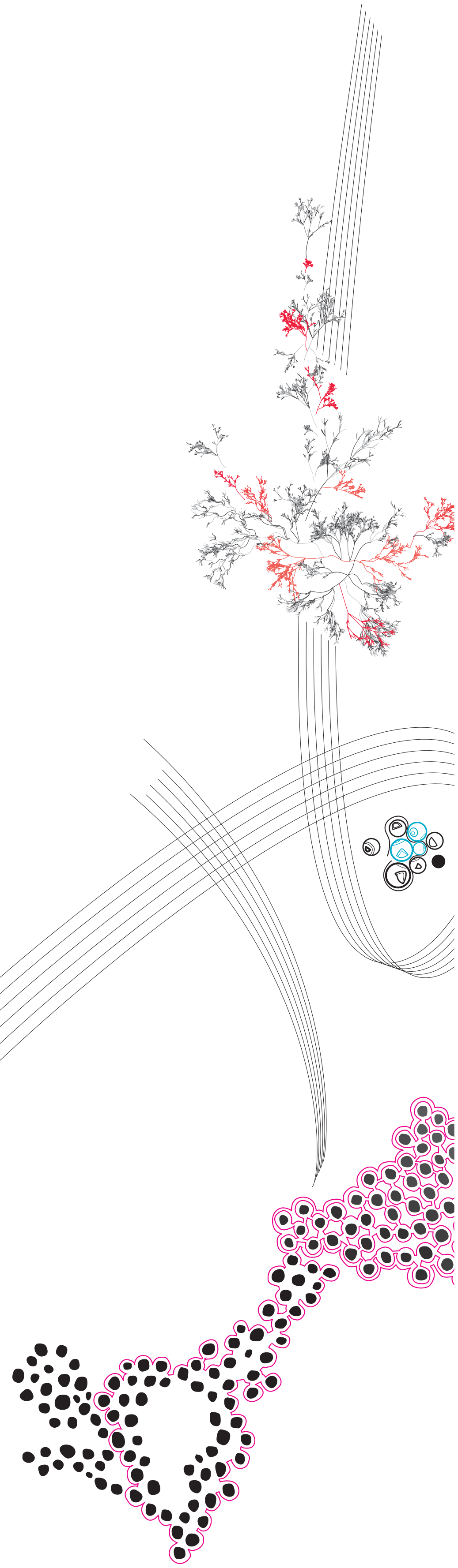
BSc Thesis Applied Mathematics and
Technical Computer Science



Comparing alternative methods for a priori optimization of the slow-mode closure for the Lorenz 96 system

Justin Christian de Groot

Supervisor: prof. dr. ir. B. Geurts, dr. E. Mocanu

July, 2022

Department of Applied Mathematics and
Department of Technical Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

This paper was written as my bachelor's thesis in Applied Mathematics and Technical Computer Science.

I want to thank prof. dr. ir. B. Geurts and dr. E. Mocanu for supervising my bachelor's thesis. They were incredibly helpful throughout the process, by guiding me with the planning and writing, as well as giving crucial insight whenever needed.

I also want to thank MSc. S. Ephrati for giving insights into the field as well as help guide me throughout the process. Additionally, he provided a framework for the a priori approach and was willing to help me solve any issue I had with the implementation.

Comparing alternative methods for a priori optimization of the slow-mode closure for the Lorenz 96 system

Justin. C. de Groot

July, 2022

Abstract

When we want to coarsen a weather forecast model with multiple time scales, we do this by approximating the changes of the slow variables caused by the fast variables, which is the true subgrid. We will narrow this down to the Lorenz 96 model, which is a simplified weather forecast model with a slow and fast scale. We try to approximate the true subgrid using a parameterized subgrid, by minimizing a loss function which defines the error of the approximation. There are currently two approaches known for defining this loss function, an a priori and an a posteriori approach, where the former used theoretical deduction and the latter uses empirical observation. A previous paper[7] which introduced the a posteriori approach showed that it gave better results. However, the results found for the a priori approach was done using linear regression, while machine learning was used for the a posteriori approach. So, we wish to use machine learning to solve the a priori approach, to determine if the a priori approach can achieve the same results when also being solved by machine learning. Here we show that this is not the case. While machine learning did improve the accuracy of the results, there was still a big gap between the results found and the a posteriori results.

Keywords: discrete parameterizations, machine learning, optimization, chaotic systems

Contents

1 Introduction

When regulations and decisions need to be made on measures against climate change, it is important that we can predict the long-term consequences that these actions have on our planet. Think about the prediction of how many degrees our earth will heat up by the year 2100 [9] and how it influences decisions like limiting emissions. A prime example would be the Paris Agreement [12], where many countries agreed to limit global warming below 2, preferably 1.5 degrees Celsius, by decreasing greenhouse gas emissions.

Of course, only predicting the increase in temperature has little benefit if we do not know the consequences that the rise of temperature has. One consequence is that the weather becomes more extreme [4] [8], including worsening many disasters like storms and heatwaves [6] [5], which makes the weather more difficult to forecast. Consequently, we need better, and often more complex, models to accurately predict the weather. This means that predicting the long term consequences actions have on the weather becomes more computationally expensive and thus less feasible.

Since more complex weather forecast models often use multiple time scales, we wish to coarse grain these models, by approximating the faster scales of the model using discrete parameterization. Here we use the Lorenz 96 model [10], since it mimics certain properties of the atmosphere, including having multiple times scales, a slow and a fast scale, making it a good comparison to more complex models. It is also a simple chaotic system, meaning it is computationally cheap as well as transparent. This allows us to determine the accuracy of our simplified models.

The basic idea of the slow-mode closure is to approximate the effects of the fast mode using a parameterized subgrid. We do this by minimizing a loss which defines the error of the approximation. There are two approaches for this, which differ in how the loss function is constructed. The a priori approach constructs the loss function using theoretical deduction[3], while the a posteriori approach uses empirical observation to construct the loss function [7].

In the same paper where the a posteriori approach was introduced[7], it was also shown that the a posteriori approach gave better results than the a priori approach. However, in their paper, they minimized the a priori loss function using a linear regression algorithm (least squares), while using machine learning, in the form of iterative optimization techniques (gradient descent), for the a posteriori approach. This might have given the a posteriori approach an advantage, since iterative optimization techniques could give better results than linear regression algorithms. Therefore, in this paper we will use iterative optimization techniques to figure out if the a priori loss function can achieve the same level of accuracy as the a posteriori approach, when also using machine learning. We do this by comparing the performance of several readily available iterative optimization techniques against the results found for the a posteriori approach.

In the next section we introduce the Lorenz 96 system as well as explaining the slow-mode closure. In section 3 we elaborate on the a priori approach to the slow-mode closure, as well as talk about the optimization algoritms we will use with the a priori approach. In section 4 we talk about the implementation, including how we will simulate the results and the metrics we use to judge the accuracy of the slow-mode closures. We also verify the implementation with prior results. In section 5 we go over the methodology we use to get the results we need, while giving the results in section 6. In section 7 we discuss flaws or other arguable points that exist in the argumentation of our thesis. In section 8 we draw a conclusion about the results found and we end with possible research directions in section 9.

2 Lorenz 96

The Lorenz 96 model [10] is a simplified weather model designed as a crude model of the atmosphere, where it incorporates the interaction of variables of different scales. One scale represents the slow movement of processes in the atmosphere itself, while the second one represent fast moving smaller scale influences, such as turbulence. Therefore, the Lorenz 96 system is a coupled system of 2 types of variables and is defined as:

$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b} \sum_{j=J(k-1)+1}^{kJ} Y_j; \qquad k = 1, ..., K \tag{1}$$

and

$$\frac{dY_j}{dt} = -cbY_{j+1}(Y_{j+2} - Y_{j-1}) - cY_j + \frac{hc}{b} X_{\lfloor (j-1)/J \rfloor + 1}; \qquad j = 1, ..., JK. \tag{1a}$$

The definitions of the variables are extended to all values of k and j by using cyclic boundaries: $X_{k+mK} = X_k$ and X_{j+nKJ}, where m and n are integers. The X_k variables are large-amplitude, low-frequency variables, which are all coupled to many small-amplitude high-frequency Y_j variables [3]. The constant K denotes the amount of low-frequency variables $X_1, ..., X_K$, while the constant J denotes the amount of high-frequency variables $Y_{1+kJ}, ...Y_{J+kJ}$ associated with each low-frequency variable X_k. Additionally, we have the forcing constant F, a coupling term h, a spatial-scale ratio b and a time-scale ratio c, all of which are real values.

2.1 slow-mode closure

When we want to simulate long-term weather effects, we only care about the slow-mode, but to calculate the low-frequency variables we also need to calculate the high-frequency variables. However, this is very expensive, so we wish to convert the coupled system into a single equation only considering the low-frequency effects. Therefore, we wish to approximate the coupling function h(Y) defined as

$$h_k(Y) = -\frac{hc}{b} \sum_{j=J(k-1)+1}^{kJ} Y_j.$$

For this purpose, we introduce the parameterized subgrid U_p that does not depend on the high-frequency variables Y_j. We then describe the low-frequency variables of the slow-mode closure X_k^θ with

$$\frac{dX_k^\theta}{dt} = -X_{k-1}^\theta(X_{k-2}^\theta - X_{k+1}^\theta) - X_k^\theta + F + U_{p,k}(\theta, X).$$

Here, we define $\theta \in \mathbb{R}^P$ as a set of parameters used to optimise the parameterized subgrid model such that the slow-mode closure accounts for the influence of the fast variables on the slow variables in the coupled Lorenz 96 system [7].

3 A priori approach

In the previous section we found an alternative to the coupled Lorenz 96 system, in which we introduced a parameterized subgrid model U_p. To let the slow-mode closure of the Lorenz 96 model mimic the slow-mode coupled model, we have to define the parameterized subgrid model U_p and choose a parameter set θ such that this slow-mode closure is an accurate representation [7]. There are two approaches to calculate this subgrid model, namely an a priori approach and an a posteriori approach. Both ways use the data calculated from the full coupled model, however the a priori approach tries to minimize a difference based on theoretical deduction, while the a posteriori approach tries to minimize a difference based on empirical observation. We will only go over the a priori approach.

We define the true subgrid model U as a replacement of h(Y) to be

$$U(X,Y) = h(Y),$$

which we estimate from the truth using the forward Euler method as follows

$$U_k(X(t)) = -[-X_{k-1}(t)(X_{k-2}(t) - X_{k+1}(t)) - X_k(t) + F] + \frac{X_k(t+\Delta t) - X_k(t)}{\Delta t},$$

for a fixed time step size Δt [3]. Inline with [7] and [3] we will approximate U using a third order polynomial

$$U_{p,k}(\theta, X(t)) = \theta_0 + \theta_1 X_k(t) + \theta_2 X_k^2(t) + \theta_3 X_k^3(t),$$

where the vector $[\theta_0, \theta_1, \theta_2, \theta_3]$ shall be denoted as θ. We then approximate U with U_p by minimizing the loss function

$$L_{priori}(\theta) = \frac{1}{2}\int_0^T ||U(X(t)) - U_p(\theta, X(t))||_2 dt, \tag{2}$$

where $||\cdot||_2$ stands for the L2 norm. As mentioned in the research question, we will try to minimize the loss function using several optimization techniques, however some optimization techniques need the first and second derivative, or otherwise known as gradient and hessian, to work. Therefore, we shall first derive those, before looking at the optimization algorithms that we will test.

3.1 derivatives

We only need the discrete form of the loss function 2, so we will only derive the discrete form, however deriving the continuous form should be similar. We start of by rewriting the loss function to a discrete form as well as writing out the L2 norm.

$$L_{priori}(\theta) = \frac{1}{2}\sum_{t=0}^{T} \sqrt{g_1(\theta, X(t))},$$

where we use a helper function g_1 defined as

$$g_1(\theta, X(t)) = \sum_{k=1}^{K} (U_k(X(t)) - U_{p,k}(\theta, X(t)))^2.$$

We calculate the first derivative, the gradient, from the definition

$$\nabla L_{priori}(\theta) = [\frac{\partial L_{priori}(\theta)}{\partial \theta_0}, \frac{\partial L_{priori}(\theta)}{\partial \theta_1}, ..., \frac{\partial L_{priori}(\theta)}{\partial \theta_P}].$$

We then calculate the partial derivative for one parameter of θ, where we have to use the chain rule and product rule to get the first term, before rewriting it with another helper function.

$$\frac{\partial L_{priori}(\theta)}{\partial \theta_i} = \frac{1}{2}\sum_{t=0}^{T} \frac{\frac{1}{2}\frac{\partial g_1(\theta, X(t))}{\partial \theta_i}}{\sqrt{g_1(\theta, X(t)}} = \frac{1}{2}\sum_{t=0}^{T} \frac{g_{2,i}(\theta, X(t)}{\sqrt{g_1(\theta, X(t)}}, \qquad i = 0, ..., P, \tag{3}$$

with

$$g_{2,i}(\theta, X(t)) = \frac{1}{2}\frac{\partial g_1(\theta, X(t))}{\partial \theta_i} = \sum_{k=1}^{K} (U_{p,k}(\theta, X(t))_k - U_k(X(t)))X_k^i.$$

We also derive the second derivative, the hessian, from the definition.

$$H_{L_{priori}(\theta)} = \begin{bmatrix} \frac{\partial^2 L_{priori}}{\partial^2 \theta_0^2} & \cdots & \frac{\partial^2 L_{priori}}{\partial \theta_0 \partial \theta_P} \\ \vdots & & \vdots \\ \frac{\partial^2 L_{priori}}{\partial \theta_P \partial \theta_0} & \cdots & \frac{\partial^2 L_{priori}}{\partial^2 \theta_P^2} \end{bmatrix}$$

We derive one element of the matrix by deriving equation 3 again, where we use the quotient rule, followed by the product and chain rule to get the first complex fraction.

$$\begin{aligned} (H_{L_{priori}(\theta)})_{i,j} &= \frac{\partial^2 L_{priori}(\theta)}{\partial \theta_i \partial \theta_j} = \frac{\partial \frac{\partial L_{priori}(\theta)}{\partial \theta_i}}{\partial \theta_j} \\ &= \frac{1}{2}\sum_{t=0}^{T} \frac{\sqrt{g_1(\theta, X(t))}\frac{\partial g_{2,i}(\theta, X(t))}{\partial \theta_j} - g_{2,i}(\theta, X(t))\frac{1}{\sqrt{g_1(\theta, X(t))}}(\frac{1}{2}\frac{\partial g_1(\theta, X(t))}{\partial \theta_j})}{g_1(\theta, X(t)} \\ &= \frac{1}{2}\sum_{t=0}^{T} \frac{g_1(\theta, X(t))g_{3,i,j}(\theta, X(t)) - g_{2,i}(\theta, X(t))g_{2,j}(\theta, X(t))}{g_1(\theta, X(t))\sqrt{g_1(\theta, X(t))}}, i, j = 0, ..., P \end{aligned}$$

with

$$g_{3,i,j}(\theta, X(t)) = \frac{\partial g_{2.i}(\theta, X(t))}{\partial \theta_j} = \sum_{k=1}^{K} X_k^{i+j}.$$

3.2 optimization algorithms

Aside from the least squares method that was used before [3] [7], we will use the iterative optimization algorithms from scipy.optimize [2]. While each iterative optimization method is different from another, their main distinguishing feature is which derivatives of the loss function they use. Aside from the linear regression method least squares that we use for reference, we also have the iterative optimization algorithm Nelder-mead simplex algorithm and Powell's method that only use the loss function itself. We then have the Broyden-Fletcher-Goldfarb-Shanno algorithm, which is the only method that uses the loss function and its first derivative. Lastly, we have the methods that use the loss function as well as both its derivatives. They are the Newton-Conjugate-Gradient algorithm, Trust-region Newton-Conjugate-Gradient algorithm, Trust-Region Truncated Generalized Lanczos algorithm and lastly the Trust-Region Nearly Exact algorithm.

4 Implementation

When working on the implementation, a framework was provided. This framework was able to generate data for the Lorenz 96 model, both the full coupled system as well as the parameterized model. It also had an example for the a priori approach, where pre-generated data of the full model was loaded and the true subgrid U was calculated using the Euler forward method. Then a parameter set θ was trained using the least squares method, before generating the data of the parameterized model. To get all the results we want, a couple important elements need to be added. We mainly need to add the methods mentioned in the last section, including the loss functions, as well as several metrics(graphs) to quantify(visualize) the performance of the parameterized models. In this section we go over the metrics we used to judge performance as well as how we ran the simulations. There are some differences from our reference [7], however we will see in the replication of their results that there do not appear to be (significant) differences, except for one metric which will be explained later on.

4.1 Accuracy metrics

When comparing the a priori and a posteriori approach[7], three different metrics were used to quantify the accuracy of the slow-mode closures found. The first metric is the mean distance metric, which compares the distance of the average value between the slow-mode closure and the coupled system. This metric is defined as

$$d(\bar{X}^{\theta}, \bar{X}) = \frac{1}{K} |\sum_{k=1}^{K} \bar{X}_k^{\theta} - \sum_{k=1}^{K} \bar{X}_k|, \tag{4}$$

where the bar ($\bar{}$) denotes a row-wise mean. The second measure is the Wasserstein-1 distance, which measures the effort necessary to transform one dataset into another and is defined as

$$W_1(X^{\theta}.X) = max(\sum_i |sort(X_i^{\theta}) - sort(X_i)|), \tag{5}$$

where the sort($\cdot$) represents a sorted copy of the data set, flattened into a single vector [11]. The last metric is the Hellinger distance, which measures the distance between 2 probability distribution functions(PDF) and is defined as

$$H(p, q) = \frac{\sqrt{2}}{2} \sum_i |\sqrt{p} - \sqrt{q}|, \tag{6}$$

where p and q are PDFs [7].

We will use the same metrics, so we can compare the results we find with prior results, however we will calculate all the metrics from the PDFs of the data, instead from the data itself. We can do that, since the long-term behaviour of the Lorenz 96 is given by the PDF of the low-frequency variables [14]. This also makes the metrics and the graphs directly correlated, thus giving a clearer representation of what the metrics mean in relation with the graph. To calculate the metrics from the PDFs does mean that we need to use different calculations of the metrics. To calculate the metrics from the PDfs we will use continuous functions, so we have to use a numerical integrator (quad from scipy) to calculate the integrals. The mean distance is now defined as

$$d(p, q) = \int_{-\infty}^{\infty} xp(x) - xq(x)dx, \tag{7}$$

where p and q are PDFs. The Wasserstein-1 metric for PDFs is defined as

$$W_1(P,Q) = \int_{-\infty}^{\infty} |P(x) - Q(x)| dx, \tag{8}$$

where P and Q are cumulative distribution functions of the PDFs p and q [13]. And lastly we have the Hellinger distance which is defined as

$$H^2(p,q) = \frac{1}{2} \int_{-\infty}^{\infty} (\sqrt{p(x)} - \sqrt{q(x)})^2 dx, \tag{9}$$

where p and q are PDFs [1].

4.2 Simulation

The coupled Lorenz 96 system as described by equation 1 is regarded as the truth and is solved using a fourth order Runge-Kutta technique, with a maximum time step size of 0.001 MTU, where 1 MTU is roughly 5 atmospheric days. The coupled system is run with the values as seen in table 1. The results we will try to replicate[7] used 7000 MTU of data, after letting the run 3000 MTU without gathering data, to allow the system to enter the chaotic attractor. However, this seemed like an unnecessary long period, so we collect 240 MTU of data, after letting the system run for 40 MTU. We will see if this is warranted in section 4.3, when we try to replicate the previously found results to test our implementation. Another difference between the simulations, is that we do not separate the gathered data into a training and testing dataset, rather having a random initial state for each slow-mode closure.

parameter	symbol	value
number of X variables	K	8
number of Y variables per X variable	J	32
coupling constant	h	1
forcing term	F	20
spatial-scale ratio	b	10
time-scale ratio	c	4

Table 1: The default parameter settings used for the L96 system.

The slow-mode closure is ran with a time step coarser than the one used for the truth, using a time step of 0.005 MTU, and even coarser for some cases used for verification of the implementation in section 4.3. The true subgrid we calculate using the Euler forward method always uses a time step of 0.005 MTU.

4.3 Verification

In this section we will try to replicate previously found results for the a priori approach [7] to validate our implementation. We will compare the trajectory generated by the parameterized models using the slow-mode closure against the behaviour of the low-frequency variables of the coupled model, i.e. the truth. We will only compare the least squares method as well as the slow-mode closure without an approximated subgrid, so $\theta = [0,0,0,0]$, against the truth, as these were the only methods that had previous results. We generate the results using the settings described above, where our results were generated slightly differently as mentioned previously. We do this several times, each time the data generated by the slow-mode closure models are coarser, so we start with a time step of 0.005 MTU, then a time step of 0.010 MTU and lastly using a time step of 0.020 MTU.

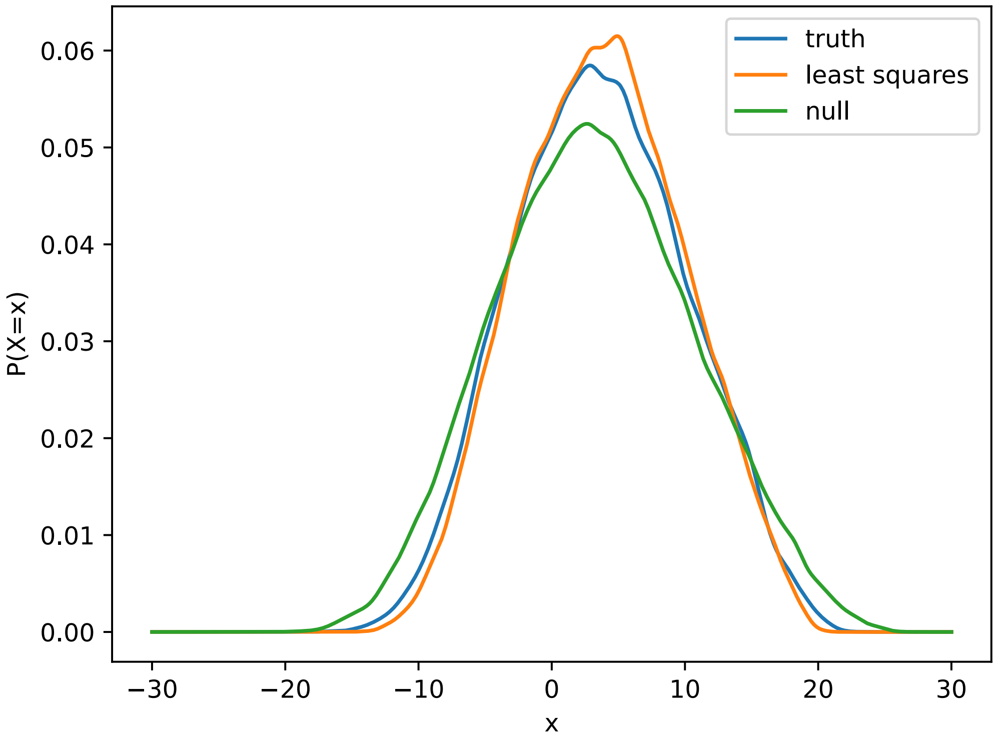


FIGURE 1: The PDFs of the systems, where both the coupled Lorenz 96 system and the slow-mode closures were solved on a 0.005 MTU grid. On the x-axis we see values of the slow-mode variables, with the corresponding probability on the y-axis.

Methods	Mean distance	Wasserstein-1 distance	Hellinger distance
Least squares (reference)	0.1532	0.3502	0.1431
Least squares	0.2157	0.3430	0.0453
Null (reference)	0.2939	0.8399	0.2745
Null	0.3343	0.8125	0.0899

TABLE 2: The several metrics between the PDFs of the slow-mode closures and the slow variables of the coupled Lorenz 96 system, with a time step size of 0.005 MTU for the slow mode closures. The results we try to replicate are annotated with (reference).

We can see the results when we run the slow-mode closures on a grid of 0.005 MTU in the figure 1 and table 2. We can also see the results when we calculate the slow-mode closures using a time step size of 0.010 MTU in the figure 2 and table 3. However, we were unable to get results for the time step size of 0.020 MTU, due to a limitation in the framework. Nonetheless, we can use the results we were able to get to verify our implementation.

When we look at the mean distance, we see that the mean difference and Wasserstein-1 distance are comparable, however the Hellinger distance from our result is vastly different from the reference results. This is because the Hellinger distance used to calculate the results for the reference metric, see equation 6, does not actually calculate the Hellinger

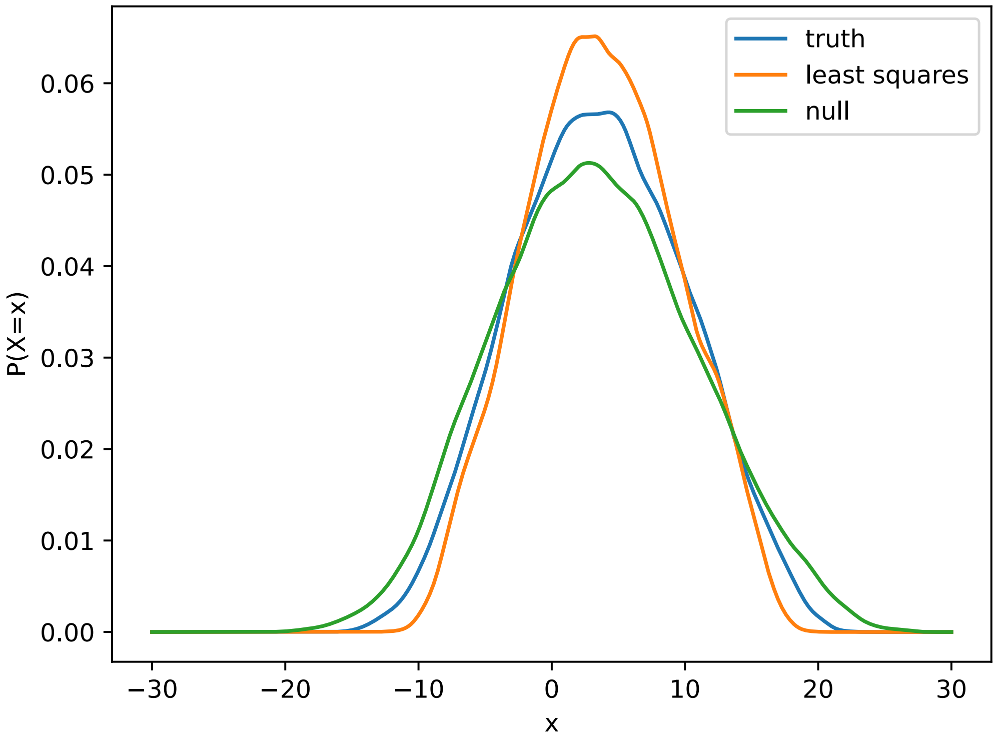


FIGURE 2: The PDFs of the systems, where the coupled Lorenz 96 system was solved with a time step of 0.005 MTU, whereas the slow-mode closures were solved on a 0.010 MTU grid. On the x-axis we see values of the slow-mode variables, with the corresponding probability on the y-axis.

Methods	Mean distance	Wasserstein-1 distance	Hellinger distance
Least squares (reference)	0.1690	0.6466	0.3032
Least squares	0.1108	0.6628	0.0978
Null (reference)	0.2902	0.8411	0.2751
Null	0.2833	0.8338	0.0970

TABLE 3: The several metrics between the PDFs of the slow-mode closures and the slow variables of the coupled Lorenz 96 system, with a time step size of 0.010 MTU for the slow mode closures. The results we try to replicate are annotated with (reference).

distance. We can prove this by showing that using their equation, we can get a Hellinger distance larger than 1, while the Hellinger distance is supposed to be bounded between 0 and 1 [1]. Assume we have probability density functions p and q, where $p(x = 1) = 1$ and $q(x = 2) = 1$. Then we have

$$H(p, q) = \frac{\sqrt{2}}{2}\sum_i |\sqrt{p} - \sqrt{q}| = \frac{\sqrt{2}}{2}(|\sqrt{1} - \sqrt{0}| + |\sqrt{0} - \sqrt{1}|) = \sqrt{2} > 1,$$

thus this metric does not calculate the Hellinger distance. So we can compare the Hellinger distance in the results we calculate but not with the reference results.

5 Methodology

Similar to what we did in section 4.3, we will compare the trajectories of the slow-mode closures against the true behaviour of the slow-modes of the coupled Lorenz 96 model. This time however, we will also compare the iterative optimization techniques mentioned in section 3.2. We will compare the methods on 3 factors.

The first factor is accuracy, where we will compare the slow-mode closures found by the methods using the 3 metrics we specified in section 4.1. The metrics are the mean distance, defined with equation 7, the Wasserstein-1 distance, see equation 8 and lastly the Hellinger distance, given by equation 9. However, from section 4.3 we know that we can not compare our Hellinger distance with the Hellinger distance calculated in our reference results. Thus, we can only use it when comparing the results we generate.

The second factor we will compare the methods on is computational complexity. Here we will compare the methods on the combined time of the time needed to preprocess the training data and the time needed to calculate the approximated subgrid using the training data. We will also gather the time needed to generate the data using the coupled system and slow-mode closures, so we can put the combined training time into reference. The training time for the null method is not the preprocessing time, rather it is simply 0.

Lastly, we wish to observe how the methods perform when they are given less training data. We will test this by running the simulation four times, after each time we use 25% less training data. To judge their performance we will use the previous 2 factors.

6 Results

Since the time needed for generating is the same for each case (we still generate the same amount to not influence the generation of the PDFs and just pass a subset of the data to the methods), we only need to give it once. All the other data is dependent on the amount of data, so it needs to be given separately. Generating 280 MTU of data using the coupled Lorenz 96 system takes about 900 seconds, while generating the same amount of data using the slow-mode closures only takes around 25s.

100% of the training data is given to the methods.

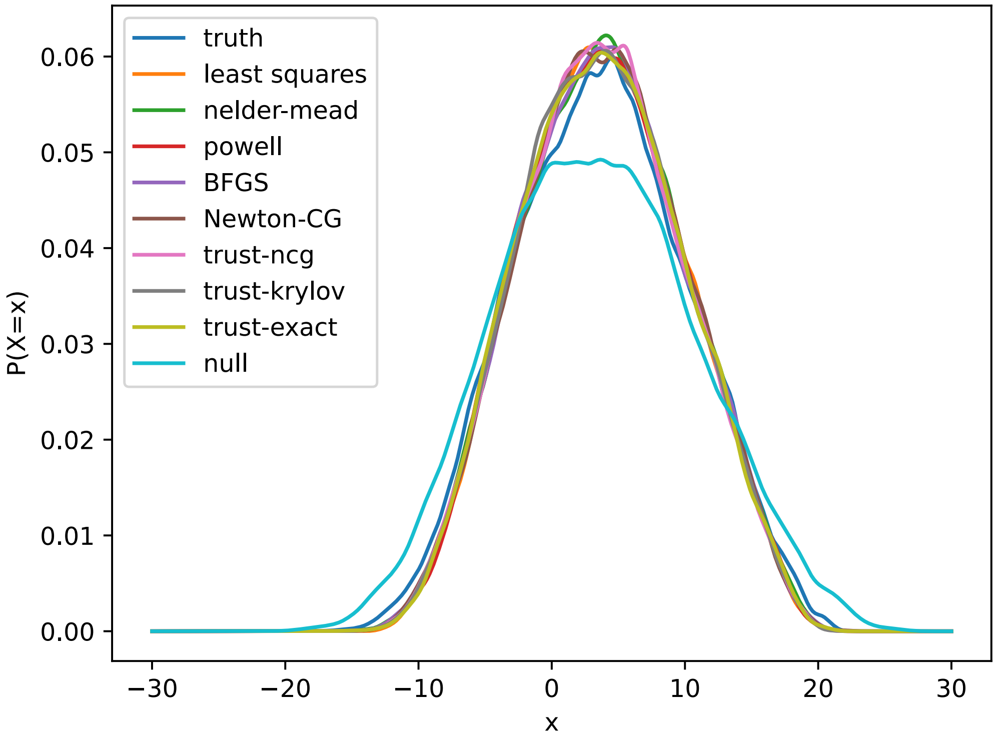


FIGURE 3: The PDFs of the systems calculated on a 0.005 MTU grid, where the methods got 100% of the training data. On the x-axis we see values of the slow-move variables, with the corresponding probability on the y-axis.

Methods	Mean distance	Wasserstein-1 distance	Hellinger distance	Runtime (s)
null	0.3137	0.8492	0.0940	0
least squares	0.1271	0.3322	0.0515	0.22
nelder-mead	0.1848	0.3016	0.0379	119.59
powell	0.1120	0.2827	0.0433	92.74
BFGS	0.1506	0.2983	0.0395	67.10
Newton-cg	0.1624	0.3243	0.0431	50.45
trust-ncg	0.0646	0.3155	0.0440	119.43
trust-krylov	0.0853	0.2916	0.0457	102.48
trust-exact	0.0962	0.2858	0.0438	98.53

TABLE 4: The several metrics between the PDFs of the slow-mode closures and the slow variables of the coupled Lorenz 96 system, where the methods got 100% of the training data.

Time needed to process the generated data into a format suitable for training using the Euler forward method took 16.93 seconds.

75% of the training data is given to the methods.

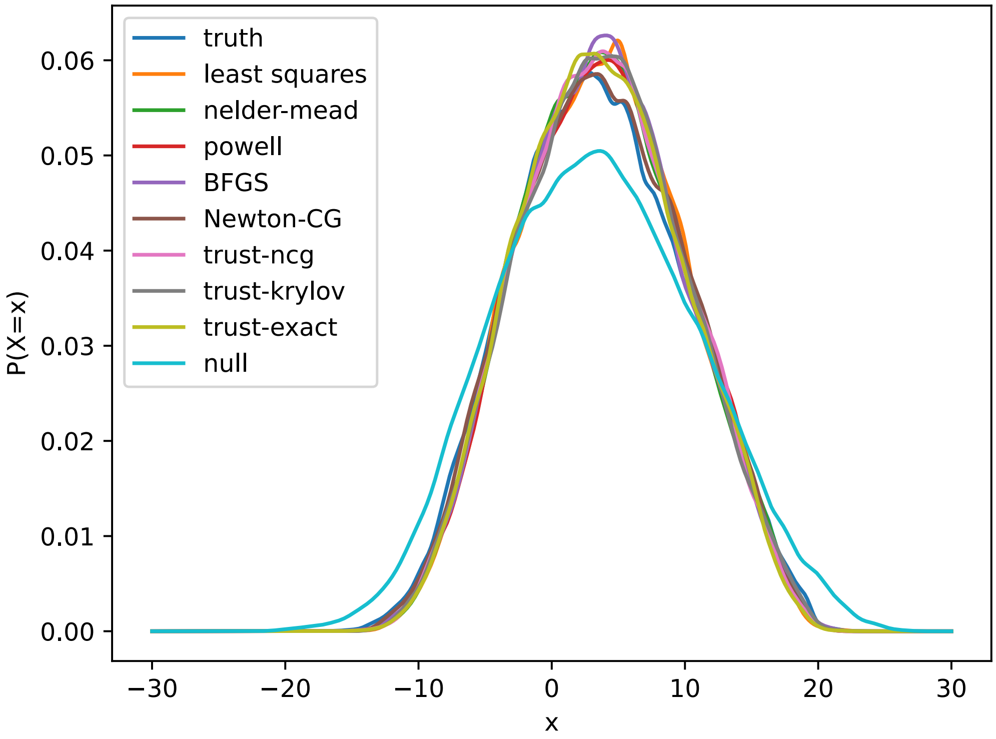


FIGURE 4: The PDFs of the systems calculated on a 0.005 MTU grid, where the methods got 75% of the training data. On the x-axis we see values of the slow-move variables, with the corresponding probability on the y-axis.

Methods	Mean distance	Wasserstein-1 distance	Hellinger distance	Runtime (s)
null	0.1744	0.8685	0.1038	0
least squares	0.2450	0.3692	0.0417	0.16
nelder-mead	0.2079	0.3022	0.0328	88.42
powell	0.2570	0.3363	0.0343	58.29
BFGS	0.1578	0.3342	0.0344	17.09
Newton-cg	0.1188	0.1708	0.0218	33.12
trust-ncg	0.1991	0.3145	0.0349	80.26
trust-krylov	0.2428	0.3327	0.0343	85.46
trust-exact	0.1120	0.3074	0.0379	63.06

TABLE 5: The several metrics between the PDFs of the slow-mode closures and the slow variables of the coupled Lorenz 96 system, where the methods got 75% of the training data.

Time needed to process the generated data into a format suitable for training using the Euler forward method took 11.64 seconds.

50% of the training data is given to the methods.

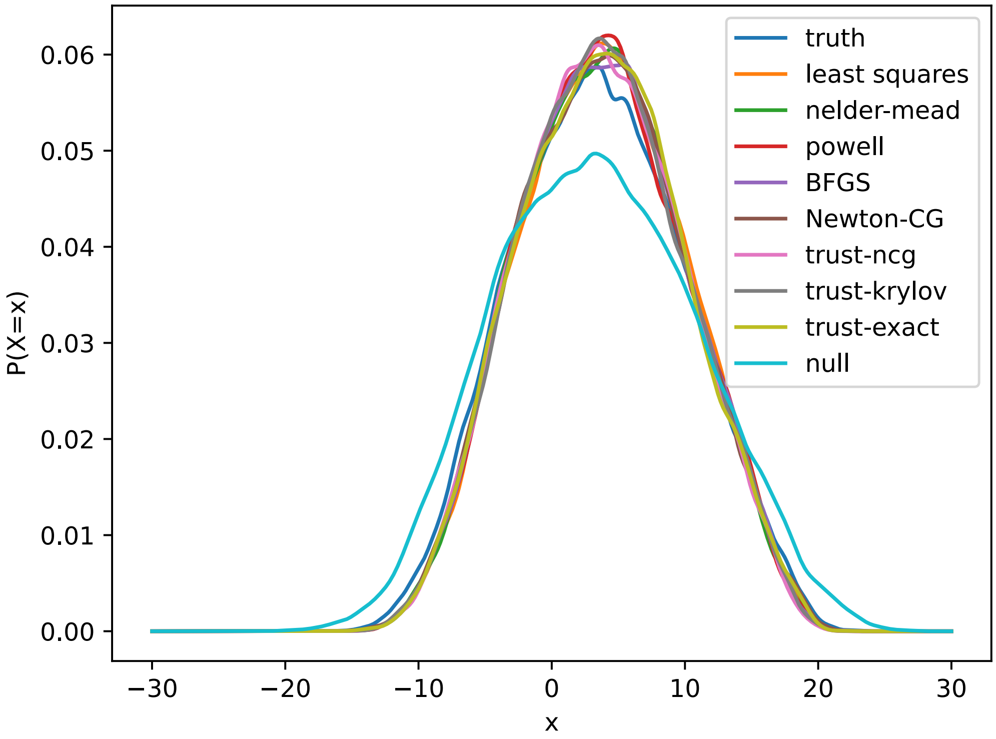


Figure 5: The PDFs of the systems calculated on a 0.005 MTU grid, where the methods got 50% of the training data. On the x-axis we see values of the slow-move variables, with the corresponding probability on the y-axis.

Methods	Mean distance	Wasserstein-1 distance	Hellinger distance	Runtime (s)
null	0.2133	0.8929	0.0965	0
least squares	0.2654	0.3508	0.0366	0.12
nelder-mead	0.1989	0.3020	0.0328	63.62
powell	0.1827	0.2806	0.0366	42.26
BFGS	0.2164	0.2968	0.0350	15.60
Newton-cg	0.1836	0.2626	0.0286	31.06
trust-ncg	0.1000	0.2950	0.0401	62.80
trust-krylov	0.1629	0.2928	0.0359	58.14
trust-exact	0.2059	0.3174	0.0341	60.26

Table 6: The several metrics between the PDFs of the slow-mode closures and the slow variables of the coupled Lorenz 96 system, where the methods got 50% of the training data.

Time needed to process the generated data into a format suitable for training using the Euler forward method took 10.30 seconds.

25% of the training data is given to the methods.

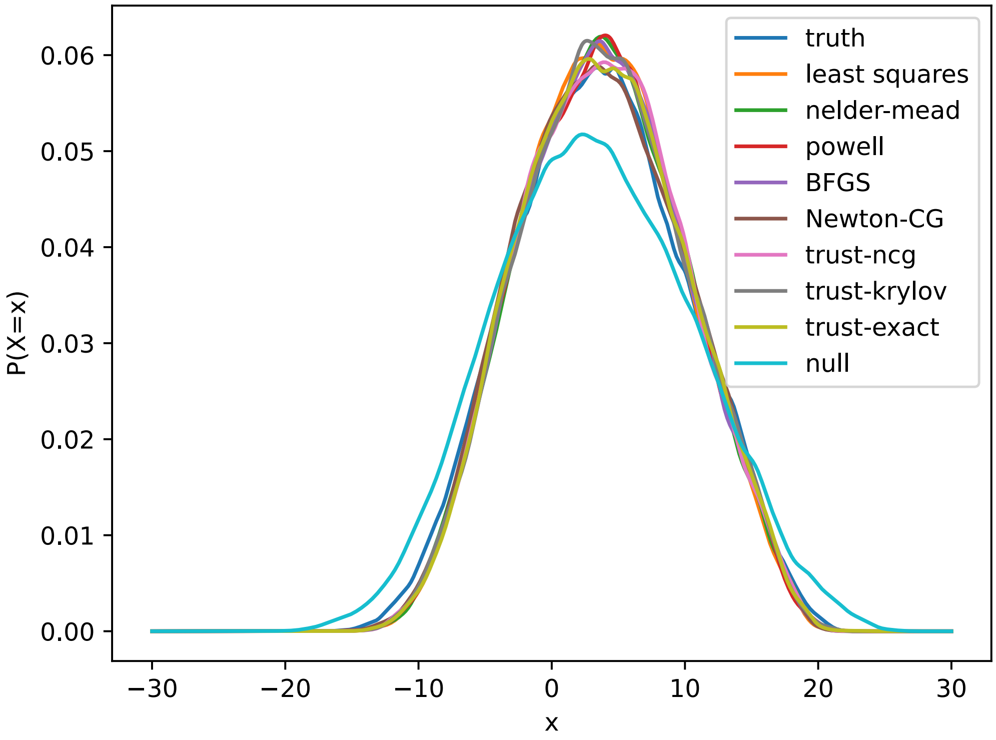


Figure 6: The PDFs of the systems calculated on a 0.005 MTU grid, where the methods got 25% of the training data. On the x-axis we see values of the slow-move variables, with the corresponding probability on the y-axis.

Methods	Mean distance	Wasserstein-1 distance	Hellinger distance	Runtime (s)
null	0.3348	0.8298	0.0934	0
least squares	0.1538	0.3407	0.0436	0.06
nelder-mead	0.2018	0.3071	0.0386	35.00
powell	0.2173	0.3164	0.0379	23.61
BFGS	0.2386	0.3164	0.0380	7.00
Newton-cg	0.1152	0.1801	0.0320	15.22
trust-ncg	0.2400	0.3274	0.0386	56.52
trust-krylov	0.2727	0.3160	0.0351	44.16
trust-exact	0.2480	0.3065	0.0391	32.78

Table 7: The several metrics between the PDFs of the slow-mode closures and the slow variables of the coupled Lorenz 96 system, where the methods got 25% of the training data.

Time needed to process the generated data into a format suitable for training using the Euler forward method took 4.69 seconds.

7 Discussion

We can see from the increase in the Mean distance as the training data given to the methods decreases that there is a correlation and that 180 MTU of training data is not enough data to get the best results. This also means that we can potentially decrease the mean distance further, by giving the methods even more training data. When we look at the Wasserstein-1 distance, it is hard to see a correlation between the metric and the amount of data given to the methods. But when we look at the first and last case, when we give 100% and 25% of the data to the methods, we do see a decrease in performance. Lastly, there does not appear to be a clear pattern with the Hellinger distance as the training data shrinks. Combining all of this, it might be possible to decrease the distance by giving it more data, but it probably will not give significantly better results, except perhaps the mean distance.

We do see an interesting phenomenon with the Newton-cg algorithm. It performs around the same level as the other methods with 100% of the data and 50%, see table 4 and 6. However it performs significantly better than the others in the other 2 cases, see tables 5 and 7. This could mean that the algorithm does not reach the minimum, rather getting stuck in a saddle or something similar. However, it could also mean that the performance of the method heavily depends on the random initial condition, which could mean that the performance of the methods fluctuates depending on the random initial state. if this is the case, then the methods could all be at a disadvantage compared to if we had used a training and testing data split, rather than a random initial condition. It is hard to know what is the case from just these results, but it is unlikely that it is fully due to the latter, since we only see such large fluctuations for the Newton-cg method. This does not mean that the latter does not have an impact on all the methods, it might still be the case that it influence all the methods, just on a smaller scale.

The fact that the Wasserstein-1 distance and Hellinger distance of the iterative optimization algorithms are generally less than 15% smaller than the results found for the linear regression algorithm, except for the Newton-cg algorithm outliers, does raise the question if the increase in runtime is worth this increase in performance. If we compare combined time of the preprocessing and training, we see that the iterative optimization techniques are a couple of times slower. I would personally say that this increase in accuracy is worth it, considering that even with the slowest iterative optimization technique, the training time is (somewhat) small, when compared to the difference in the time necessary to generate the data between the coupled system and slow-mode closure.

8 Conclusion

From the results we see that iterative optimization algorithms give a better result than the linear regression algorithm least squares. It does not however bridge the gap between the a priori and a posteriori approach, since the best results found was using Newton-cg with a mean distance of 0.1188 and wasserstein-1 distance of 0.1708, while the best results we have for a posteriori are a mean distance of 0.0152 and a wasserstein-1 distance of 0.0368 [7]. This means that the results found for the a priori approach are still worse and while it is possible that factors like random initial states for the slow-mode closures have a negative impact on the performance, it is unlikely that it made the results several times worse. So we were unable to achieve the same level of accuracy using a priori as shown for the a posteriori approach. This also means we did not disprove the claim that the a posteriori approach gives better results than the a priori approach [7].

9 Recommendations

Since the a priori approach gives worse results than the a posteriori approach, even while both using machine learning, it would indicate that the loss function plays an important role determining the maximum performance. It could be worthwhile to explore even more loss functions to see if any give a better result than the current a posteriori loss function. When testing the loss function, it should be considered if calculating the derivatives is worth it, when wanting to use iterative optimization algorithms that use the derivatives.

Given the great performance of the a posteriori approach, it is likely that the performance now is close to the limit for a third order polynomial. So rather than trying to find another approach or loss function, it might be better to try to improve the a posteriori approach by using a different parameterized subgrid. This could be simply a higher order polynomial, but it could also be of a different form.

Another thing to consider is that we are using the Lorenz 96 model as a toy model for coarse-graining and would like to be able to do the same for more complex models. Rather than trying a more complex model, we could also try to find out if certain parameters of the Lorenz 96 model allows us to make coarse-graining easier/harder. It might for example be the case that by increasing the forcing constant F, we increase the space of possible values for the slow-mode variables, which might make it more difficult to approximate using our current approach. If we are able to find these parameters, we can investigate (to a certain degree) if the approach we currently use holds up for more complex models.

References

[1] Hellinger distance. https://en.wikipedia.org/wiki/Hellinger_distance. Version: May 2022

[2] Optimization(scipy.optimize). https://docs.scipy.org/doc/scipy/tutorial/optimize.html. Version: 2022

[3] Arnold, H. M.: Moroz, I. M. and Palmer, T. N. : Stochastic parametrizations and model uncertainty in the Lorenz '96 system. In: Philosophical Transations of the Royal Society A: Mathematical, Physical and Engineering Sciences 371 (2013). http://dx.doi.org/10.1098/rsta.2011.0479. – DOI 10.1098/rsta.2011.0479

[4] Easterling, D. R.: Meehl G. A., Parmesan C., Changnon S. A., Karl T. R. and Mearns L. O.: Climate extremes: Observations, modeling, and impacts. In: Science 289 (2000), S. 2068–2074. http://dx.doi.org/10.1126/science.289.5487.2068. – DOI 10.1126/science.289.5487.2068

[5] Elsner, J. B.: Kossin J. R. and Jagger, T. H. : The increasing intensity of the strongest tropical cyclones. In: Nature 455 (2008), S. 92–95. http://dx.doi.org/10.1038/nature07234. – DOI 10.1038/nature07234

[6] Emanuel, K.: Increasing destructiveness of tropical cyclones over the past 30 years. In: Nature 436 (2005), S. 686–688. http://dx.doi.org/10.1038/nature03906. – DOI 10.1038/nature03906

[7] Haan, D. de: A priori versus a posteriori optimization of the slow-mode closure for the Lorenz 96 system. 2021

[8] IPCC, 2013: Summary for Policymakers. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)], 1-30. – DOI 10.1017/CBO9781107415324.004

[9] IPCC, 2018: Summary for Policymakers. In: Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty, Cambridge University Press, Cambridge, UK and New York, NY, USA, [Masson-Delmotte, V., P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P.R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J.B.R. Matthews, Y. Chen, X. Zhou, M.I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield (eds.)], 3–24. – DOI 10.1017/9781009157940.001

[10] Lorenz, E. N.: Predictability: a problem partly solved. In: Seminar on Predictability, 4-8 September 1995 Bd. 1. Shinfield Park, Reading : ECMWF, 1995, 1-18

[11] Ramdas, A.: Garcia N. and Cuturi M.: On Wasserstein Two Sample Testing and Related Families of Nonparametric Tests. (2015). http://dx.doi.org/10.48550/arXiv.1509.02237. – DOI 10.48550/arXiv.1509.02237

[12] UNFCCC: Paris Agreement. https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement. Version: 2022

[13] Vallendar, S. S.: Calculation of the Wasserstein Distance Between Probability Distibutions on the Line. In: Theory of Probability & Its Applications 18 (1972). http://dx.doi.org/10.1137/1118101. – DOI 10.1137/1118101

[14] Wilks., D. S.: Statistical methods in the atmospheric sciences. Bd. 100. Academic Press, 2011