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. Finally, we stress the importance of the availability of tools allowing to estimate precisely and simply the uncertainty of the predictions of the different models.

Contents

1 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 AOGCMs with History Matching.

2 Scope

The first phase of this work consisted in using the History Matching tool and investigating its relevance for AOGCMs by applying to the Lorenz-96 toy model. We then start this report by presenting the History Matching technique in Subsect. 3.1 then introducing the Lorenz-96 model in Subsect. 3.2. We will 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 is given in the Subsect. 3.4.

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 3.3) is cubic which makes them hardly usable 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. 3.3 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. 3.5

3 Methodology

3.1 History Matching

The term History Matching first appeared (Craig et al. [7]) in the oil engineering community. 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, these 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 3.1.1. The simulation of the PPE and the choice of metrics to represent it is

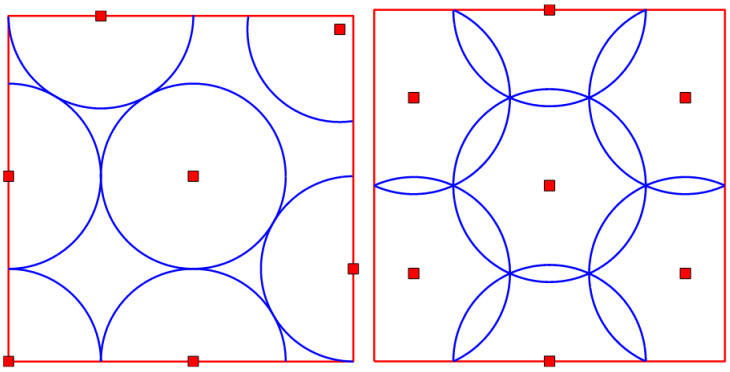

Figure 1: Maximin (left) and minimax (right) designs for 7 points in $[0,1]^2$. From Pronzato and Müller [12].

explained in section 3.1.2. 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 3.1.3. We describe in section 3.1.4 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 3.1.5.

3.1.1 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

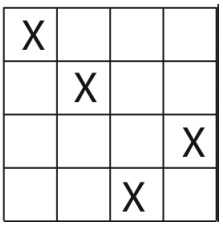

Figure 2: Latin Hypercube Sampling in 2 dimensions with 4 points.
Source : Wikipedia

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

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.

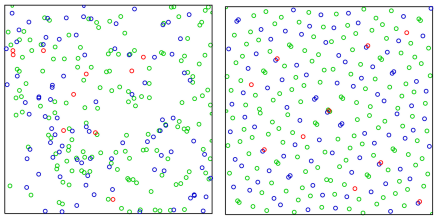

Figure 3: Monte Carlo Sampling (left) and Quasi Monte Carlo Sampling (right) for red=1,..,10, blue=11,..,100, green=101,..,256.
Source : Wikipedia

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]).

3.1.2 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 3.1.4) 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]).

3.1.3 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 3.3.

3.1.4 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.

3.1.5 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].

3.2 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 3.2.2.

3.2.1 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] 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].

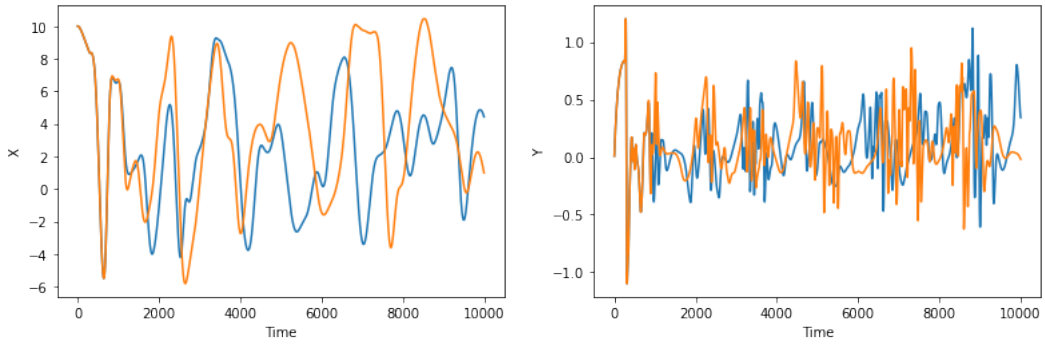

Figure 4: Evolution of $X_0(t)$ (left) and $Y_{0,0}(t)$ (right) for the 10 first iterations (with dt=0.001) for ground truth parameters with different initializer, $X_k(t=0) = 10, \forall k \neq 18$ and with $X_{18}(t=0) = 1.001$ (blue) and $X_{18}^0 = 1.002$ (orange)

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.

3.2.2 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 4) 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.

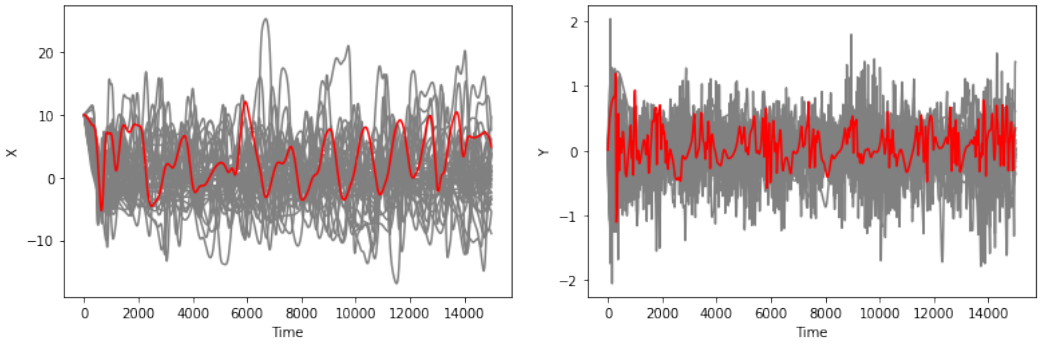

Figure 5: Evolution of $X_0(t)$ (left) and $Y_{0,0}(t)$ (right) for the 15 first iterations (with dt=0.001) for ground truth parameters (red) and 40 samples of tested parameters (grey)

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 (Swanson & Pierrehumbert, 1997). 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.

3.2.3 Limits

Julie/Balaji please help

- Caractère non-stationnaire du climat (contrairement au L96)
- Limite pour les modèles couplés

3.3 Statistical emulators

3.3.1 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.

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.

3.3.2 Commonly used emulators

As mentioned earlier (see 3.1.3), 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

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

- **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

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

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

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

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

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.

3.3.3 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**

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.

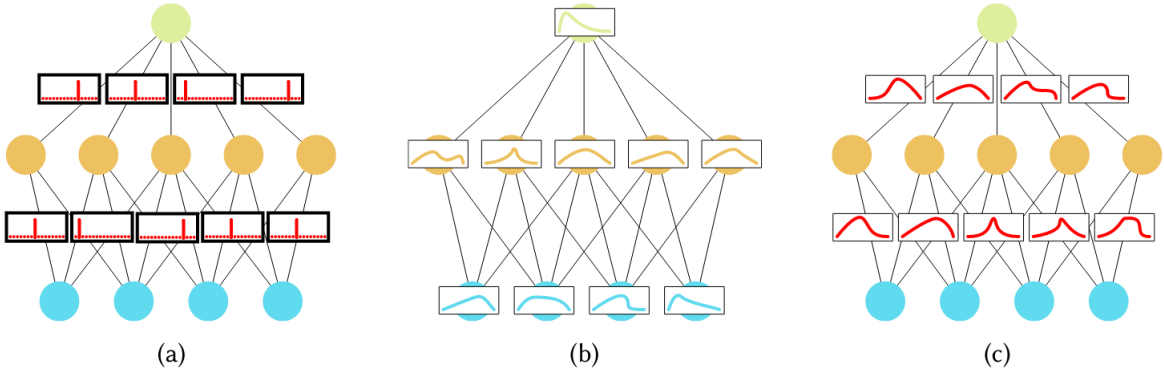

Figure 6: Artificial neural network (a), stochastic activation neural networks (b), stochastic coefficients neural networks (c). From Jospin et al. [25]

.

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

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. 6a). 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. 6b) or stochastic weights (see Fig. 6c).

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

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

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.

3.4 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 3.2.2), 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.

3.4.1 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.

3.4.2 OMIP - Ocean Model Intercomparison Project

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

need help Parallèlement à un AMIP experiment, nous faisons référence à un OMIP experiment pour un Oceanic General Circulation Model constrained by the sea level atmospheric temperature (??).

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 $f(X) = (X, X^2)^T$.

3.5 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.

3.5.1 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. 25) and it may therefore seem useful to try to reduce the dimension of the metrics.

3.5.2 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 $\sum_{i=1}^{p'} \lambda_i \geq 0.99$.

3.5.3 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. $y' = \sigma(Wy + b)$.

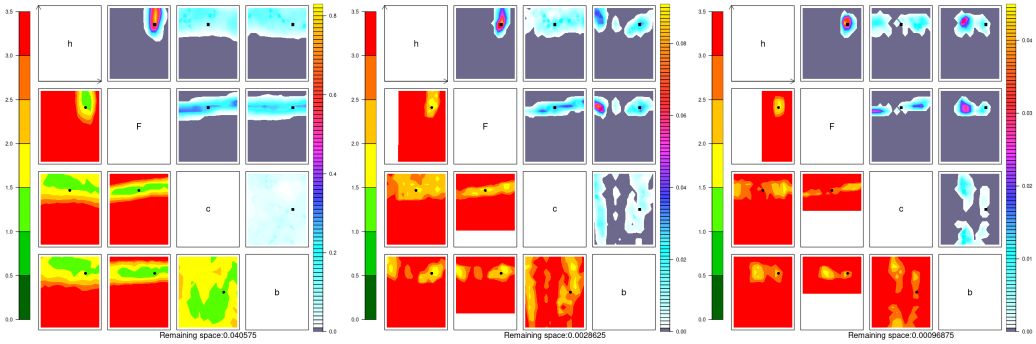

Figure 7: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS**. Wave 1 (left), wave 2 (center), wave 3 (right).

4 Experimental results

4.1 Exploratory approach

In this section we will take an exploratory approach to observe the behaviour of the History Matching methodology applied to Lorenz-96. To do so, we will test the methodology under several constraints: the choice of metrics, the number of samples in the parameter space for each wave, the integration scheme of the numerical model or the effect of noise on the observations.

4.1.1 Metrics choice

The choice of metrics is particularly important for numerical model parameterization. As we have explained, Schneider et al. [6] show in their article that the metrics $f(X,Y) = (X, \bar{Y}, X^2, X\bar{Y}, \bar{Y^2})^T$ allow to reconstruct the set of the 4 parameters.

This is confirmed experimentally (see Fig.). A first wave of History Matching performed with a GPR emulator on 40 samples (sampled with an LHS maximin) allows to exclude 0.959425 from the parameter space (see Fig. 7(left)) and we can see that the method converges by excluding 0.99903125 from the parameter space after three waves (see Fig. 7(right)).

It seems interesting, now that the method has shown its convergence in the simple case, to determine whether it converges in the case where the metrics chosen represent only the slow component (metrics X and X^2) or the fast component (metrics $\bar{Y}$ and $\bar{Y^2}$). Using the analogy of a coupled ocean-atmosphere model, the question is whether the whole model can be parameterised by observing only the atmosphere or by observing only the ocean.

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. 8) in the third wave that the space of excluded parameters is smaller than that of the second wave. explanation

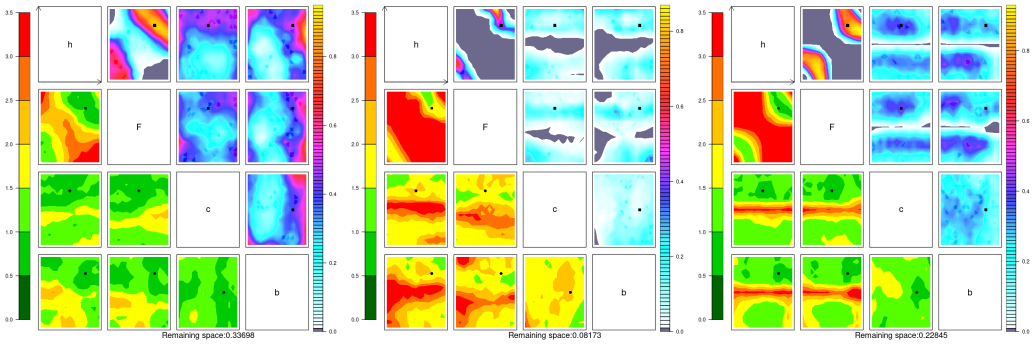

Figure 8: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS** only using fast component metrics ($\bar{Y}, \bar{Y^2}$). Wave 1 (left), wave 2 (center), wave 3 (right).

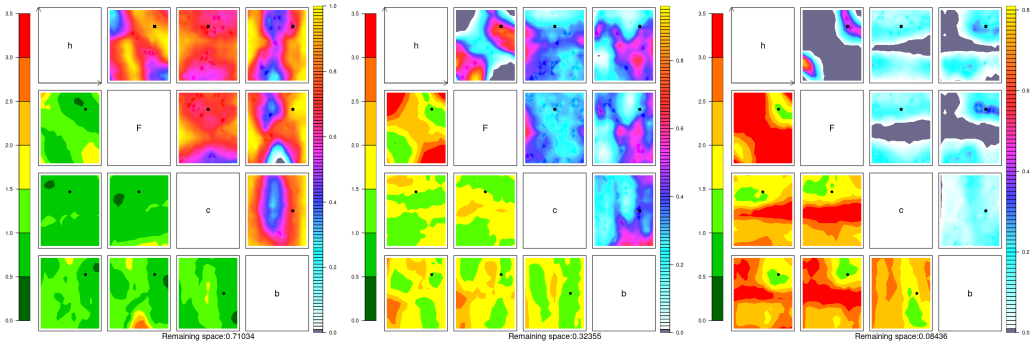

Figure 9: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS** only using Eq. [18] metrics. Wave 1 (left), wave 2 (center), wave 3 (right).

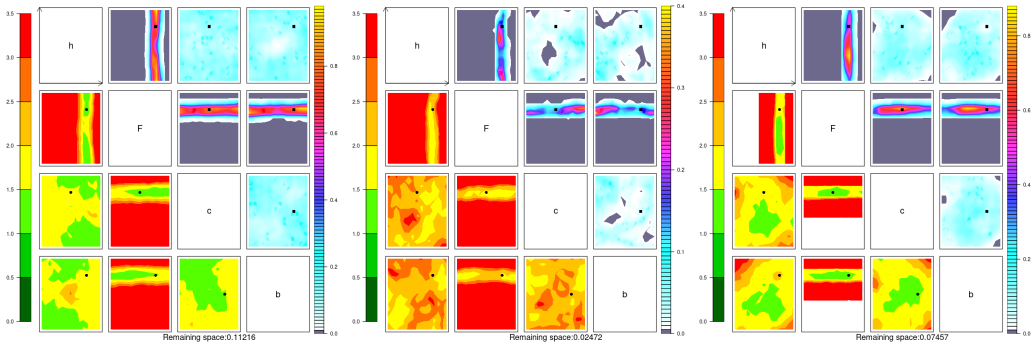

Figure 10: History Matching performed with **GPR emulator** on **40 samples** for each wave sampled with **maximin LHS** only using Eq. [18] metrics. Wave 1 (left), wave 2 (center), wave 3 (right).

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. 10) that the use of metrics describing the fast components allow 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 7).

4.1.2 Non-iteravtive History Matching

Also, it seems interesting to compare the results obtained using an iterative approach (see Fig. 7) and using a non-iterative approach, i.e. sampling the parameter space only once with a larger number of samples in order to train an emulator with smaller uncertainties on its predictions. The experiment described in Fig. Refreffig:HM performs 3 waves each using 40 samples, so we will use $40 \times 3 = 120$ samples to evaluate the non-iterative approach with the same number of simulated samples as with the iterative approach.

It seems that an iterative approach to HM is more efficient than a non-iterative approach. Indeed, the iterative approach reduces the parameter space by 0.99903125 after three waves and a total of 120 samples, while the non-iterative method reduces it by 0.97761.

This can be explained by the distribution of samples that an iterative approach leads to. We can indeed see (Fig. 12) that the samples become more and more concentrated around the ground truth parameters with an iterative approach which allows the emulator to become more accurate in its predictions in this area.

This leaves an important question. Given a fixed number of samples (due to limited computational capacity), what would be the optimal distribution of these samples over a set of waves in order to reduce the NROY estimate as much as possible while remaining conservative enough to ensure that ground truth parameters are not

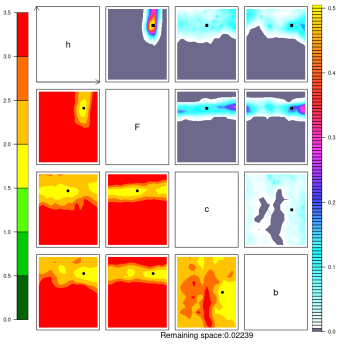

Figure 11: First wave of HM performed with **GPR emulator** on **120 samples** sampled with **maximin LHS**

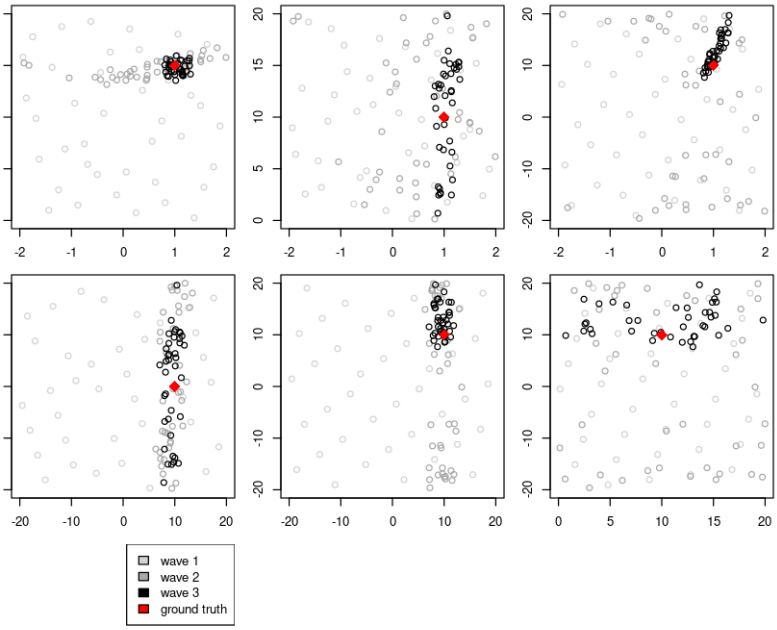

Figure 12: Inputs (generated by LHS sampling) for three waves of History Matching with 40 samples for each wave using GPR

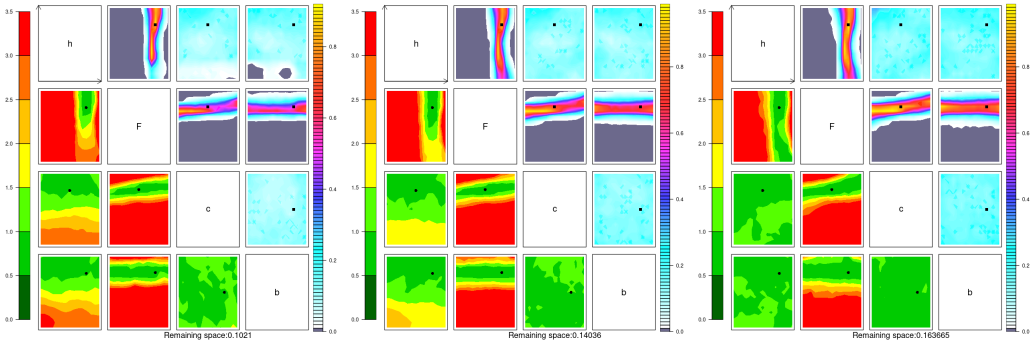

Figure 13: First wave of History Matching performed with different levels of noise. $\mu = 0.05$ (left), $\mu = 0.1$ (center), $\mu = 0.5$ (right).

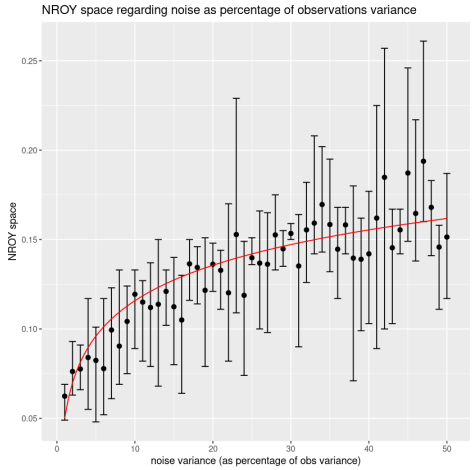

Figure 14: NROY space after first wave of HM regarding noise variance

rejected.

4.1.3 Noise effect

When parameterising a climate model (of the ocean, atmosphere or other), the observations are subject to various uncertainties, mainly related to the measuring instruments used to record them. We are therefore interested here in the effect of increasing uncertainty on the observations of Lorenz-96.

By

4.1.4 Intergration scheme

The integration scheme, which provides a numerical solution to the system of partial differential equations describing the dynamic system under study, is accompanied by an error (which is bounded) intrinsic to it. We can therefore consider this error as a divergence of the model and it is then interesting to know the impact of such an error on the parameterisation by History Matching.

We will therefore use here the Euler integration scheme (the one used for the other experiments is an RK4) and observe the results of 3 waves of History Matching.

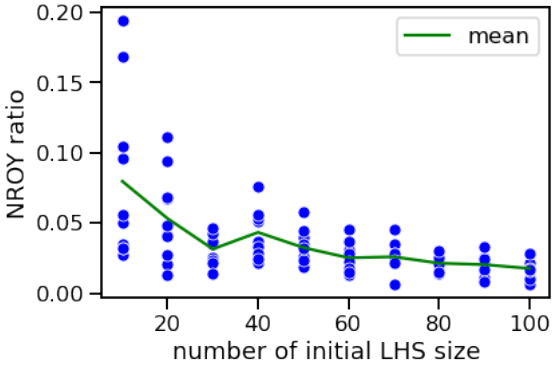

Figure 15: NROY ratio after first wave of HM regarding the number of inital samples in parameters space using LHS sampling methods.

4.2 Space filling design

4.2.1 Number of samples

As we have described previously (subsection 3.1.1), the number of samples is of great importance for the parameterisation of dynamic systems. It is recommended, as a rule of thumb (see Williamson et al. [5]), to use $10 \times p$ samples, with p the number of parameters. In this section, we seek to determine the extent to which this rule is applicable to the parameterisation of Lorenz-96 with History Matching.

The first thing to note is that a larger number of samples ensures that the parameter space is reduced to a higher order. Indeed, we can see (Fig. 15) that the NROY ratio and its variance decrease with the number of initial samples. It is then necessary to find a compromise between calculation time and performance. It seems that beyond 60 samples the performance gain is not worth the cost in terms of computation time, as well as below 20 samples the variance on the NROY space is too large to obtain consistent results.

We will therefore stay in the $[20, 60]$ sample range for our future experiments. Note that this seems to confirm the rule of $\sim 10 \times p$ samples which we will apply for the majority of experiments (unless explicitly stated otherwise).

4.2.2 Sampling methodology

The most widely used sampling methodology for History Matching is the maximin LHS. Here we want to know whether this shows a significantly better performance than random sampling and whether the Quasi-Monte Carlo Sampling Sobol method shows interesting results.

- **Random Sampling**

  It is first interesting to note that random sampling initially reduces the parameter space significantly after two waves, by 0.99371 (see Fig. 16(enter)). On the other hand, it seems that when the space is strongly reduced, random sampling eventually leads to a rejection of the ground truth parameters as can be seen in Fig. 16(right) where the ground truth value of the parameter b ends up being ruled-out.

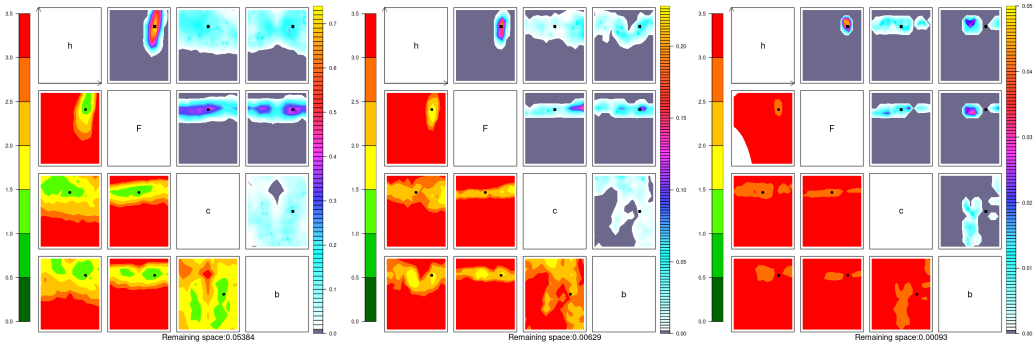

Figure 16: History Matching performed with GPR for 40 samples per waves sampled with **random sampling**. Wave 1(left), wave 2 (center), wave 3 (right).

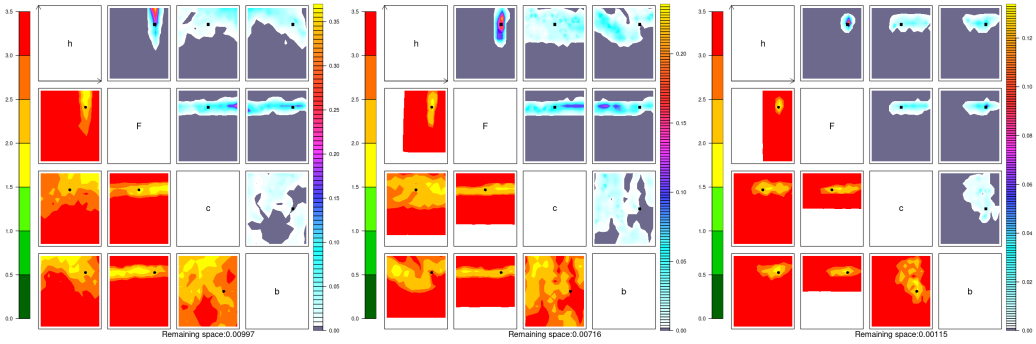

Figure 17: History Matching performed with GPR for 40 samples per waves sampled with **QMC Sobol sampling**. Wave 1(left), wave 2 (center), wave 3 (right).

It might be interesting to see to what extent being less conservative in calculating implausibility – choosing the v-th maximum value as explained in 3.1.4– would allow the sampling to converge to the correct parameters.

- **Quasi-Monte Carlo with Sobol sequence Sampling**

  We can first note that the sampling by Quasi-Monte Carlo method with Sobol sequence allows HM to converge well towards the ground truth parameters. Moreover, the reduction of the parameter space is of the same order of magnitude as the Latin Hypercube maximin sampling, i.e. 0.99885 (see Fig. 20) with QMC Sobol sampling and 0.99903125 (see Fig. 7) with LHS maximin sampling after three waves.

  A major advantage of QMC methods is that they are much faster than LHS sampling which can become costly in terms of computing time when the number of points to be sampled becomes large which occurs when the remaining NROY space is small. For this reason, we think that it could be interesting to consider Quasi-Monte Carlo methods for History Matching as they seem to show similar performances to the LHS maximin with a reduced computation time.

- **Correlation optimized Latin Hypercube Sampling**

  It would seem that sampling by LHS with correlation as an optimization crite-

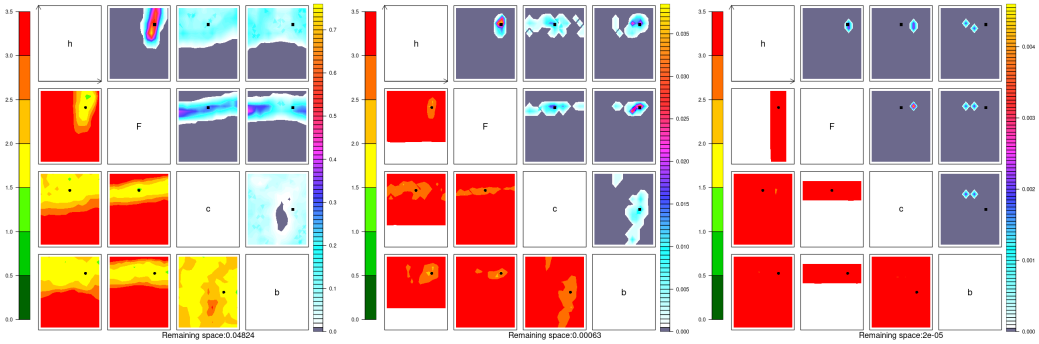

Figure 18: History Matching performed with GPR for 40 samples per waves sampled with **LHS optimized with correlation criterion**. Wave 1(left), wave 2 (center), wave 3 (right).

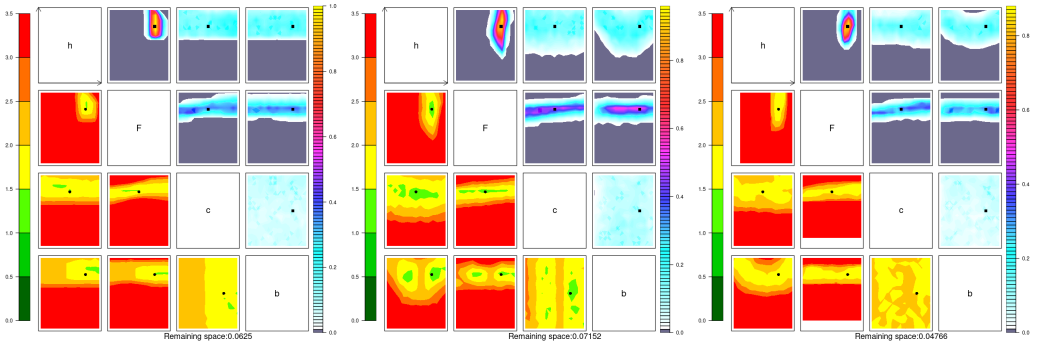

Figure 19: History Matching performed with **linear regression** for 40 samples per waves sampled with maximin LHS. Wave 1(left), wave 2 (center), wave 3 (right).

rion would be the method that would allow the HM to converge most rapidly to the ground truth parameters.

4.3 Emulators

In this section, we explore the possibility of replacing Gaussian Processes and linear regressions as emulators by two other types of statistical models, namely random forests and Bayesian Neural Networks. We start by evaluating the performance of the first two types of emulators in order to be able to compare their results with the two models proposed to replace them.

4.3.1 Linear regressor

It would appear (see Fig. 19) that linear regression models alone do not significantly reduce the parameter space. In particular, this emulator performs poorly in reducing the parameter space projected into the $c(b)$ dimension. This can be explained by the very strong non-linearity and the chaotic aspect of the variable Y where b is involved.

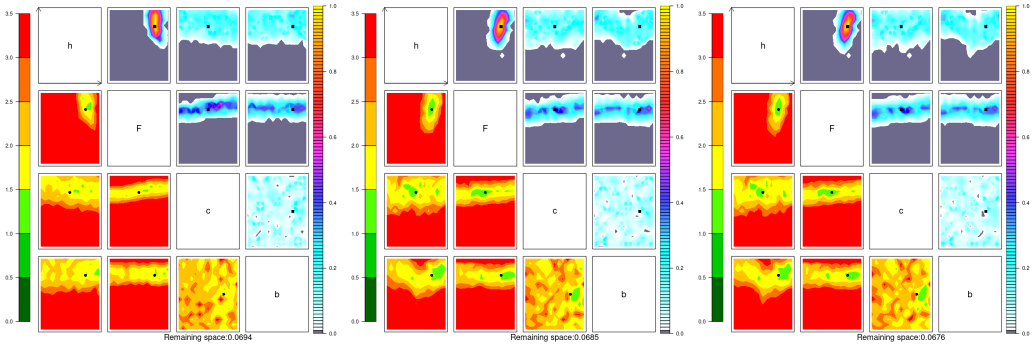

Figure 20: History Matching performed with **Random Forest Regressors** for 40 samples per waves sampled with maximin LHS. Wave 1(left), wave 2 (center), wave 3 (right).

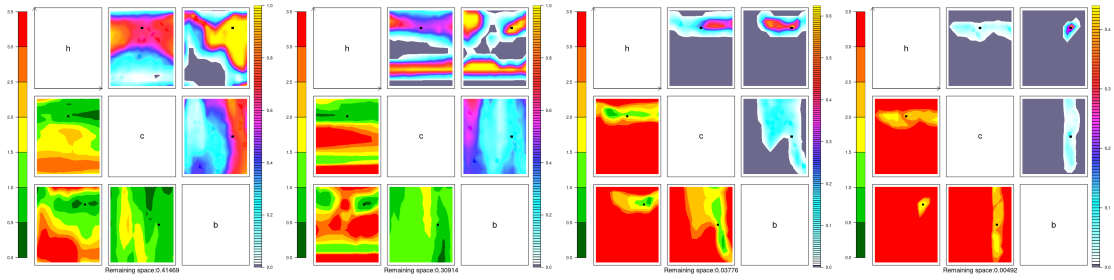

Figure 21: First four waves of History Matching performed in an AMIP style experiment with $(\bar{Y}, \bar{Y^2})^T$ metrics.

4.3.2 Gaussian Process regressor

4.3.3 Random Forest

4.3.4 Bayesian Neural Networks

As the Bayesian Neural Networks model has only recently been integrated with the History Matching library, the experiments for this emulator are not complete enough and we will not develop this section here. The first results are available in appendix (see Appendix **??**).

4.4 CMIP style experiments

4.4.1 AMIP experiments

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. 22 and 21) 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 to converge. The first metrics reduce the parameter space by 0.99987 after three

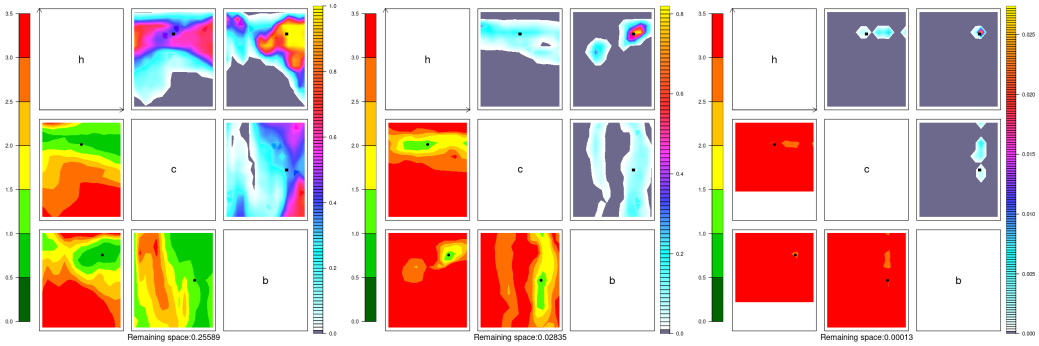

Figure 22: First three waves of History Matching performed in an AMIP style experiment with Eq. [18] metrics

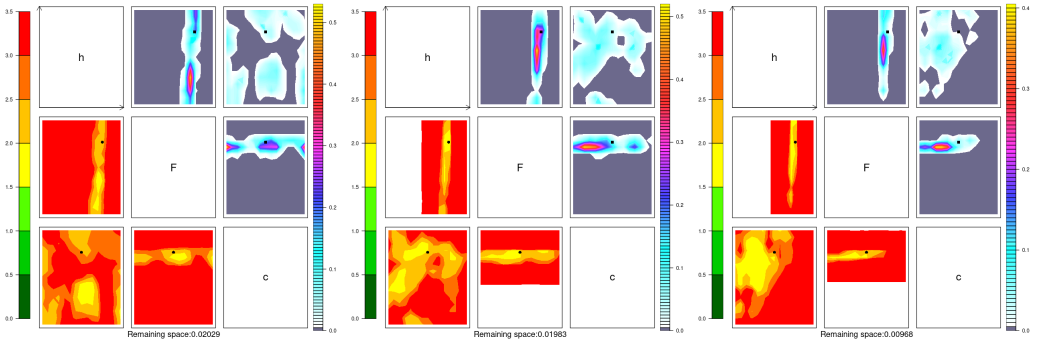

Figure 23: First three waves of History Matching performed in an OMIP style experiment with $f(X) = (X, X^2)^T$ metrics

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.

4.4.2 OMIP experiments

We now place ourselves in the context of an OMIP-style experiment, i.e. we seek to parameterise the slow component by forcing it with observations of the fast component (see 3.4.2 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. 23) that after one wave of MH we have reduced the parameter space by 0.97971 and by 0.98032 after three waves.

4.4.3 Conclusion CMIP

In general, HM seems to be suitable for CMIP-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.

Aide pour finir : sur l implication de l application pour modele couples ocean atmosphere en mip style experiment

4.5 Dimensionality Reduction of metrics space

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.

4.5.1 Principal Component Analysis

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. 7). 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

But as we can see in Fig. 24, 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.

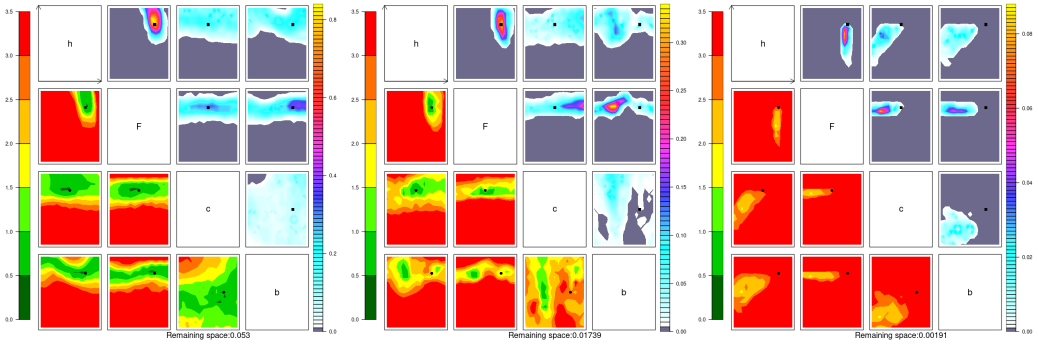

Figure 24: First three waves of History Matching performed with PCA (10 cp). First wave (left), second wave (center), third wave (right)

4.5.2 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 Opening

5.1 Environemental and societal impact

5.2 Opening

6 Conclusion

7 Appendix

7.1 Bayesian Neural Networks Results

7.2 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.

7.3 Pictures

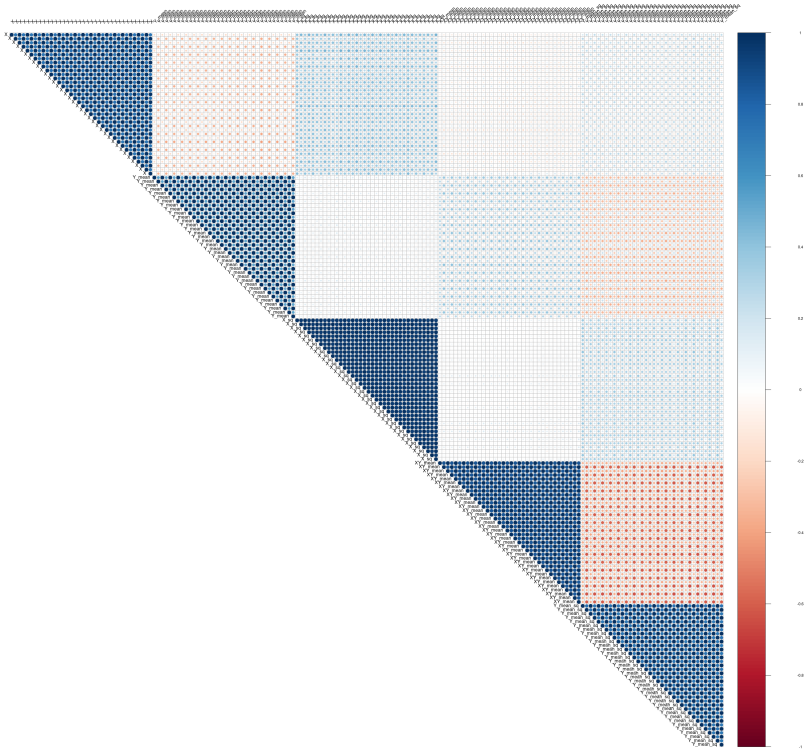

Figure 25: Correlation of metrics $(X, \bar{Y}, X^2, X\bar{Y}, \bar{Y^2})^T$ for 40 samples generated with LHS sampling method.

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.