X_0(t)$ (left) and $Y_{0,0}(t)$ (right) for the 10 first iterations (with dt=0.001) for ground truth parameters with different initializer, $X_k(t=0) = 10, \forall k \neq 18$ and with $X_{18}(t=0) = 1.001$ (blue) and $X_{18}^0 = 1.002$ (orange)

4.2.2 Interest

This model in addition to the fact that it has been widely studied on several aspects - as a toy model of chaotic systems for the study of dynamical system forecasting, parameterization or data assimilation - presents two major interests for our studies.

First of all, its chaotic aspects make it a particularly difficult model for prediction.

We can indeed see (figure 5) that the system is very sensitive to the initial conditions, a slight variation on the initial state of the system leads to uncorrelated variations after a few iterations.

Secondly, as its two components (X and Y) evolve at different spatial and temporal scales, it can be assimilated to a simplified version of a coupled ocean-atmosphere model where the slow component X would represent the state of the ocean and the fast component Y the state of the atmosphere. The slow variables X may be viewed as resolved-scale variables and the fast variables Y as unresolved variables in an ESM. Each of the K slow variables X_k may represent a property such as surface air temperature in a cyclic chain of grid cells spanning a latitude circle. Each slow variable X_k affects the J fast variables $Y_{j,k}$ in the grid cell, which might represent cloud-scale variables such as liquid water path in each of J cumulus clouds. In turn, the mean value of the fast variables over the cell, Y_k, feeds back onto the slow variables X_k. The strength of the coupling between fast and slow variables is controlled by the parameter h, which represents an interaction coefficient, for example, an entrainment rate that couples cloud-scale variables to their large-scale environment. Time is nondimensionalized by the linear-damping time scale of the slow variables, which we nominally take to be 1 day, a typical thermal relaxation time of surface temperatures. The parameter c controls how rapidly the fast variables are damped relative to the slow; it may be interpreted as a microphysical parameter controlling relaxation of cloud variables, such as a precipitation efficiency. The parameter F controls the strength of the external large-scale forcing and b the amplitude of the nonlinear interactions among the fast variables.

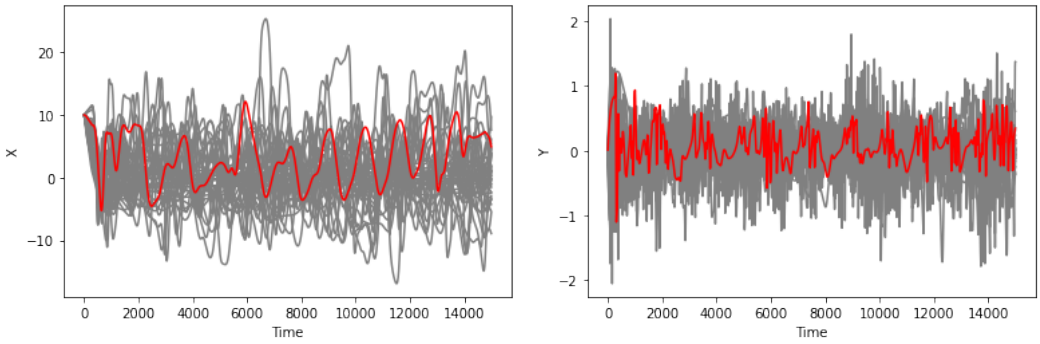

Figure 6: Evolution of $X_0(t)$ (left) and $Y_{0,0}(t)$ (right) for the 15 first iterations (with dt=0.001) for ground truth parameters (red) and 40 samples of tested parameters (grey)

4.2.3 Limits

In this internship we intend to reproduce the process of automated calibration of a system with multiple temporal scales, in order to help us anticipate what might transpire when the HM method is applied to a full coupled climate model. We underline that this is a gross simplification, as the real system has many interlocking feedbacks and more than just two timescales. The forcing F is also stationary here, whereas the external forcing in the real system has both short-lived and slowly-evolving components. Nevertheless, we will demonstrate that there are important lessons to be learned regarding the coupling of multiscale systems, and this article will walk through the steps of calibration to illustrate the strengths of the objective approach, as well aspects requiring caution.

4.3 Statistical emulators

4.3.1 Definition

In order to simplify the notations, we will refer to the PPEs previously noted $F_{[n]} = \{f(x_1), ..., f(x_n)\}$ by Y and to the Ensemble Design previously noted $X_{[n]} = \{x_1, ..., x_n\}$ by considering that they form a training set of n samples noted (X, Y) of which X represents the inputs and Y the outputs. We draw the reader's attention to the fact that the variables X and Y described here are not related to those mentioned in the description of Lorenz-96. We use this notation because it is the one commonly used in the field of statistical learning and it avoids making the writing more cumbersome.

In order to use HM, the emulator used must provide the expectation $E[f(x)]$ and the variance $var[f(x)]$ for any parameter $x \in \chi$. In this respect, the HM method is more permissive than the Bayesian calibration because it only requires access to the probability distribution of $f(x)$ as a whole.

4.3.2 Commonly used emulators

As mentioned earlier (see 4.1.3), the reference statistical models for HM are Gaussian Process Regressors (GPRs). Those are widely used by the data assimilation and Uncertainty Quantification (UQ) communities to emulate computationally expensive numerical models, particularly when few training samples are available.

Despite this, it seems to us that linear regression models are of interest for several reasons. Firstly, GPRs are generally trained on the residuals of a linear regression and it is therefore necessary to understand the latter. Secondly, linear regression models are simpler to train which, in the case of a large number of samples and/or metrics can be important. On the other hand, linear regression is a simple model and generally known by the majority of the scientific communities, it is thus a particularly accessible method both in its implementation and in its understanding. Finally, as proposed in Salter and Williamson [4], it can be interesting in an iterative approach to carry out a certain number of waves using a linear regression model as emulator in order to identify the main trends of the studied system and then to refine the parameterization on several waves with a GPR as emulator. This may saves time without reducing the performance of the HM.

- **Linear Regression**

The linear regression, following Andrianakis et al. [9] notation, might be described by:

$$f(x) = \sum_{i=1}^{q} h_i(x)\beta_i + \epsilon(x), \tag{7}$$

where $h_i(x)$ are functions of the inputs x, β_i are their respective coefficients and $\epsilon(x)$ is residual noise. The term "linear" comes from the linear relationship between $h_i(x)$ and β_i. Thus the function $h_i(x)$ can take any form, whether linear, quadratic, or any other polynomial of higher degree, sinusoidal or any non-linear function. Determining the best form of $h_i(x)$ is a tough question and it can be done using different methodologies.

By noting $h(x) = (h_1(x), h_2(x), ..., h_q(x))$ and $\beta = (\beta_1, \beta_2, ..., \beta_q)^T$ we can rewrite the equation 7 as follows

$$f(x) = h(x)\beta + \epsilon \tag{8}$$

Thus, by noting H the matrix of dimension $n \times q$ having for columns $h(x_1), h(x_2), ..., h(x_n)$ the maximum likelihood estimate of β is given by

$$\hat{\beta} = (H^T H)^{-1} H^T Y \tag{9}$$

Thus the prediction of the model for a given set of parameters x^* is

$$\underset{lr}{\mathrm{E}}[f(x^*)] = h(x^*)\hat{\beta} \tag{10}$$

As previously described, it is also necessary to have an estimate of the uncertainty of the model on this prediction. Still following Andrianakis et al. [9], the maximum likelihood estimate of this uncertainty is given by

$$\underset{lr}{\mathrm{Var}}[f(x)] = (Y^T Y - Y^T H(H^T H)^{-1} H^T Y)/N \tag{11}$$

- **Gaussian Process Regressors**

We can now describe the most commonly used models for History Matching (Williamson et al. [10], Williamson et al. [5], Vernon et al. [8]), namely Gaussian Process Regressors. As explained in the introduction to this section, GPRs are generally trained on the residuals of a linear regression that is trained under the conditions described above. Thus, we can describe them as

$$f_i(x) = \sum_j \beta_{ij} g_j(x) + \epsilon_i(x) \tag{12}$$

$$\epsilon_i(x) \sim \mathrm{GP}(0, C_i(.,.;\phi_i i)) \tag{13}$$

where the vector $g(x)$ contains specified basis functions in x, the matrix β is a set of coefficients to be fitted. The GP stands for a Gaussian process, with C_i as pre-specified covariance functions, and with the ϕ_i being their parameters. One can think of the term $\sum_j \beta_{ij} g_j(x)$ as an average describing the large-scale trends of the dynamical system and the term $\epsilon_i(x)$ as a residual term, capturing the local variations around the mean function.

A common choice, for the covariance function is the separable exponential power covariance function

$$C(x_i, x_j; \phi) = \sigma^2(\nu \mathbf{1}_{x_i = x_j} + (1-\nu)\prod_{k=1}^{d} \exp\{\theta_k |x_k - x'_k|^{\kappa_k}\}) \tag{14}$$

$$\phi = \{\sigma, \nu, \theta, \kappa\} \tag{15}$$

$$\tag{16}$$

The emulator can be trained by first specying a prior distribution over the parameters of the model, knowing (β, ϕ) and update them with our train data (X, Y). Following Williamson et al. [5], the posterior distribution $f_i(x)|Y, \{\beta, \phi\}$ is

$$f_i(x)|Y, \{\beta, \phi\} \sim \mathrm{GP}(m^*(x), C^*(.,.;\phi_i))$$

with

$$m^*(x) = \sum_j \beta_{ij} g_j(x) + K(x)V^{-1}(Y - \beta_{ij} g_j(X))$$
$$C^*(x, x', \phi) = C(x, x', \phi) - K(x)V^{-1}K(x')^T$$

where V is the $n \times n$ matrix with ijth element $C(X_i, X_j; \phi)$ and $K(x)$ is the vector with jth element $C(x, X_j, \phi)$.

Thus, we have

$$\underset{gp}{\mathrm{E}}[f(x)] = m^*(x)$$
$$\underset{gp}{\mathrm{Var}}[f(x)] = C^*(x, x, \phi)$$

In this work, we use the library https://github.com/BayesExeter/ExeterUQ_MOGP for training linear regressions and GPRs.

4.3.3 Emulators from the machine learning community

We are interested here in the search for new statistical models to replace linear regressions or GPRs. We propose the study of two models: Random Forest (RF) which have been widely studied in the machine learning community during the last decades and Bayesian Neural Networks which have recently attracted some attention due to the need to provide neural networks with a good estimate of the uncertainty of their predictions (see Jospin et al. [25]).

- **Random Forest**

We will here give a quick description of Random Forests, for more details we refer the reader to the original paper (see Breiman [26]) or to Zhang et al. [27] which gives a good description.

The RF model is based on decision tree learning and aims at correcting several drawbacks of this type of learning by constructing a set of partially independent decision trees. Following Breiman [26] notations, those are constructed following this process.

Create an ensemble of B decision trees $T_1^*, ..., T_B^*$. In order to grow each tree T_i^* with some independance,

1. Bootstrap the training dataset to create $C_N^* = \{(X_i^*, Y_i^*), i = 1, ..., N\}$ by randomly, with-replacement drawing N samples.

2. Place all the training data are in the root node N.

3. Draw $mtry < p$ predictor variables from the set of all predictors, creating the ensemble of predictors S.

4. Partition N into N_1 and N_2 by selecting a predictor variable $x \in S$ and spliting cases as follow : $x \leq c$ goes in N_1 and $x > c$ cases goes in N_2. Note that x and $c \in \mathbb{R}^d$ for a multi-outputs regression with d outputs. The value of c is chosen in such a way that it maximises the inter-class variance (having subsets whose values of the target variable are as dispersed as possible).

5. For each new node $\tilde{N}$ that has more than *nodesize* cases, create two new nodes by repeating steps (3) and (4), if there is variation in the values of the response and in the values of at least one predictor. Otherwise $\tilde{N}$ become a *terminal node* of the tree T_i^*.

6. The prediction of tree T_i^* for a given $\mathbf{X}$ is calculated by applying all the partitioning rules learned by the tree during steps (2), (3) and (4) to $\mathbf{X}$ and by averaging the predictors of the training phase that are in the *terminal node* reached by $\mathbf{X}$. This prediction is noted $\hat{Y}_i^*$.

The prediction of the RF is determined by calculating the average of the predictions of each tree in the forest, for a certain input $\mathbf{X}$, it is noted

$$\underset{rf}{\mathrm{E}}[f(\mathbf{X})] = \frac{1}{B}\sum_{i=1}^{B}\hat{Y}_i^*$$

We also need to acces to the uncertainty of the RF over its prediction $\mathrm{E}_{rf}[f(\mathbf{X})]$, knowing $\mathrm{Var}_{rf}[f(\mathbf{X})]$. Several methodologies have been proposed for this purpose, Zhang et al. [27] proposes a comparison of the main ones and seems to show that the "out-of-bag" error would be one of the most interesting. The idea being to learn the error distribution $D = Y - \mathrm{E}_{rf}[f(x)]$ and thus to have access to the uncertainty on the predictions $\mathrm{Var}_{rf}[f(x)] = \mathrm{E}[(Y - \mathrm{E}_{rf}[f(\mathbf{X})]]^2] = \mathrm{E}[D^2]$. We therefore want to calculate the error D of a given prediction $\mathrm{E}_{rf}[f(\mathbf{X})]$ using a RF that has not been trained on Y. For each $Y_i, i = 1, ..., N$ we need a forest $\mathrm{RF}_{(i)}$ constructed without (X_i, Y_i). Following [27], such a forest is available for each $i = 1, ..., N$ due to the bootsrap sampling in step (1) and this forest is composed of approximately $(\frac{n-1}{n})^n \times B \approx \exp(-1) \times B \approx 0.368 \times B$ trees. For each $i = 1, ..., n$, we can use RF(i) to obtain a prediction of Y_i, denoted as $\mathrm{E}_{rf}[f(\mathbf{X})]_{(i)}$. We thus have access to the OOB error $D = \{Y_i - \mathrm{E}_{rf}[f(\mathbf{X})]_{(i)}\}_{i=1}^N$. By calculating the mean of this error squared, we access to the uncertainty of a new prediction

$$\underset{rf}{\mathrm{Var}}[f(x)] = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \underset{rf}{\mathrm{E}}[f(\mathbf{X})]_{(i)})^2$$

An important issue that may be raised is whether $0.368 \times B$ corresponds to a sufficient data set to properly assess this uncertainty knowing that we generally do not have access to large dataset with History Matching.

- **Bayesian Neural Networks**

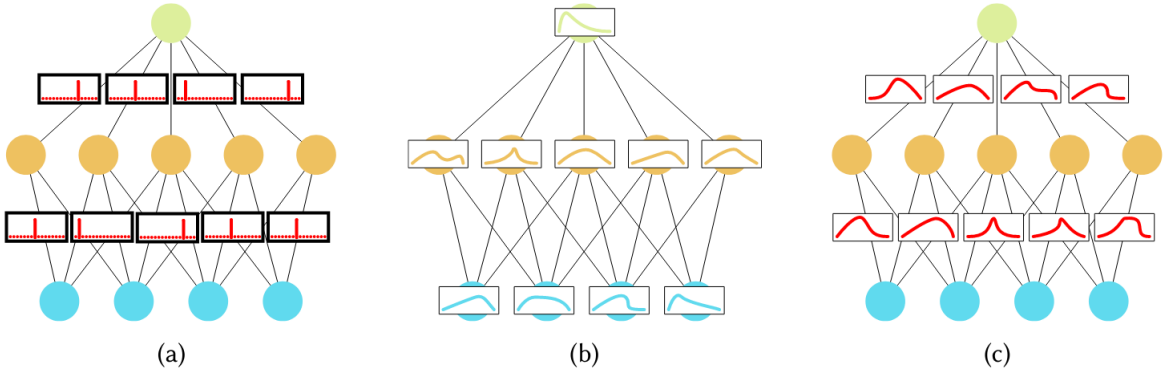

Figure 7: Artificial neural network (a), stochastic activation neural networks (b), stochastic coefficients neural networks (c). From Jospin et al. [25]

.

We will now describe Bayesian Neural Networks (BNN) which can be summarized as stochastic neural networks trained using bayesian inference. It is therefore important to recall the functioning of a classical neural network (NN).

The purpose of a neural network is to represent a function $y = NN(x)$. They are built with an input layer $l_0 = x$, which represents the input data of the model, followed by a number of hidden layers $l_i, i = 1, ..., n-1$ and an output layer $l_n = y$ which represents the data predicted by the model. In the classical NN feedforward, each layer is a linear transformation ($li = W_i l_{i-1} + b_i$) of the previous one, followed by a non-linear operation ($\sigma(.)$), known as activation function.

$$
\begin{aligned}
l_0 &= x \\
l_i &= \sigma(W_i l_{i-1} + b_i), \forall i \in [1, n-1] \\
l_n &= y
\end{aligned}
$$

There are more complex architectures involving particular layers (Convolutional Neural Networks) or whose layers are linked recursively (Recurent Neural Networks). Here we restrict ourselves to simple feedforward neural networks (Fig. 7a). A neural network is thus a set of functions isomorphic to a set of possible coefficients θ where θ represents all weights $W_i, \forall i \in [1, n-1]$ and biases $b_i, \forall i \in [1, n-1]$. The training of a NN is thus done by regressing the parameters θ using the training data set. The standard approach is to approximate a minimal cost point estimate $\hat{\theta}$ using the back-propagation algorithm.

Stochastic (or bayesian) neural networks are constructed by introducing stochastic components into the NN by giving the networks stochastic activation (see Fig. 7b) or stochastic weights (see Fig. 7c).

As mentioned earlier, the objective of BNNs is primarily to get a better idea of the uncertainty of the model on its predictions. This is achieved by comparing the predictions of several possible parameterisations of the model. Following Jospin et al. [25], it can be summarized as follow

$$\theta \sim p(\theta)$$
$$y = NN_\theta(x) + \epsilon \tag{17}$$

where ϵ represents random noise to account for the fact that the function $NN(.)$ is just an approximation.

In order to design a BNN, it is necessary to follow the following steps

1. Choose a neural network architecture
2. Choose a prior distribution over the possible model parametrization $p(\theta)$
3. Choose a prior confidence in the predictive power of the model $p(y|x, \theta)$
4. Compute the posterior distribution $p(\theta|(X, Y))$ using bayesian inference

$$p(\theta|(X,Y)) = \frac{p(Y|X,\theta)p(\theta)}{\int_\Theta p(Y|X,\theta')p(\theta')d\theta'} \propto p(Y|X,\theta)p(\theta)$$

The difficulty of step (4), as is often the case in Bayesian inference, comes from the fact that the calculation of the term $\int_\Theta p(Y|X,\theta')p(\theta')d\theta'$ is often intractable. For this, two approaches can be used. Directly estimate the posterior distribution using a Markov Chain Monte Carlo (MCMC) algorithm or use a variational inference approach, which learns a variational distribution to approximate the exact posterior.

Once the posterior is approximated, it becomes possible to calculate for an input x a marginal probability distribution of the output y, which will model the uncertainty on the latter

$$p(y|x, D) = \int_\theta p(y|x,\theta')p(\theta'|(X,Y))d\theta'$$

In practice $p(y|x, D)$ is calculated using eq. 17. Thus the prediction of the model for a given x^* will be

$$\mathrm{E}[f(x^*)] = \frac{1}{|\Theta|}\sum_{\theta_i \in \Theta} NN_{\theta_i}(x^*)$$

And its uncertainty about this prediction will be

$$\mathrm{Var}[f(x^*)] = \frac{1}{|\Theta| - 1}\sum_{\theta_i \in \Theta}(NN_{\theta_i}(x^*) - \hat{y})(NN_{\theta_i}(x^*) - \hat{y})^T$$

In this work we will use https://github.com/Harry24k/bayesian-neural-network-pytorch library to create BNN models.

4.4 CMIP - Coupled Model Intercomparison Project

The Coupled Model Intercomparison Project seeks to better understand past, present and future climate changes by studying different types of General Circulation Models (GCMs). They particularly investigate on Coupled GCMs (like coupled ocean-atmosphere GCMs). In this kind of experiment, we usually have a model (e.g. an atmospheric model) and observations on the environment of this model (e.g. observations of the state of the ocean) which will act as a forcing. In this section, we are investigating to what extent the model calibration by History Matching is applicable to this kind of experiment.

As previously explained (section 4.2.2), the fast variable (Y) of the two-layers Lorenz-96 can be considered as an approximation of an Atmospheric General Circulation Model (AGCM) and the slow variable (X) as an Oceanic General Circulation Model (OGCM). For this reason, we can consider the Lorenz-96 as a set of two independent models that we can try to parameterize independently. We will discuss the methodology employed for this purpose in the two next subsections.

4.4.1 AMIP style experiments

In this section, we will investigate learning about parameters from the fast dynamics alone.

As stated by the World Climate Research Programme (WCRP), an AMIP experiment is an Atmospheric General Circulation Model constrained by a realistic sea surface temperature and sea ice.

In order to get as close as possible to the experimental conditions of an AMIP, we must first generate observations of the ocean which we will then use to force our atmospheric model (the fast component). We will use the history of the slow component of the model launched with the ground truth parameters for this purpose. Since we are only interested in the parameterization of the fast component, our model is therefore the following

$$\frac{1}{c}\frac{dY_{j,k}}{dt} = -bY_{j+1,k}(Y_{j+2,k} - Y_{j-1,k}) - Y_{j,k} + \frac{h}{J}\underset{obs_k}{\mathrm{X}}$$

Where $\underset{obs}{X}$ is the current state of the slow component observation register earlier. We therefore have three parameters to calibrate : h, c and b. A complete description of the algorithm is available in the appendix (see 0).

In their experiments, Schneider et al. [6] stated that the one-point statistics $(\bar{Y}_1, Y_1^2)$ of the fast variables are not enough to recover our three parameters and they therefore consider the moment function

$$\boldsymbol{f}_k(Y) = \begin{pmatrix} Y_{j,1} \\ Y_{j,1}Y_{j',1} \end{pmatrix}, \forall j, j' \in \{1, ..., J\} \tag{18}$$

Because the reasons are not explicitly detailed in their paper and because we use a different parameter search methodology, we will investigate to what extent this is the case.

4.4.2 OMIP - Ocean Model Intercomparison Project

In this section, we will investigate learning about parameters from the slow dynamics alone.

In parallel to an AMIP experiment, we refer to an OMIP experiment for an Oceanic General Circulation Model constrained by non-interactive atmospheric conditions. The rationale of this type of experiment is to focus on simulating ocean properties, free of structural or parametric biases coming from the atmospheric model.

We are generating the observations in the same way that for an AMIP experiment but instead of generating oceanic observations we are generating atmospheric observations by saving the fast component history (Y) run with the ground truth parameters. We will then force the the oceanic model describe by the X partial derivative equation

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - hc \underset{obs_k}{\bar{\mathrm{Y}}}$$

The metrics used in this case will be $f(X) = (X, X^2)^T$.

4.5 Dimensionality reduction of the metrics space

In this section, we consider reducing the dimensionality of the metrics using two types of methods, empirical orthogonal functions (EOFs), also known as principal component analysis (PCA) based on singular value decomposition (SVD), and then neural network based autoencoders.

4.5.1 Interest

Training a large number of emulators can be very time consuming. Gaussian process regressors have a time complexity of $o(n^3)$ for their learning phase and when the number of metrics increases, it can become quite complicated to train p emulators. Moreover, the information in the chosen parameters can be redundant (see Fig. 20) and it may therefore seem useful to try to reduce the dimension of the metrics.

4.5.2 Empirical Orthogonal Functions

Empirical Orthogonal Functions is a well studied dimensionality reduction procedure in the climate sciences community. The idea of this method is to project the variables into a lower dimensional space by seeking to minimise the correlation between the different dimensions.

We are therefore looking for a linear combination of the columns maximising the variance, we will note $Ya = \sum_{i=1}^{p} a_i y_i$. Its variance is given by $\mathrm{Var}[Ya] = a^T C a$ where C is the covariance matrix of Y. For this problem to have a well-defined solution, an additional restriction must be imposed and the most common restriction

involves working with unit-norm vectors, i.e. requiring $a^T a = 1$. Thus the problem can be posed as

$$\max_{s.c._a} a^T C a \tag{19}$$

$$a^T a = 1 \tag{20}$$

or by its Lagrangian relaxation $\max_a a^T C a - \lambda(a^T a - 1)$ where λ is a Lagrange multiplier. By deriving with respect to a we then obtain the maximum in $Ca - \lambda a = 0 \Leftrightarrow Ca = \lambda a$ thus represents an eigenvector (of unit norm) and λ is the associated eigenvalue.

By classifying the eigenvalues (and their associated eigenvectors) by order of magnitude $\{a_1, a_2, ..., a_p\}$, it will then be possible to reconstruct the space Y into a space Y' of dimension $n \times p'$ with $p' < \min(n, p)$ as follows

$$Y' = (Ya_1, Ya_2, ..., Ya_{p'})$$

The variance explained by each of the dimensions corresponds to the eigenvalue associated with the vectors. Thus the ith dimension explains λ_i of the variance.

In this work we will use the library *scikit-learn* to perform the PCA. When this is not specified, we will use the number of dimensions that explain 99% of the variance, i.e. we will choose p' in such a way that $\sum_{i=1}^{p'} \lambda_i \geq 0.99$.

4.5.3 Autoencoder

Autoencoders are non-linear dimensionality reduction models based on neural networks. The idea is to train a neural network to predict its inputs while passing through a layer where the number of neurons is lower than the number of neurons in the inputs. This layer is called the bottleneck layer. In order to keep the methodology as simple and reproducible as possible, we are interested here in single-layer autoencoders of the form

$$\begin{aligned} l_{in} &= y \\ l_{bottleneck} &= \sigma(W l_{in} + b) \\ l_{out} &= y \end{aligned} \tag{21}$$

We will take as activation function the $\sigma = tanh$ function and as loss function the mean squared error (mse). The number of neurons in the central layer will always be specified. The training of the model will be done with the library *keras* using backpropagation.

To transform the metrics of a sample we use the encoder trainer, i.e. $y' = \sigma(Wy + b)$.

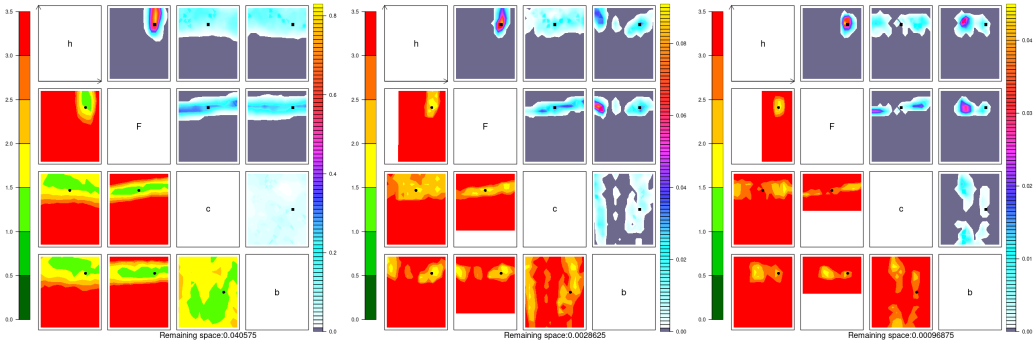

Figure 8: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS**. Wave 1 (left), wave 2 (center), wave 3 (right).

5 Experimental results

We present here the results that seem to us to be the most interesting and/or the most promising for the continuation of this research. We start by describing different results that allow us to evaluate History Matching method as a whole according to several variations in the methodology, being the choice of the sampling method of the parameter space, the number of samples in each wave or the distribution of the total number of samples per wave. We then focus on the choice of metrics for the Lorenz-96 search of parameters. We show, first, that AMIP or OMIP style experiments permit to History Matching to tune models that are forced by observations. In a second step, we show that it is possible to significantly reduce the parameter space, thus reducing the computational cost of training and predicting the emulator, by using dimensionality reduction methods. Finally, we perform comparison of several ML-based emulators, Linear regression, Random Forest and Gaussian Process, and which show their relevance in the context of History Matching.

5.1 Exploratory approach

First, it is important to present the results of History Matching in the most usual framework, i.e. with a maximin LHS sampling with $10 \times p$ samples at each wave and using a Gaussian Process Regressor as emulator.

It seems that in this framework HM converges well to the ground truth parameters (see Fig. 8). We see that even after one wave the parameter space has been reduced by about a factor of 25 and that after three waves less than 0.1% of the parameter space remains.

We can now use this benchmark result to assess the impact of our modifications to this classical approach.

5.1.1 Non-iterative History Matching

We now want to know if the iterative approach is of interest for History Matching. We therefore place ourselves in a non-iterative framework (only one wave is performed) but we use the same total number of samples as for the classical iterative

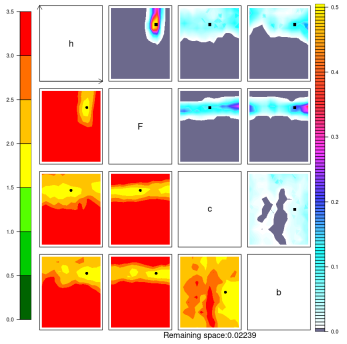

Figure 9: First wave of HM performed with **GPR emulator** on **120 samples** sampled with **maximin LHS**

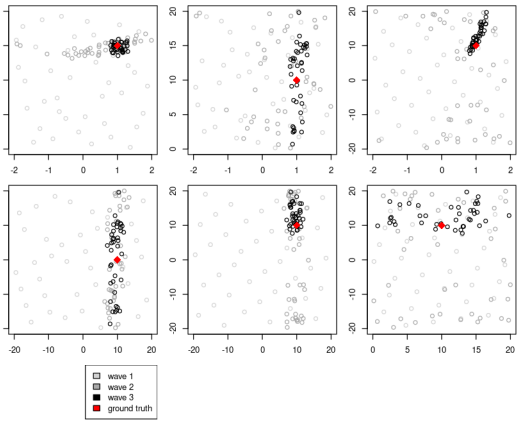

Figure 10: Inputs (generated by LHS sampling) for three waves of History Matching with 40 samples for each wave using GPR

approach, i.e. $3 \times 40 = 120$ samples.

It is not surprising that the non-iterative approach (see Fig. 9) gives better results than the first wave of the iterative approach (see Fig. 8) because the number of samples is larger than the emulators and therefore the uncertainty on these predictions is smaller. On the other hand, we can see that even with only two waves (and thus 40 fewer samples than for the non-iterative approach) the iterative approach shows better results than the non-iterative approach. This can be explained by the fact that with each wave the samples become more and more concentrated around the ground truth parameters (voir Fig. 10) and the emulator is therefore particularly accurate in its predictions in this area (the uncertainty on its predictions is smaller).

The iterative approach thus seems to be more efficient for tuning the Lorenz-96. It also has the advantage of being less expensive in terms of computation time for a Gaussian Process Regressor emulator because its training becomes expensive (as $O(N^3)$ with N the number of samples) when the number of samples is large. We think that it could be interesting to focus on the distribution of the number of

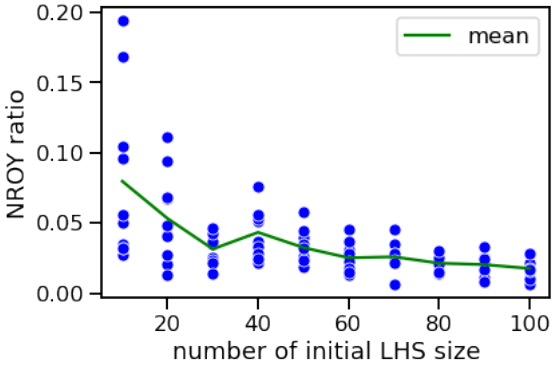

Figure 11: NROY ratio after first wave of HM regarding the number of inital samples in parameters space using LHS sampling methods.

samples for future work, i.e. to know how to distribute the number of samples on each wave given a fixed number of samples. For example, whether it is better to start with a small number of samples and increase it at each wave (e.g. $(N_1 = 20, N_2 = 40, N_3 = 60)$ for 120 samples) or to do the opposite, i.e. to start with a large number of samples and reduce it at each wave (e.g. $(N_1 = 60, N_2 = 40, N_3 = 20)$ for 120 samples)

5.1.2 Sampling methodology

While we have previously raised the issue of the distribution of the number of samples, we propose to evaluate the impact of the number of samples on the results of History Matching in a more general way.

To do so, we evaluate the evolution of the NROY space after a wave of HM as a function of the initial number of samples. For that we carry out 10 iterations of HM for a number of samples of $N_i = i \times 10, \forall i \in \{1, ..., 10\}$. We thus obtain figure 11 on which we can see that the greater the number of samples, the greater the NROY space after History Matching. In particular, we see that the variability of the NROY ratio decreases as the number of samples increases.

It seems to us in view of this result that the $N = 10 \times p$ rule is relevant because it is a good compromise between calculation time, performance and variability of results. We will therefore use $10 \times p$ samples, i.e. 40 samples to calibrate the Lorenz-96 in a classical setting and 30 in an AMIP or OMIP-style experiment.

As described in Subsect. 4.1.1, the method commonly used to sample the parameter space when tuning by HM is maximin sampling. We propose here to compare it to two other methodologies, random sampling and Quasi-Monte Carlo sampling with Sobol sequence.

For this purpose we propose to compare the result of the third wave of HM for the three compared methods.

First of all, it seems that all three methods of scaling reduce the parameter space by the same amount (i.e. about 0.1% of the parameter space remains after three waves). However, the distribution of the rejection area is different. First of all, even if the NROY space obtained by HM with random sampling covers well the ground truth parameters, it did not allow the HM to converge to the parameter space since

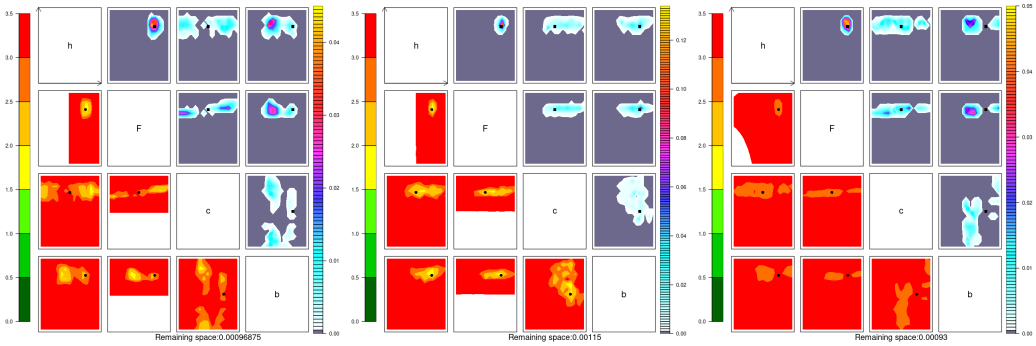

Figure 12: History Matching performed with GPR for 40 samples per waves sampled with **maximin LHS sampling** (left), **QMC Sobol sampling** (center), **random sampling** (right).

the $c(b)$ area of the ground truth parameters was rejected as shown in Fig. 12 (right). Moreover, it can be noted that the NROY space obtained by HM with random sampling is closer to rejection than the other two (with an implausibility between 2.5 and 3 which makes its results less consistent. On the other hand, the two maximal sampling methods LHS and QMC with Sobol sequence allow to obtain very similar NROYs spaces after three waves. They reduce it by the same order of magnitude and do not reject the ground truth parameters. It should be noted that in general the NROY space obtained by HM with QMC sampling with Sobol sequence is more concentrated around the ground truth parameters (see Fig. 12 (left) and (centre)) which makes its results more consistent.

It seems that the Quasi-Monte Carlo sampling method with Sobol sequence is as good as, or better than, the LHS maximin for sampling the parameter space. This can be explained by the fact that this method minimises the correlation between the different samples in the parameter space which can, for certain types of function, allow better modelling. In addition, this method has the advantage of being particularly fast and therefore has an advantage when the parameter space is very small and a large number of samples are needed in the initial space to ensure that there are enough samples left in the NROY space.

There is an LHS-type sampling method that also minimises the correlation between the samples in the parameter space, and we believe that it may be interesting to evaluate the results of this method in future work.

Even if the sampling by QMC method with Sobol sequence seems to show interesting results we prefer to restrict ourselves to the sampling by LHS maximin for the remainder of this work so that our results can be compared with those obtained in a classical approach (Fig. 8).

5.2 Metrics space and MIP-style experiment

We have thus shown that History Matching can significantly reduce (by an order of 1000) the parameter space in a classical framework. We now wish to determine whether all the metrics are necessary for this. Indeed the computational cost of training and predicting the emulator is linear in the number of metrics, so it could be interesting to reduce it significantly.

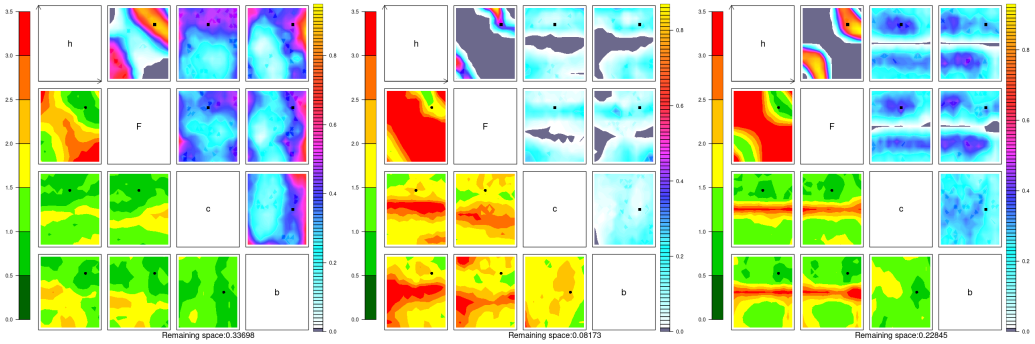

Figure 13: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS** only using fast component metrics $(\bar{Y}, \bar{Y^2})$. Wave 1 (left), wave 2 (center), wave 3 (right).

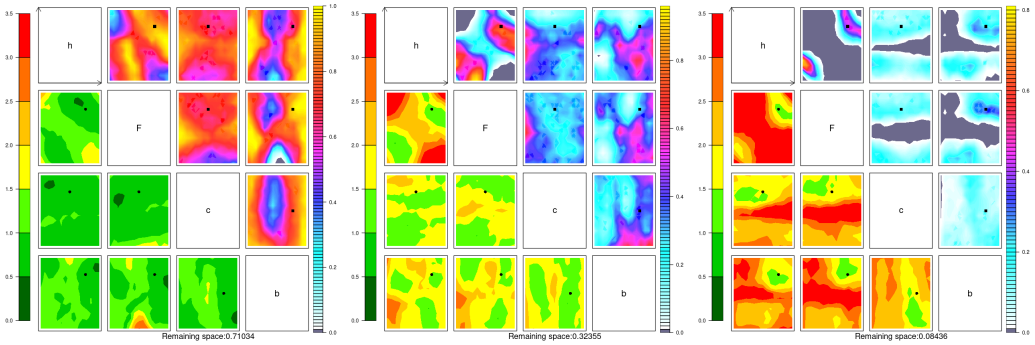

Figure 14: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS** only using Eq. [18] metrics. Wave 1 (left), wave 2 (center), wave 3 (right).

5.2.1 Metrics selection

First, we propose to determine whether metrics concerning only one component (either the slow one, i.e. X, or the fast one, i.e. Y) can significantly reduce the parameter space. We justify this approach by the interest it would generate if it allowed us to obtain good results. Indeed, by making the analogy between Lorenz-96 and an AOGCM model, this would mean that it would be possible to calibrate an AOGCM only with atmospheric or oceanic observations, which would be of interest.

To do this, we will first test the capacity of the model using only observations of the fast component, first with only the metrics $\bar{Y}$ and $\bar{Y^2}$ and then with the metrics proposed by Schneider et al. [6] to describe the fast component (see Eq. [18])

It seems that the only metrics $\bar{Y}$ and $\bar{Y^2}$ do not cause the History Matching to converge. We can indeed see (Fig. 13) in the third wave that the space of excluded parameters is smaller than that of the second wave.

On the other hand, the metrics described by Eq. 18 seem to converge slowly to ground truth parameters.

We now want to know whether the metrics describing the slow component alone can significantly reduce the parameter space. We therefore perform an experiment using exclusively the metrics $f(X) = (X, X^2)^T$.

We can see (Fig. 15) that the use of metrics describing the fast components allow

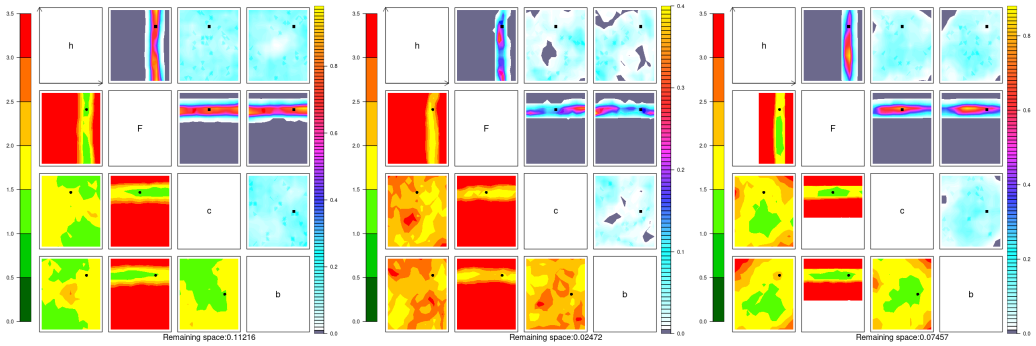

Figure 15: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS** only using $f(X) = (X, X^2)^T$ metrics. Wave 1 (left), wave 2 (center), wave 3 (right).

after a wave to significantly reduce the parameter space (by a factor of about 10) and thus seem to be much more informative about the state of the system than the metrics describing the fast component. However, these do not seem to be sufficient to converge towards the ground truth parameters as is the case when all the metrics are used (see 8).

In general, it does not seem possible to calibrate the model using only the metrics describing the behaviour of either the fast or slow component. Even if the parameter space can be reduced, HM does not converge and only the parameter F seems to be found from the metrics X and X^2 which is consistent with the form of the equations describing the model since F has a direct impact on the slow component.

5.2.2 AMIP- and OMIP-style experiments

While it seems that the observation of a single component does not allow the calibration of the whole model in a classical framework, it seems interesting to us to determine whether this is also the case in the framework of MIP-style experiment. That is to say, to know whether observing a single component allows us to calibrate a model comprising only this component and being forced by the second. The protocol of these experiments is described in the Subsect. 4.4. We recall here that this type of experimentation is particularly important for the climate study community since AMIP is part of the CMIP6 Deck and OMIP could be included if a new version of CMIP is made.

We therefore start with an AMIP-style experiment. We seek to calibrate the fast component (thus to find the parameters h, b and c) by forcing it with observations of the slow component.

Firstly, we can note that History Matching seems to be well applicable in the context of an AMIP-style experiment. The method converges relatively quickly to the ground truth parameters (see Fig. 17 and 16) and thus allows to significantly reduce the parameter space (by an order of 100 to 1000 depending on the metrics used) after a few waves.

The metrics proposed by Schneider et al. [6], namely Eq. [18], seem to allow the History Matching parametrization to converge faster towards the ground truth parameters than the metrics $f(Y) = (\bar{Y}, \bar{Y^2})^T$ even if the latter also allow the HM

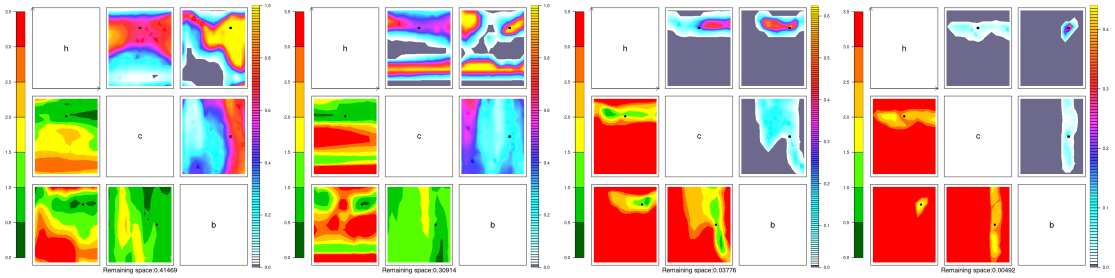

Figure 16: First four waves of History Matching performed in an AMIP style experiment with $(\bar{Y}, \bar{Y^2})^T$ metrics.

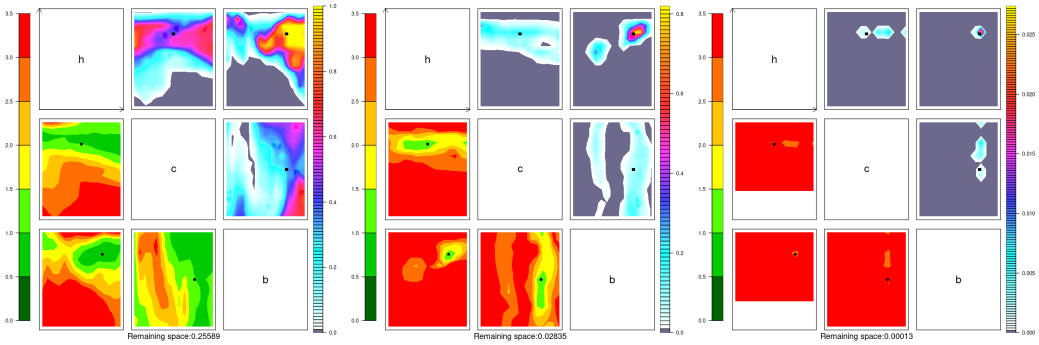

Figure 17: First three waves of History Matching performed in an AMIP style experiment with Eq. [18] metrics

to converge. The first metrics reduce the parameter space by 0.99987 after three waves while the second metrics reduce it by 0.99508 after four waves. Moreover, it seems important to note that the two approaches differ strongly in their logic in that the metrics described by Eq. [18] describe in a very precise but also very localised way a part of the state of the system whereas the metrics $f(Y) = (\bar{Y}, \bar{Y^2})^T$ give a global description of the whole system.

We now place ourselves in the context of an OMIP-style experiment, i.e. we seek to calibrate the slow component (parameters h, F and c) by forcing it with observations of the fast component (see 4.4.2 for more details). As a reminder, this experiment tries to come close to the parameterisation of an oceanographic model that would be forced with observations of the atmosphere.

As one might have expected, the parameterisation of this type of dynamic model does not pose a problem. Indeed, in the framework of the parameterisation of OMIP-style experiments, we only have three parameters to tune, namely h, F and c, which excludes the parameter b which, as we have seen in the previous experiments, is the most difficult parameter to tune. We can thus see (Fig. 18) that after one wave of HM we have reduced the parameter space by 0.97971 and by 0.98032 after three waves.

In general, HM seems to be suitable for OMIP and AMIP-style experiments. We have seen that it is able to significantly reduce the parameter space for both the slow and fast components. It thus seems possible, by considering two independent models, to overlap parameters that will allow a coherent system.

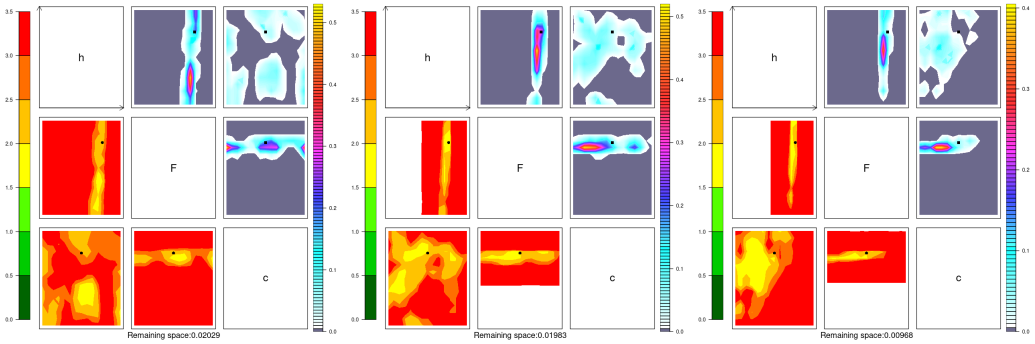

Figure 18: First three waves of History Matching performed in an OMIP style experiment with $f(X) = (X, X^2)^T$ metrics

5.2.3 Dimensionality Reduction of metrics space

In the section 5.2.1 we saw that it was not possible to select only the metrics of the slow component or the fast component to calibrate the model as a whole (the two coupled components). We will show in this section that it is possible to significantly reduce the number of metrics by dimensionality reduction methods and thus reduce the computational cost of training and predicting the emulator.

The reduction of the dimensionality of the space of metrics could, if it is applicable in the framework of History Matching, prove to be particularly interesting by allowing to significantly reduce the training time of prediction of the emulator used. We will observe the results obtained for two dimensionality reduction methodologies, a linear, knowing the Empirical Orthogonal Functions (or Principal Component Analysis) and a non-linear, knowing Autoencoders. We will mainly use two criteria to evaluate these methods, the mean square error of the recontrusion of a set of validation metrics and the proportion of the NROY space remaining at each wave.

- **Empirical Orthogonal Functions**

We test here the application of the EOF for the reduction of dimmensionality of metrics space.

We can see (Tab. 2) that the NROY space reduces significantly after two waves and mainly that it reduces by the same order of magnitude as during the HM without dimensionality reduction (see Fig. 8). However, the dimension of the metrics is considerably smaller here, since we have 10 dimensions compared to 180 for the HM without dimensionality reduction, so the computational cost is reduced by a factor of 18 for both the training and the prediction of the model.

Table 2: Mean Squared Errors for PCA with 10 components (explained variance ≥ 0.99)

wave	train MSE	val MSE	% of original space
1	0.0186	0.0571	0.053
2	0.0574	0.1010	0.01739
3	0.0321	0.0415	0.00191

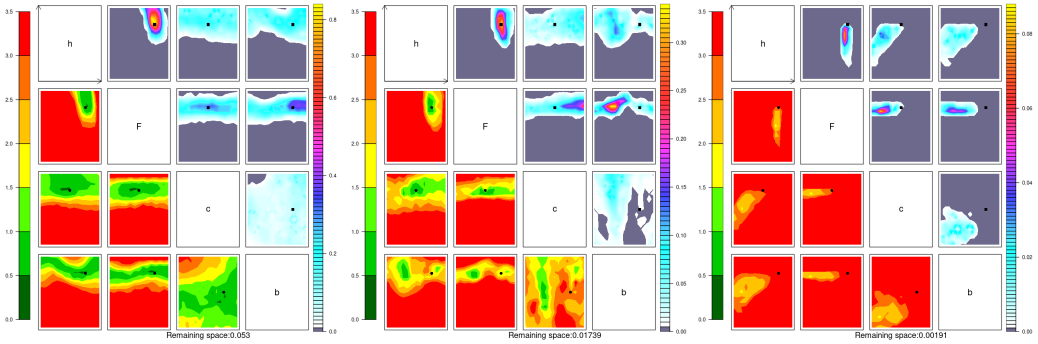

Figure 19: First three waves of History Matching performed with PCA (10 cp). First wave (left), second wave (center), third wave (right)

But as we can see in Fig. 19, at the third wave, HM rejects the ground truth parameters. This is certainly explained by the uncertainty not taken into account on these metrics (we have an MSE of about 4% on the reconstruction of the validation set at the third wave). It therefore seems necessary to take into account this uncertainty on our metrics which will then reduce the capacity of the method in its reduction of the parameter space.

- **Autoencoder**

Table 3: Mean Squared Errors for Autoencoder with 32 dimensions

wave	train MSE	val MSE	% of original space
1	0.004	0.1251	0.07115
2	0.0045	0.2238	0.02693
3	0.0057	0.1369	0.09456

We notice that the mean square error of the validation dataset reconstruction is much larger with Autoencoders than with a PCA. We also notice that the mean square error on the training set is of the same order of magnitude for the Autoencoders and for the PCA, it is the mean square error of the validation set that is much larger for the Autoencoders, probably due to the larger number of parameters compared to the size of the data set resulting in some overfitting from the Autoencoder. Thus we can see that the HM does not converge to the ground truth parameters, the uncertainty on the metrics being too large.

Several methods have been considered to solve this problem, including the use of a regularisation term during training (norm L_1, L_2 or dropout) but this brings an additional concept into play and we prefer to restrict ourselves to the following method. Rather than training the Autoencoder only on the simulated data for each wave, we will train it on all the generated data, i.e. for wave n we will use the data $((X_1, Y_1), ...(X_n, Y_n))$ where (X_i, Y_i) corresponds to the data simulated during wave i.

Table 4: Mean Squared Errors for Autoencoder with 32 dimensions with additive approach

wave	train MSE	val MSE	% of original space
1	0.0043	0.1255	0.0510
2	0.0099	0.1065	0.03755
3	0.0132	0.0647	0.02638

We thus obtain better results, both for the convergence of the HM to the ground truth parameters and for the mean square error, whose value on the training set is now closer than that of the validation set, which implies a less important overfitting. Thus the calibration by History Matching with dimension reduction by Autoencoder to 12 dimensions made it possible to reduce the space of the parameters by 0.97362 while preserving the ground truth parameters which seems interesting to us since it also makes it possible to reduce the time of training and prediction by more than 5 (with 32 dimensions)

We therefore consider that Autoencoders could be particularly interesting when the space of metrics is high dimensional in order to reduce the learning and prediction time of the emulator. We believe that these will perform best in the case where the number of parameters to be tuned is large and the simulated dataset is therefore also relatively large and the metrics have particularly non-linear structures. It would then be interesting to study the application of Autoencoders to more complex structures and more adapted to the data.

5.3 Emulators from the machine learning community

Finally, we propose in this section to evaluate if the two proposed emulators Subsect. 4.3 are efficient in the context of the calibration of Lorenz-96 by History Matching. These results are not definitive and it is difficult to say whether these emulators can be used in the general context. However, we believe that this part of the work has given rise to sufficiently interesting reflections for these results to be presented.

The different emulators will be compared with regard to several criteria. Firstly, we will evaluate their mean square error, which will give us information on their overall modelling quality. We will then analyse their uncertainties to see how they impact History Matching. Finally we will have a more qualitative approach by observing the NROY space generated by History Matching with these different emulators.

5.3.1 First results

As described in the section 4.3, emulators are usually described as a linear regression model on which a complementary model can be added to model the residuals, for example this is what is described in the section "Gaussian Process Regressor" for modelling the residuals. We therefore use this approach in this section and present the results obtained for each wave of History Matching with linear regression to see if the models model the residuals well. We note in passing that the results of Bayesian Neural Networks are not included in this section as they are not yet consistent and we propose to study them in future work.

First, we perform a first wave of History Matching on a set of 40 samples in the initial parameter space. We will test the Mean Squared Error on a validation set also containing 40 samples which will allow us to have a first view of the modelling quality of the considered emulator. Then we test the mean uncertainty for each sample. The idea is that the implausibility will strongly depend on the uncertainty and that a too large uncertainty will not allow to reject a sufficient part of the space of the parameters. Finally we observe the ratio of original space in the NROY space.

Table 5: Mean Squared Errors on test set for different emulators on initial parameters space

heightemulators	val MSE	mean uncertainty	% of original space
Linear Regression	0.4071	0.1832	0.0992
Gaussian Process	0.2734	0.1420	0.0744
Random Forest	0.2730	0.2898	0.0226

We can see (see Tab. 5) that Random Forest has as good an overall predictive quality as Gaussian Process (it has the same MSE) and that these two emulators allow a significant gain in predictive quality compared to a simple linear regression. On the other hand, we note that the uncertainty on the predictions of the Random Forest is much larger than that of the other two emulators, which can be problematic since it will not allow sufficient rejection of the parameter space since the uncertainty on the predictions will often be too large. This has an impact on the NROY ratio as we see that Random Forest rejects a smaller ration of the parameter space than linear regression despite the fact that it predicts the data much better. We believe that this may be due to the methodology for estimating the variance on the predictions which may not be sufficiently efficient. This could be due to the fact that the already low number of samples does not allow for a good assessment of implausibility because too few data are used to calculate this uncertainty measure.

Table 6: Mean Squared Errors on test set for different emulators on second wave parameters space

heightemulators	val MSE	mean uncertainty	% of original space
Linear Regression	1.6768	0.1153	0.0502
Gaussian Process	1.7004	0.0216	0.0021
Random Forest	1.6821	0.1410	0.0588

We observe that after a second wave of History Matching (see Tab. 6), the predictive qualities of the emulators become similar but that on the other hand there is now a large gap between the uncertainty estimated by the Gaussian Process and that estimated by the Random Forest or the Linear Regression. The Random Forest NROY space was significantly reduced in this wave (from 0.0992 to 0.0588) but this is not sufficient to approach the performance of the Gaussian Process in a single wave of History Matching.

We do not continue History Matching with additional waves because the Gaussian Processes have sufficiently reduced the parameter space after 2 waves but the

Random Forest and Linear Regression emulators have converged to the minimum NROY space they allow. Indeed after a third wave the NROY ratio of these two emulators will not decrease.

Overall, it seems that Random Forest has a good predictive quality on the data, but the method used in this work to estimate their uncertainty on the predictions (the Out Of Box error) does not seem to be sufficiently efficient for Random Forest to be of real interest in this context to replace Gaussian Processes.

6 Discussion

6.1 Future work

Various avenues of work have been explored in this report, both in the History Matching methodology itself and in the statistical models used as emulators. We believe that several avenues deserve to be explored.

First of all, in the Subsect. 5.1.2 we raised the possibility of exploring different distributions of the number of samples on the whole waves. Indeed, even if it seems that an iterative approach is more efficient than a non-iterative one, it is not clear how to distribute the number of samples for each wave of History Matching, i.e. if it is more efficient to start with a large number of samples and then to reduce it at each wave or to do the opposite. Then, still for the Space Filling Design stage, we propose to explore the possibility of using the LHS optimised snail method with a correlation criterion, which, in view of the good results of the LHS maximin and the QMC sampling with Sobol sequence, could be efficient. Moreover, we were not able to reach strong conclusions regarding the best ML-based emulator that could replace GPs already used extensively in the literature, each emulator has its own advantages and difficulties. For example, Random Forests models fit our data correctly (with a mean squarred error similar to that of GPs on a test dataset), but it is time-consuming and complex to measure the uncertainties on their predictions. Different methodologies exist to evaluate these uncertainties, such as the Quantiles Regression Forest, and it could be interesting to explore this avenue to improve the estimation of the model's uncertainties and thus obtain better performances for History Matching. We also think that Bayesian Neural Networks can be a viable candidate for futur experiments. Finally, we remind the reader that the experiments have been carried out on a toy model, the Lorenz-96, it seems important to us to see if our results are confirmed on real coupled ocean-atmosphere climate models, and mainly for the dimensionality reduction, which in the framework of the Lorenz-96 calibration has shown interesting results.

6.2 Environemental and societal impact

Scientific knowledge of the processes at work in climate change now makes it very likely that the increase in the Earth's average temperature over the last century is a human consequence, mainly due to our greenhouse gas emissions. The Intergovernmental Panel for Climate Change (IPCC) regularly publishes a series of reports describing the state of scientific knowledge about climate mechanisms, projections

of future climate change under different scenarios that are mostly dependent on human factors, and recommendations on measures to reduce human impact on climate change or to adapt to inevitable climate change. The increase in temperature is not evenly distributed and some geographical areas are already and will be more impacted than others by the evolution of the Earth's climate (melting ice at the North Pole and extreme precipitation in East Asia for example). Furthermore, with the increase in the Earth's average temperature comes an increase in climatic variability, which in turn increases the probability of the occurrence of extreme climatic events such as droughts, heavy precipitation or extreme heat. These changes are profoundly transforming our environment more rapidly than ever before, impacting not only human populations but also the biodiversity that is essential to their survival. In this context, it seems reasonable to think that human societies will experience major upheavals in the years to come due to restricted access to water, food or extreme climatic events making certain regions uninhabitable. The work done by the scientific community on climate and climate change is therefore a valuable tool to push societies to transform and act to prevent climate change from making our planet unlivable and to prepare for events that may occur in the near future. The work in this field of research is very varied and touches on a large number of phenomena and scales in order to have the most precise vision of the transformations that we are going to undergo and to be able to put in place the necessary tools to prevent them as well as possible.

The work carried out during this internship is part of a general, albeit very small-scale, scientific collaboration to understand the climate system. It is always difficult in a field of this magnitude to assess individual contributions. It is possible that some methodologies are never applied in practice or that some discoveries are ultimately of little value in understanding climate phenomena. But even these misunderstandings seem necessary, and any path that seems promising should be explored in order to maximise our chances of discovering the most effective and relevant methods.

Finally, it seems important to question the real impact of the evolution of scientific knowledge on the understanding of the mechanisms governing the climate system. On a global scale, greenhouse gas emissions are not decreasing and have been growing linearly since the 1950s, seemingly unaffected by the IPCC reports or the various COPs. Even if the environmental issue is now central to public debate, it does not seem that decision-makers at the global level have assessed the extent of the changes that climate change could imply for our societies. The scientific tools developed by the climate search community seem to be one step ahead of political decisions for the time being.

7 Conclusion

In this work, we have been able to conduct several experiments and reach a set of results that we consider interesting for the calibration of climate models by History Matching. First of all, we have been able to show that the History Matching method allows the calibration of the two-layer Lorenz-96 numerical model with relatively few samples and therefore little numerical simulation. As Lorenz-96 can be considered as a simplified version of a coupled ocean-atmosphere model, this result is promising

for the calibration of coupled climate models in general, even if it seems important to confirm these results in experiments on real climate models.

Furthermore, we have shown that while the *maximin LHS* sampling method commonly used for History Matching has shown good results when calibrating the Lorenz-96 by History Matching, it seems that other techniques such as Quasi-Monte Carlo sampling method with Sobol sequence, can also be used with the same perfomance or even better sometimes, with a reduced computational cost and minimising the correlation between samples in the parameter space.

We have also seen that the History Matching calibration method can also be used to tune the Lorenz-96 in AMIP or OMIP experiments.

A particularly promising result concerning the calibration of climate models by History Matching is the reduction of the dimensionality of the metric space. We have indeed shown that it is possible to significantly reduce the metric space (by a factor of 18 for EOF) using the Empirical Orthogonal Functions and Autoencoders dimension reduction methods. While Empirical Orthogonal Functions are commonly used by the climate research community, Autoencoders are, to our knowledge, less so and we have seen in this work that they have several advantages over Empirical Orthogonal Functions. Being non-linear, they have the advantage of being able to model models with a strong non-linear component. Also they allow to reduce the dimensionality of the metrics of the desired order and are thus very flexible when the Empirical Orthogonal Functions are limited by the number of samples as well as the maximum number of dimensions once reduced. We therefore believe that it would be interesting to continue research on the application of Autoencoders for dimensionality reduction for the calibration of climate models, by applying them to a real climate model.

Finally, we explored the possibility of replacing the Gaussian Processes commonly used for History Matching by Random Forest. It seems difficult at this time to comment on their performance. However, we note that this model has a predictive quality equivalent to that of Gaussian Processes for the Lorenz-96 calibration data. The Out Of Box error measure of uncertainty seems to be ill-suited for History Matching because the number of samples is relatively small and the estimate of the variance on the predictions is relatively poor. We have therefore not yet been able to show that Random Forest has any real advantage over Gaussian Processes.

From a more individual point of view, my participation in this work within the LOCEAN-IPSL laboratory has allowed me to acquire different knowledge. Whether it is in the field of climatology, where I was able to discover the different problems linked to the calibration of climate models, or more generally on the approach of the climate study community to succeed in gathering a large number of results in a coherent way, but also in machine learning where I was able to deepen my knowledge of certain statistical models such as linear regression or random forest and to discover others such as gaussian process or bayesian neural networks. I was also confronted with the questions that arise for the calibration of numerical models, and for the resolution of inverse problems more generally, such as sampling methods. But, on a level that I am sure will bring me much more later on, this internship allowed me to see how difficult it can be to set up a working method that allows to obtain consistent and reproducible results. It has therefore allowed me to start putting in

place certain work habits, which even if they are not perfectly calibrated at this time, will allow me to work more efficiently later on. This internship also allowed me to realize the importance of communication and sharing in order to work in the best conditions. Indeed, as this one was carried out in remote working, I was able to realize the importance of human contact for intellectual stimulation.

I would like to thank once again my supervisors for making this internship possible and for the help they gave me during these six months of collaboration.

8 Appendix

8.1 Algorithms

Algorithm 1 AMIP style experiment with two layers Lorenz96 model

Require: p_T (the ground truth parameters), P (the set of tested parameters)
  $metrics \leftarrow ()$
  $l96_T \leftarrow L96(p_T)$
  $l96_T.iterate(10)$ ▷ Reach the attractor
  $l96_T.erase\_history()$ ▷ Erase history
  $l96_T.iterate(100)$
  $X\_hist \leftarrow l96_T.history\_X$ ▷ This is our **ocean observations**
  for $p \in P$ **do** ▷ Tested parameters
    $l96 \leftarrow L96(p)$
    $l96.iterate(10)$ ▷ Reach the attractor
    $l96.erase\_history()$ ▷ Erase history
    $l96.iterate(100)$
    $Y\_hist \leftarrow l96.history\_Y$ ▷ Store Y history
    $m \leftarrow compute\_metrics(Y\_hist)$
    $metrics \leftarrow (metrics, m)$
  end for
  Return($metrics$)

Here L96(.) is the Lorenz96 model, it has two functions, knowing $iterate(n)$ that ierate the model for n iterations and store the histories in $history\_X$ and $history\_Y$ and $erase\_history()$ that delete the previously stored histories. The $compute\_metrics()$ function compute the metrics described earlier.

8.2 Additional figure

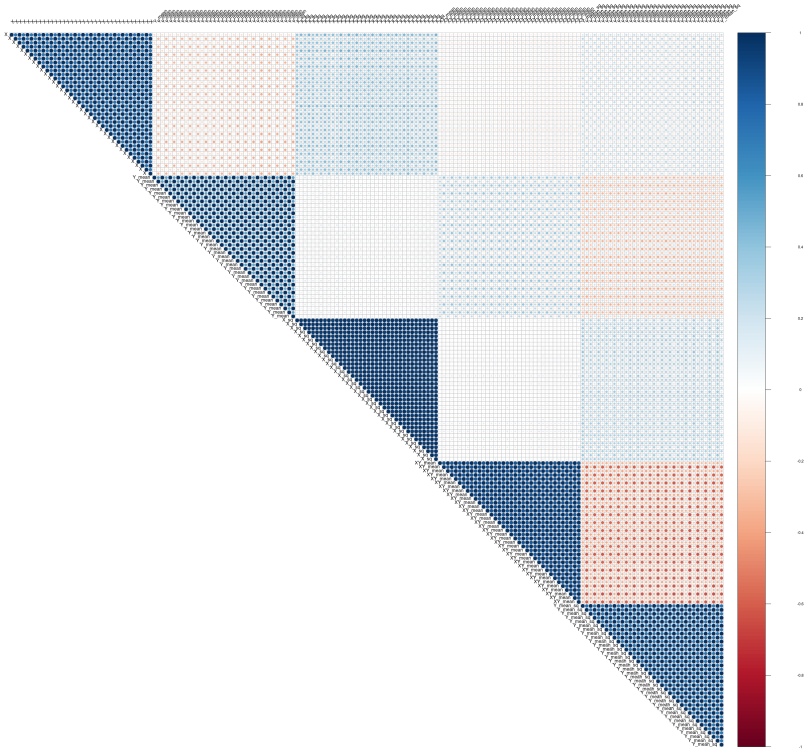

Figure 20: Correlation of metrics $(X, \bar{Y}, X^2, X\bar{Y}, \bar{Y^2})^T$ for 40 samples generated with LHS sampling method.

References

[1] George W Platzman. The ENIAC Computations of 1950 – Gateway to Numerical Weather Prediction. *Bulletin of the American Meteorological Society*, 60 (4):302–312, 1979.

[2] S. Manabe and K. Bryan. Climate calculations with a combined ocean-atmosphere model. *J. Atmos. Sci*, 26(4):786–789, 1969.

[3] Syukuro Manabe and Richard T Wetherald. The Effects of Doubling the CO2 Concentration on the climate of a General Circulation Model. *Journal of Atmospheric Sciences*, 32:3–15, 1975.

[4] James Salter and Daniel Williamson. A comparison of statistical emulation methodologies for multi-wave calibration of environmental models. *Environmetrics*, 27, 12 2016. doi: 10.1002/env.2405.

[5] Daniel Williamson, Adam Blaker, and Bablu Sinha. Tuning without overtuning: parametric uncertainty quantification for the nemo ocean model. *Geoscientific Model Development Discussions*, pages 1–41, 08 2016. doi: 10.5194/gmd-2016-185.

[6] Tapio Schneider, Shiwei Lan, Andrew Stuart, and João Teixeira. Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44, 08 2017. doi: 10.1002/2017GL076101.

[7] Peter S. Craig, Michael Goldstein, Allan H. Seheult, and James A. Smith. Pressure matching for hydrocarbon reservoirs: A case study in the use of bayes linear strategies for large computer experiments. In Constantine Gatsonis, James S. Hodges, Robert E. Kass, Robert McCulloch, Peter Rossi, and Nozer D. Singpurwalla, editors, *Case Studies in Bayesian Statistics*, pages 37–93, New York, NY, 1997. Springer New York. ISBN 978-1-4612-2290-3.

[8] Ian Vernon, Michael Goldstein, and Richard Bower. Galaxy formation: a bayesian uncertainty analysis. *Bayesian Analysis*, 5, 12 2010. doi: 10.1214/10-BA524.

[9] Ioannis Andrianakis, Ian Vernon, Nicky McCreesh, Trevelyan McKinley, Jeremy Oakley, Rebecca Nsubuga, Michael Goldstein, and Richard White. Bayesian history matching of complex infectious disease models using emulation: A tutorial and a case study on hiv in uganda. *PLoS Computational Biology*, 11, 01 2015. doi: 10.1371/journal.pcbi.1003968.

[10] Daniel Williamson, Michael Goldstein, Lesley Allison, Adam Blaker, Peter Challenor, Laura Jackson, and Kuniko Yamazaki. History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble. *Climate Dynamics*, 41:1703–1729, 10 2013. doi: 10.1007/s00382-013-1896-4.

[11] V. Joseph. Space-filling designs for computer experiments: A review. *Quality Engineering*, 28:28–35, 01 2016. doi: 10.1080/08982112.2015.1100447.

[12] Luc Pronzato and Werner Müller. Design of computer experiments: Space filling and beyond. *Statistics and Computing*, pages 1–21, 05 2011. doi: 10.1007/s11222-011-9242-3.

[13] M.E. Johnson, L.M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990. ISSN 0378-3758. doi: https://doi.org/10.1016/0378-3758(90)90122-B. URL https://www.sciencedirect.com/science/article/pii/037837589090122B.

[14] M. Mckay, Richard Beckman, and William Conover. A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 05 1979. doi: 10.1080/00401706.1979.10489755.

[15] Nan Chen and L. Hong. Monte carlo simulation in financial engineering. pages 919–931, 01 2008. ISBN 978-1-4244-1306-5. doi: 10.1109/WSC.2007.4419688.

[16] Sergei Kucherenko, Daniel Albrecht, and Andrea Saltelli. Exploring multi-dimensional spaces: a comparison of latin hypercube and quasi monte carlo sampling techniques. 05 2015.

[17] D. J. McNeall, P. G. Challenor, J. R. Gattiker, and E. J. Stone. The potential of an observational data set for calibration of a computationally expensive computer model. *Geoscientific Model Development*, 6(5):1715–1728, 2013. doi: 10.5194/gmd-6-1715-2013. URL https://gmd.copernicus.org/articles/6/1715/2013/.

[18] Redouane Lguensat, Pierre Tandeo, Pierre Ailliot, Manuel Pulido, and Ronan Fablet. The analog data assimilation. *Monthly Weather Review*, 145:4093–4107, 10 2017. doi: 10.1175/MWR-D-16-0441.1.

[19] Edward Ott, Brian Hunt, Istvan Szunyogh, Aleksey Zimin, Eric Kostelich, Matteo Corazza, Eugenia Kalnay, D. Patil, and James Yorke. A local ensemble kalman filter for atmospheric data assimilation. *Tellus*, 10 2004. doi: 10.1111/j.1600-0870.2004.00076.x.

[20] Edward Lorenz. Optimal sites for supplementary weather observations: Simulation with a small model. 06 2001.

[21] J.L. Anderson. An ensemble adjustment kalman filter for data assimilation. *Monthly Weather Review*, 129:2884–2903, 12 2001.

[22] David Gagne, Hannah Christensen, Aneesh Subramanian, and Adam Monahan. Machine learning for stochastic parameterization: Generative adversarial networks in the lorenz'96 model. *Journal of Advances in Modeling Earth Systems*, 12:e2019MS001896, 03 2020. doi: 10.1029/2019MS001896.

[23] Edward N. Lorenz. Predictability - a problem partly solved. *Cambridge University Press*, 1996.

[24] Stephan Rasp. Online learning as a way to tackle instabilities and biases in neural network parameterizations, 07 2019.

[25] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. Hands-on bayesian neural networks – a tutorial for deep learning users, 2020.

[26] L Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001. doi: 10.1023/A:1010950718922.

[27] Haozhe Zhang, Joshua Zimmerman, Dan Nettleton, and Daniel Nordman. Random forest prediction intervals. *The American Statistician*, 74:1–20, 04 2019. doi: 10.1080/00031305.2019.1585288.