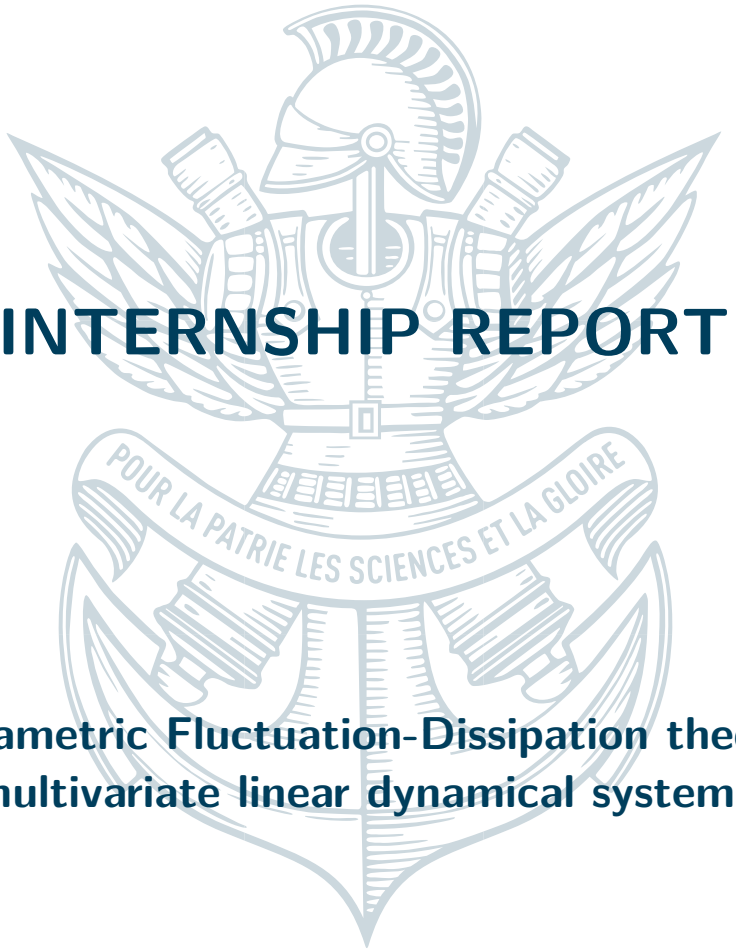

INTERNSHIP REPORT

Non-parametric Fluctuation-Dissipation theorem on multivariate linear dynamical systems

August 27, 2022

MAP594

Amaury LANCELIN, X2019
Supervisor: | William D. Collins (LBNL)

ÉCOLE POLYTECHNIQUE

INSTITUT POLYTECHNIQUE DE PARIS

BERKELEY LAB

ABSTRACT

In this report, we study a tool for evaluating the response to a perturbation of dynamical systems by using the *fluctuation-dissipation* theorem (FDT). Such approach could for instance be very useful to assess the climate response to an increase in CO_2. Following in the footsteps of F. Cooper in [3], we study a version of this theorem based on non-parametric estimation of PDFs of systems. Such procedure makes use of *Kernel Density Estimators*.
After successfully reproducing his results, we try to tackle the issue of dimensionality. Since the *curse of dimensionality* is likely to appear quickly for multivariate PDF estimation, we try to quantify precisely the decrease in performance due to an increase in the number of dimensions. We restrict our study to *linear models*, for which the FDT holds perfectly.

CONTENTS

ACKNOWLEDGMENTS

I would like to warmly thank my supervisor William D. Collins for his constant support throughout the project, his great availability, and his deep kindness.

I am very grateful for the chance given to me to work in such a good laboratory, in a wonderful working environment. I thank in this regard the Lawrence Berkeley National Laboratory *for material support.*

I also thank my sponsor Ankur Mahesh for his warm welcome, for his recurrent technical support, and for turning me into a Warriors *basketball team fan. Finally, I would like to thank Fenwick Cooper and Travis O'Brien for their contribution to my work, and for having so kindly taken the time to answer my questions.*

PLAGIARISM INTEGRITY STATEMENT

I declare on my honour that what has been written in this work has been written by me. In the case of parts taken from scientific publications, from the Internet or from other documents, I have expressly and directly indicated the source at the end of the quotation or by referencing the publication.

I also declare that I have taken note of the sanctions provided for in case of plagiarism by the current Study Regulations.

INTRODUCTION

One major concern in climate science - and more broadly in the general public - for the past few decades has unquestionably been to know how much the climate will warm up in response to an increase of CO_2 in the atmosphere.

The cause of this warming is due to the so-called *greenhouse effect*. This last is very complex and many parameters come into play. Roughly speaking, the following phenomenon happens: an increase in the greenhouse gases concentrations in the atmosphere (which absorbs longer infrared wavelengths) creates an imbalance between the radiative energy received as input by the sun (mostly shorter visible and near-infrared wavelengths) and that emitted as output by the earth (mostly infrared wavelengths). The system warm up to regain balance between input and output radiations.

In climate science, several quantities are commonly studied to quantify this warming precisely. Two of them are the *Equilibrium Climate Sensitivity* (denoted *ECS* or ΔT_{2xCO_2}) and the *transient climate response* (*TCR*) which are defined as follows according to the *IPCC AR5 WG1 report, Chapter 9, page 817*:

> *Equilibrium climate sensitivity (ECS) is the equilibrium change in global and annual mean surface air temperature after doubling the atmospheric concentration of CO2 relative to pre-industrial levels.*
>
> *The transient climate response (TCR) is the change in global and annual mean surface temperature from an experiment in which the CO2 concentration is increased by 1% yr, and calculated using the difference between the start of the experiment and a 20-year period centered on the time of CO2 doubling.*

The classical way of assessing the climate sensitivity is to rely on very costly *general circulation models* (GCM) which are huge simulations of the Earth's atmosphere or oceans using the *Navier–Stokes* equations on a rotating sphere, with thermodynamic terms for various energy sources (radiation, latent heat). Even if the complexity of these models has continued to grow over the years, **the incertitude on the climate sensitivity is still very high** (with the Coupled Model Intercomparison Project Phase 6 (CMIP6), the range of equilibrium climate sensitivity (ECS) is between 1.8°K and 5.6°K).

In this context, every initiative trying to reduce this range is most welcome since even 1°K can change everything in global warming scenarios. In 1975, C.E Leith [5] came with the idea to use the *Fluctuation-Dissipation theorem* (FDT) - a statistical mechanics result which relates the mean response of a dynamical system to a perturbation to its natural undisturbed variability - in order to estimate climate response to some forcing. Since then, multiple tentative of applying the FDT to assess the equilibrium climate sensitivity on GCMs have been conducted with more or less success. Most of them rely on an approximation of the FDT in which the probability density function of the equilibrium state of the system is assumed to be **Gaussian** (see for instance [2]) : the *quasi-Gaussian FDT*.

The whole point of the project I've been part of is trying to use what is called the *non-parametric Fluctuation-Dissipation Theorem (npFDT)* - a version of the FDT in which one makes no assumption

on the PDF of the system and instead estimate it by a *Kernel Density Estimator* (KDE) - to assess the ECS or the TCR. Moreover, the big idea is to use for the first time satellite observations of the Earth's infrared and near-infrared spectrum (as suggested by [4]) as our "data points" for the estimation instead of the outputs of a GCM. The real interest of this approach **could potentially be to reduce the uncertainty range significantly** on climate sensitivity due to the reduced number of assumptions and approximations made during the process.

To achieve this goal, many obstacles stand in the way, and it is certainly a long-term job. In the next part, I will present what my contribution to this project has been by tackling a small piece of the larger problem.

1
POSITION OF THE PROBLEM

1.1 The Fluctuation Dissipation theorem

First, let's introduce the important statistical mechanics result we are relying on in all the project. Our issue is an example of the general mathematical problem of predicting the change in a dynamical system, for example, the change in the probability density function (PDF) of the system as a response to a change in the parameters of the system. In a broad class of physical systems this problem can potentially be addressed using the *fluctuation–dissipation theorem* (FDT). We introduce this result as presented in [3].

Consider a dynamical system described by a state vector $\mathbf{x}$ and governed by a set of evolution equations that give $d\mathbf{x}/dt$ in terms of $\mathbf{x}$.
The equilibrium state of the system is described by a PDF $\rho(\mathbf{x})$, which is assumed to be differentiable. A forcing $\delta\mathbf{f}(t)$ is applied to the system, meaning that $\delta\mathbf{f}(t)$ is added to the expression for $d\mathbf{x}/dt$.

The mean response to the forcing is given by

$$\langle \delta\mathbf{x}(t) \rangle = \langle \mathbf{x}_f(t) \rangle - \langle \mathbf{x}_0(t) \rangle \,,$$

where $\langle \mathbf{x}_0(t) \rangle$ is the mean state vector of the system at time t in the undisturbed system and $\langle \mathbf{x}_f(t) \rangle$ is the corresponding mean state vector with the applied forcing.

The mean denoted by $\langle \cdots \rangle$ is an ensemble mean, for example, over all realizations of explicitly random terms that appear in the evolution equations, or over a large number of initial conditions if the randomness arises through deterministic equations. The probabilist reader could simply see there an expected value.

The FDT relates the response $\langle \delta\mathbf{x}(t) \rangle$ to a small amplitude forcing $\delta\mathbf{f}(t)$, assumed only applied at times $t > 0$, through the equation

$$\boxed{\langle \delta\mathbf{x}(t) \rangle = -\int_0^t \left\langle \mathbf{x}(\tau) \left[\frac{\nabla \rho(\mathbf{x}(0))}{\rho(\mathbf{x}(0))} \right]^{\mathrm{T}} \right\rangle \delta\mathbf{f}(t-\tau) d\tau} \tag{1}$$

Remark 1
- *The FDT remains an active topic of research interest, for example to extend the range of systems to which it can be applied and to clarify the conditions under which it holds.*
- *The requirement of differentiability of $\rho(\mathbf{x})$ is clear from the appearance of $\nabla\rho$ in expression (1).*
- *The expression (1) is a linear relation that will be valid only in the limit of a small forcing $\delta\mathbf{f}(t)$.*
- *For a linear dynamical system, the expression (1) will be valid without restriction.*
- *For a nonlinear system, the skill of estimate (1) will depend on the magnitude of the forcing vector, but we expect that there is some small but finite range of magnitudes for which Eq. (1) will be of practical use.*

In our case, motivated by the **steady-state response** of the climate, or any dynamical system, we will consider the special case of a steady forcing applied for $t > 0$ and the steady-state response $\langle\delta\mathbf{x}(\infty)\rangle$ in the limit $t \to \infty$, which we hereafter denote $\langle\delta\mathbf{x}\rangle$, given by

$$\langle\delta\mathbf{x}\rangle = \mathbf{L}\delta\mathbf{f} \tag{2}$$

where from expression (1) the matrix $\mathbf{L}$ is given by

$$\boxed{\mathbf{L} = -\int_0^\infty \left\langle \mathbf{x}(\tau)\left[\frac{\nabla\rho(\mathbf{x}(0))}{\rho(\mathbf{x}(0))}\right]^{\mathrm{T}}\right\rangle d\tau = \int_0^\infty \mathbf{\Lambda}(\tau)d\tau} \tag{3}$$

The whole point of the procedure is then the estimation of $\mathbf{L}$, which consist of first estimating $\mathbf{\Lambda}(\tau)$, and then compute an approximation of the integral.

Remark 2 *Throughout the estimation, two important approximations are made. First, since we assume our system is sufficiently ergodic, we are replacing the ensemble mean $\langle . \rangle$ by a time average over the trajectory. Secondly, we are replacing the infinite upper limit in the integral in Eq. (3) by a finite (but large enough) upper limit T.*

It is useful to note at this point that $\mathbf{\Lambda}(0) = \mathbf{I}$, with $\mathbf{I}$ the identity matrix, which may be shown by expressing the expected value as an integral over the state space and integrating by parts.

N.B: In both expressions (1) and (3) a significant simplification arises if the equilibrium system is assumed to be Gaussian, that is, that

$$\rho(\mathbf{x}) = \frac{1}{\sqrt{\det[2\pi\mathbf{C}(0)]}}\exp\left[-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{C}(0)^{-1}\mathbf{x}\right] \tag{4}$$

in which $\mathbf{C}(\tau) = \left\langle \mathbf{x}(\tau)\mathbf{x}(0)^{\mathrm{T}}\right\rangle$ is the lag τ covariance matrix, $\mathbf{C}(0)$ is the covariance matrix, and we have assumed without loss of generality that $\langle\mathbf{x}\rangle = 0$. Combining Eqs. (3) and (4) implies that

$$\langle\delta\mathbf{x}\rangle = \int_0^\infty \mathbf{C}(\tau)\mathbf{C}(0)^{-1}d\tau\delta\mathbf{f} \tag{5}$$

Note that the matrix $\mathbf{\Lambda}(\tau)$ is in this case directly proportional to the lag τ covariance matrix.
This form of the FDT is known as the *quasi-Gaussian FDT*.

1.2 The non-parametric FDT

What follows is the procedure that Fenwick Cooper has introduced in [3].

1.2.1 • Application of the density estimation procedure

If $\rho(\mathbf{x})$ appearing in Eqs. (1) and (3) is not known, then in principle it may be estimated from data. A standard method of estimating a multidimensional PDF is to use the kernel density estimator (Silverman 1986 [9]). For a given choice of kernel function K we estimate the PDF by centering a kernel function of specified scale h at each data point

$$\hat{\rho}(\mathbf{x}; h, n) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right). \tag{6}$$

Here $\hat{\rho}(\mathbf{x}; h, n)$ is the estimated PDF, n is the number of available data points, $\mathbf{X}_i$ is data point i, d is the number of dimensions of the state vector $\mathbf{x}$, and $h > 0$ is conventionally known as the bandwidth in the statistical literature (Silverman 1986 [9]).

Remark 3 *In fact, the density estimate is simply a convolution of the data and the kernel function. As the number of data points $n \to \infty$, the estimate of the PDF $\hat{\rho}(\mathbf{x}; h, n)$ tends to the convolution of the true* PDF $\rho(\mathbf{x})$ *and the kernel function K,*

$$\hat{\rho}(\mathbf{x}; h, n \to \infty) = \rho(\mathbf{x}) * K(\mathbf{x}/h),$$

where the asterisk denotes the convolution operator. Therefore, for any finite value of h the estimated PDF can only be an approximation to the true PDF and, indeed, should be regarded as a biased estimator with the bias being a function of h. As $h \to 0$, K tends to a delta function and the bias tends to zero. We will discuss a bit further the role of h in 1.2.2.

In his paper F. Cooper [3] uses for the sake of simplicity, an isotropic Gaussian kernel with identity covariance - we do the same throughout the project. Hence, it may be written as

$$\boxed{\hat{\rho}(\mathbf{x}; h, n) = \frac{1}{n} \sum_{i=1}^{n} N\left(\mathbf{x}; \mathbf{X}_i, h\right)} \tag{7}$$

where

$$N(\mathbf{x}; \mathbf{y}, h) = \frac{1}{(2\pi h^2)^{d/2}} \exp\left[-\frac{(\mathbf{x} - \mathbf{y})^{\mathrm{T}}(\mathbf{x} - \mathbf{y})}{2h^2}\right]$$

denotes the multivariate normal distribution function with mean $\mathbf{y}$ and standard deviation h, evaluated at $\mathbf{x}$. It is useful to note that

$$\nabla_{\mathbf{x}} N(\mathbf{x}; \mathbf{y}, h) = -\frac{\mathbf{x} - \mathbf{y}}{h^2} N(\mathbf{x}; \mathbf{y}, h).$$

As mentioned in the Remark 2, we make two main approximations in estimating the operator $\mathbf{L}$ in Eq. (3). First, we truncate the integral to have finite upper limit T. Second, we express the ensemble average appearing in Eq. (3) as a time average, justified by requiring the system to be sufficiently

ergodic. Assume that the time interval between each data point $\mathbf{X}_i$ is Δt. Then for some lag $s\Delta t$ the integrand, $\mathbf{\Lambda}(s\Delta t)$, may be approximated by

$$\boxed{\hat{\mathbf{\Lambda}}(s\Delta t) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{X}_{j+s}\left[\frac{\sum_{i=1}^{n}\left(\mathbf{X}_j - \mathbf{X}_i\right)^{\mathrm{T}} N\left(\mathbf{X}_j;\mathbf{X}_i,h\right)}{h^2\sum_{i=1}^{n} N\left(\mathbf{X}_j;\mathbf{X}_i,h\right)}\right]} \tag{8}$$

where, to recap, h is the bandwidth of the kernel density estimation, n is the number of points included in the sum used for the density estimate, and m is the number of points included in the sample used to obtain the time average (in practice we choose $m = n$ as in [3]).

The operator $\mathbf{L}$ may then be approximated as

$$\hat{\mathbf{L}}(m,n,h,\boldsymbol{\mu}) = \sum_{s=0}^{r} \mu_s \hat{\mathbf{\Lambda}}(s\Delta t), \tag{9}$$

where $\boldsymbol{\mu}$ is a vector of weights used to estimate the integral appearing in Eq. (3). In the calculations to be reported subsequently, the method of estimation is the *trapezium rule* applied to the integral truncated at some finite upper limit.

Remark 4 *It is expected that* $\hat{\mathbf{L}}(m,n,h,\boldsymbol{\mu})$ *will be a good approximation of* $\mathbf{L}$ *provided that the numbers* m *and* n *are sufficiently large, the bandwidth* h *is sufficiently small, and the vector* $\boldsymbol{\mu}$ *extends to large enough time lags and has a sufficient number of elements* r *to accurately approximate the integral. Eq. (9) may be regarded as the practical implementation of a non-parametric FDT (npFDT) that uses the methods of non-parametric statistics.*

Remark 5 *Note further that, while* $\mathbf{\Lambda}(0) = \mathbf{I}$ *, it is not necessarily the case that* $\hat{\mathbf{\Lambda}}(0) = \mathbf{I}$*. This suggests the alternative estimator*

$$\hat{\hat{\mathbf{\Lambda}}}(s\Delta t) = \hat{\mathbf{\Lambda}}(0)^{-1}\hat{\mathbf{\Lambda}}(s\Delta t) \tag{10}$$

which is constrained to satisfy $\hat{\hat{\mathbf{\Lambda}}}(0) = \mathbf{I}$*, with a correspondingly defined*

$$\hat{\hat{\mathbf{L}}} = \hat{\mathbf{\Lambda}}(0)^{-1}\hat{\mathbf{L}}. \tag{11}$$

As a result of this condition, we expect to get rid of a potential bias, and this is the final estimator we will keep in the following.

Finally, it's important to note that due to the complexity of this estimation procedure we **lack of true theoretical guaranties and quantification on the convergence of our estimator**. This is why our approach will be mostly practical, forced to rely on heuristics.

1.2.2 • Choice of bandwidth parameter h

Let's now discuss the role of the hyperparameter h, since it will be important in the following experiments.

For the practical application of Eq. (9) picking an optimal value of the bandwidth parameter h is crucial. Some insight into how to make this choice may be obtained by considering the PDF estimation procedure (6). The variance and bias of the kernel PDF estimator may be approximated, using Taylor's theorem assuming bounded and continuous second derivatives (Silverman 1986 [9]), as

$$\begin{aligned} \text{var}[\hat{\rho}(\mathbf{x})] &\approx \frac{\beta}{nh^d}\rho(\mathbf{x}), \\ \text{bias}[\hat{\rho}(\mathbf{x})] &= \langle \hat{\rho}(\mathbf{x}) - \rho(\mathbf{x}) \rangle \\ &\approx \frac{1}{2}h^2 \alpha_{ij}\chi_{ij}, \end{aligned} \tag{12}$$

where

$$\chi_{ij}(\mathbf{x}) = \frac{\partial^2 \rho(\mathbf{x})}{\partial x_i \partial x_j} \tag{13}$$

$$\alpha_{ij} = \int y_i y_j K(\mathbf{y}) d\mathbf{y} \tag{14}$$

and

$$\beta = \int K(\mathbf{y})^2 d\mathbf{y} \tag{15}$$

In Eqs. (12)-(14) standard suffix notation is used and the summation convention is used in Eq. (12). So, for large h the estimate has a large bias but small variance, but for small h the estimate has a small bias but a large variance. When choosing h there is, therefore, a tradeoff between an incorrect answer (large bias) and uncertain answer (large variance). The variance is also a function of n, the number of data points, with larger n implying a more certain estimate of the PDF.

One approach to choosing an optimal value for h is to minimize the mean square difference between the true and estimated PDF, averaged over all $\mathbf{x}$. According to [9], the optimal value of h is then given by

$$h_{opt}^{d+4} = \frac{d\beta}{n}\left[\alpha_{ij}\alpha_{kl}\int \frac{\partial^2 \rho(\mathbf{x})}{\partial x_i \partial x_j}\frac{\partial^2 \rho(\mathbf{x})}{\partial x_k \partial x_l} d\mathbf{x}\right]^{-1} \tag{16}$$

where again the summation convention is applied. Note that h_{opt} reduces with n at the rate $n^{-1/(d+4)}$, with precise details depending on the density being estimated (Silverman 1986 [9]).

N.B: Fig. 8 illustrates perfectly what we presented above in the case of a uni-dimensional linear model.

1.3 Tackling one part of the problem

1.3.1 • Remaining issues

We have therefore laid down the first theoretical bricks essential for our final objective: estimate the climate sensitivity via the application of the npFDT with satellite data.

Still many questions and steps remain unclear at this point :

1. How the linearity approximation holds for the true climate system?
2. How many samples are required to estimate such PDFs?
3. Recent GCMs (and so the true climate system) have millions of degrees of freedom. So, there is obviously a need for a dimension reduction to avoid the curse of dimensionality. Then, how to apply/adapt the npFDT on such dimension-reduced systems?
4. Likewise, even for such dimension-reduced systems (with, let's say, a hundred dimensions), how the relatively high dimensionality of the systems considered will play a role in the quality of the estimation? Namely, how bad will behave the estimator while increasing the dimension?
5. Since we plan to work with relatively recent satellite observations, we know that the climate system considered will be out of energy balance (because of the *greenhouse effect*). So one can wonder what will be the implication of observing system under disequilibrium?
6. Regarding the observations, there is also a need to determine the minimum necessary and sufficient sampling frequency (could monthly be sufficient?).
7. Finally, we will have to estimate TCR with npFDT across a multi-climate-model ensemble to test the procedure.

1.3.2 • Objectives of my internship

Many of the previous questions have to be addressed by expert climate scientists. For the period of my internship, I decided to focus on the 4^{th} question regarding the role of the dimensionality in the estimation.

Indeed, many of the applications involving the estimation of multivariate densities so far focus on two or three-dimensional problems. Furthermore, the persistent interest among practitioners is contrasted by a relative decline of methodological contributions in the last two decades.

A probable reason is the prevalence of the *curse of dimensionality*: due to sparseness of the data, non-parametric density estimators converge more slowly to the true density as dimension increases. Put differently, the number of observations required for sufficiently accurate estimates grows excessively with the dimension. As a result, there is very little benefit from the ever-growing sample sizes in modern data. Section 7.2 in [8] illustrates this phenomenon for a kernel density estimator when the standard Gaussian is the target density: to achieve an accuracy comparable to $n = 50$ observations in one dimension, more than $n = 10^6$ observations are required in ten dimensions. More generally, Stone [10] proved that any estimator $\widehat{f}$ that is consistent for the class of p times continuously differentiable d-dimensional density functions converges at a rate of at most $n^{-p/(2p+d)}$.

One can wonder which impact this fact will have on the quality of our estimator for multivariate systems, even for simple models and distributions. Thus, to address this question, I decided to focus on the simplest models we can think of : linear models. Namely, models of the form :

$$\frac{d\mathbf{x}}{dt} = \mathbf{B}\mathbf{x} + \xi$$

with $\mathbf{B}$ a matrix and ξ a white noise.

Indeed, it has the advantage that the FDT holds perfectly (with no restrictions) on these models due to linearity (see [3]). A major goal will then be to observe the behavior of the estimator while increasing the dimension of the linear models considered. Obviously, we expect that the performance of the estimator will decrease due to the emergence of the *curse of dimensionality*. The interesting question will be to quantify this deterioration precisely in performance.

But first, since the original code of Fenwick Cooper in his npFDT paper [3] is written in C, my first objective will be to re-implement his method in a modern language like *python*. My code will have to be as effective as possible to avoid the decrease in computational performance due to the use of that high-level interpreted language. Also, making the code as modular as possible will allow me to accomplish another objective: try a particular KDE (with a module in *python*) named *fastKDE* [6] instead of a traditional KDE with a Gaussian kernel as done by Cooper.

I will first test my implementation of the procedure on some models presented in [3] to see if I can reproduce its results.

2
IMPLEMENTING THE METHOD

2.1 The algorithm

In order to reproduce the results of [3] and also to generalize to other models (like linear models in d dimensions with $d > 3$, or some simple climate models), we have completely redesigned the structure of the code. All is written in *python*, and the code is vectorized and parallelized as much as possible to compensate the slow computational performance of such high-level interpreted language compared to C. We use the library *numpy* extensively in all the code.

Thanks to the Lawrence Berkeley National Laboratory (LBNL), all the simulations are run with the supercomputer *Cori* (NERSC) which allows us to take advantage of parallelization.

The structure of the algorithm is the following :

1. First, we simulate a trajectory of the dynamical system of n points by using a discretization scheme. For stochastic linear models we use a *Euler-Maruyama* method (see 2.3.1 for details). These simulated points of the trajectory $(X_i)_{1\leq i\leq n}$ are our "dataset".

2. To avoid computational bottlenecks, we pre-compute what we called the PDF-dependent terms (the terms under brackets in Eq. (8) or (3), namely $(\frac{\nabla\rho}{\rho})$) at the beginning of the algorithm. Looking at the equation (8), we see that, for each specific j, we have a sum over all i of terms which only depends on one X_j. This allows us to parallelize this part, since all j are treated independently.

3. Then, we compute $\hat{\mathbf{\Lambda}}(s\Delta t)$ as in Eq. (3), for each value of s. All $\hat{\mathbf{\Lambda}}(s\Delta t)$ terms are independent with respect to s, so once again we parallelize this part.

4. Finally, we approximate the integral with the *trapezium rule* as in (9), with r being equal to $T/\Delta t$, with Δt the sampling time interval and T the upper limit in the truncation of the integral.

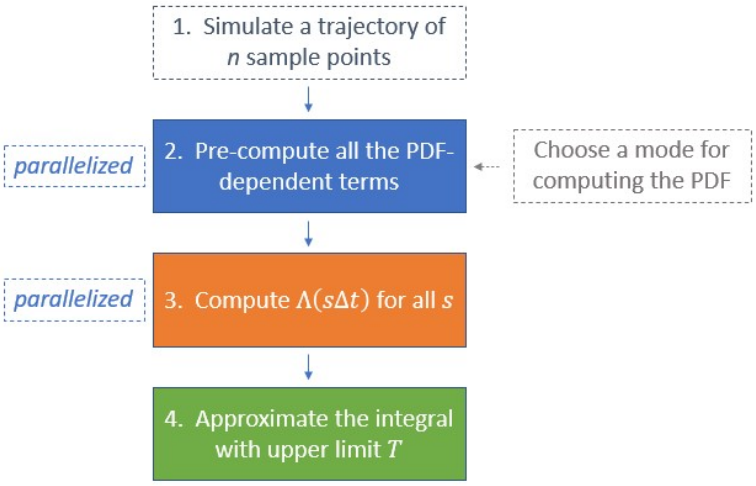

Figure 1: Structure of the code

We can repeat the whole procedure over K iterations to average the results and estimate the standard deviation of the estimator.

2.2 Presentation of the different modes

In order to compare different methods of estimating the PDF, the code is articulated in 4 modes :

- The 'gaussian' mode: aimed to reproduce the quasi-Gaussian form of the FDT
- The 'cooper' mode: aimed to reproduce the way F. Cooper implemented the non-parametric FDT in [3]. The main issue is that the algorithm is in n^2, which makes it not very usable in practice for $n > 10^6$.
- The 'subcooper' mode: our tentative to get rid of the n^2 order of calculations in 'cooper', while achieving similar results.
- The 'fastKDE' mode: the idea is to compute the pdf with $fastKDE$. This is the approach that we initially wanted to take for the future.

Basically, the only thing that differs in these modes is how to estimate the PDF and its derivative (part 1. of the algorithm). The rest of the algorithm is unchanged.

2.2.1 • The 'gaussian' mode: the quasi-Gaussian FDT

This mode is made in order to compare the *npFDT* with the *quasi-Gaussian FDT*.
As mentioned, the key question is how to compute the PDF term in Eq 3. Namely, the term under

bracket in

$$\mathbf{\Lambda}(\tau) := \left\langle \mathbf{x}(\tau) \left[\frac{\nabla \rho(\mathbf{x}(0))}{\rho(\mathbf{x}(0))} \right]^{\mathrm{T}} \right\rangle \tag{17}$$

Replacing the ensemble mean $\langle . \rangle$ by a time average over m steps of the trajectory, we get the estimator :

$$\hat{\mathbf{\Lambda}}(s\Delta t) = \frac{1}{m} \sum_{j=1}^{m} \mathbf{X}_{j+s} \left[\frac{\nabla \rho(\mathbf{X}_j)}{\rho(\mathbf{X}_j)} \right]^{T} \tag{18}$$

This is our starting point for all the 4 modes.
Finally, assuming that $\mathbf{X}$ as Gaussian density, we obtain :

$$\hat{\mathbf{\Lambda}}(s\Delta t) = \frac{1}{m} \sum_{j=1}^{m} -\mathbf{X}_{j+s} (\mathbf{C}(0)^{-1} (\mathbf{X}_j - \mu))^{T} \tag{19}$$

Where $\mathbf{C}(0)^{-1}$ is the inverse covariance matrix of $\mathbf{X}$, and μ its mean. Note that it is consistent with Eq. (5) if we assume $\mu = 0$ without loss of generality.

We evaluate the covariance matrix and the mean of $\mathbf{X}$ over m time steps of the trajectory. After this step, we do a simple matrix inversion and multiplication.

2.2.2 • Mode 'cooper': the classical non-parametric FDT

This is the approach proposed by Cooper in the original paper [3].

As a reminder, the matrix $\mathbf{\Lambda}$ is approximated as follows :

$$\hat{\mathbf{\Lambda}}(s\Delta t) = \frac{1}{m} \sum_{j=1}^{m} \mathbf{X}_{j+s} \left[\frac{\sum_{i=1}^{n} (\mathbf{X}_j - \mathbf{X}_i)^{\mathrm{T}} N(\mathbf{X}_j; \mathbf{X}_i, h)}{h^2 \sum_{i=1}^{n} N(\mathbf{X}_j; \mathbf{X}_i, h)} \right], \tag{20}$$

where N is a Gaussian kernel.

As one can imagine, the main issue is that the algorithm is in $\boldsymbol{n^2}$ (since $m = O(n)$) due to the double for-loop, which makes it not very usable in practice for $n > 10^6$.

2.2.3 • The 'subcooper' mode: a tentative to speed up 'cooper' bellow n^2

The intuition is the following : we probably need many points to evaluate ρ with a kernel density estimator, but we may not need to evaluate $\rho(\mathbf{x})$ at every $\mathbf{x}$.

Indeed, one can imagine that evaluating ρ only on a relevant sub-sample of points is sufficient. Thus, we could replace the value $\rho(\mathbf{x}_j)$ for each query point $\mathbf{x}_j$ by $\rho(\mathbf{x}_j^{sub})$ with $\mathbf{x}_j^{sub}$ being the nearest (in terms of euclidean distance) point to $\mathbf{x}_j$ in the sub-sample. In other words, this is simply choosing a sub-sample on which to compute the PDF and perform an interpolation for setting the value of the PDF for each point. The interpolation here is indeed a *Nearest-neighbor interpolation*, but other choices of interpolation could have been made.

Many things have still to be discussed such as the number of points in the sub-sample n_{sub} or the way to choose a relevant sub-sample. In practice, we have implemented the mode such as specifying

n_{sub} directly in the name of the mod (example : 'subcooper1000' has $n_{sub} = 1000$). In most applications, we chose $n_{sub} = 1000$, but this number is still a subject of work, in particular in order to scale up to multiple dimensions.

Moreover, we chose to **choose the sub-sample randomly (uniformly)** in all the points of the trajectory, assuming that for n_{sub} large enough we would explore the phase space relatively well for simple densities and thus this would give us a sufficiently representative sample of the distribution.

The calculation of the PDF-dependent terms is thereby made in three steps :

1. First, we choose randomly the sub-sample of length n_{sub} in the whole trajectory of length n.
2. Then, we compute the PDF-dependent terms for that sub-sample exactly as for the mode 'cooper' (but with evidently fewer samples - shape of the sub-sample : (n_{sub}, d) - cost : $\mathcal{O}(n_{sub}^2)$).
3. Finally, we compute the PDF-dependent terms for the m points of the trajectory with for each point $\mathbf{x_j}$ the value associated to the nearest (in terms of euclidean distance) point to $\mathbf{x_j}$ in the sub-sample. (What we obtain is a vector of shape : (m, d)) - cost : $\mathcal{O}(n_{sub}m)$).

The total computational cost is then in $\boldsymbol{\mathcal{O}(n_{sub}m)}$ which can be much better than previously ($\mathcal{O}(m^2)$). However, although the heuristic is quite simple, we have at this stage no theoretical guarantee on the performance of this estimator. It should therefore be used with caution.

In this mode, $\hat{\mathbf{\Lambda}}$ could then be expressed as follows :

$$\hat{\mathbf{\Lambda}}(s\Delta t) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{X}_{j+s} \left[\frac{\sum_{i=1}^{n} \left(V(\mathbf{X}^{sub}, \mathbf{X}_j) - \mathbf{X}_i \right)^{\mathrm{T}} N\left(V(\mathbf{X}^{sub}, \mathbf{X}_j); \mathbf{X}_i, h \right)}{h^2 \sum_{i=1}^{n} N\left(V(\mathbf{X}^{sub}, \mathbf{X}_j); \mathbf{X}_i, h \right)} \right], \tag{21}$$

with $V(\mathbf{X}^{sub}, \mathbf{X}_j)$ being the closest point to $\mathbf{X}_j$ in terms of euclidean distance in the sub-sample $\mathbf{X}^{sub}$ (of shape (n_{sub}, d)).

N.B : The choice of the distance (e.g. *euclidean* in our case) can still be discussed.

2.2.4 • The 'fastKDE' mode: The approach we wanted to take

As mentioned previously, the idea here is to compute the PDF with the module $fastKDE$ in python. This method has been developed by Travis O'Brien (see [6]) at the *Lawrence Berkeley National Laboratory* (LBNL). As indicated by its name, this algorithm is really fast compared to traditional KDE in small dimensions but has to be tested in higher dimensions. This is the approach we initially wanted to adopt for our final goal.

As explained previously, for a given kernel, KDEs require careful selection of the hyperparameter h, called the bandwidth parameter. As one might see in the Eq. (16), the optimal value of this parameter depends on true PDF itself.
To face this problem, Bernacchia and Pigolotti ([1]), developed a self-consistent method to choose the whole shape of the optimal kernel to use, by an iterative procedure after a Fourier transform of the KDE.
fastKDE is the generalization to multivariate dynamical systems of the original idea of Bernacchia

and Pigolotti. This *python* module also takes advantage of fast Fourier transform (FFT) algorithms, making it much faster than the original algorithm of [1].

The module estimates the PDF on a subset of values, which is a uniform grid dependent on the distribution and evaluated during the algorithm. The dimensions of this grid have to be specified with the constraint that in each dimension the number of points is a power of 2 plus 1 (e.g 33, 65, 129, ...). This constraint is directly due to the FFT algorithm used. Finally, *fastKDE* return the PDF values on the grid, and the grid itself. For instance, if we choose the number of points in each dimensions to be 129 for a 3D model, the estimated PDF and the associated grid will be vectors of shape (129,129,129).

According to [6], the algorithm scales as $\mathcal{O}\left(n \cdot q^d + M^d \cdot \log\left(M^d\right)\right)$, where q is the size of the convolution kernel used in the FFT algorithm, n is the number of samples, and M^d is the total number of grid points on which the KDE is estimated. Indeed, we expect a very bad behavior (namely *exponential*) of *fastKDE* with respect to the number of dimensions d.

Since in Eq. (18) we need to estimate $\frac{\nabla\rho}{\rho}$ for each point $\mathbf{X}_j$ and the algorithm only provide the value of ρ on the grid, we therefore need to interpolate. We chose once again to use a Nearest-neighbor interpolation, with this time a different distance. Namely, we take the $||.||_\infty$ distance. This choice is motivated by the fact that the grid is uniform so a choice that appeared natural was to take for each coordinate of a point $\mathbf{X}_j$, the closest coordinate in the grid.

One formula for $\hat{\mathbf{\Lambda}}$ could then be :

$$\hat{\mathbf{\Lambda}}(s\Delta t) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{X}_{j+s}\left[\frac{\nabla\hat{\rho}(V_\infty(Grid, \mathbf{X}_j))}{\hat{\rho}(V_\infty(Grid, \mathbf{X}_j))}\right]^T, \tag{22}$$

with $V_\infty(Grid, \mathbf{X}_j)$ being the closest point to $\mathbf{X}_j$ in terms of the distance $||.||_\infty$ in the grid calculated by *fastKDE*.

N.B: In practice, we compute the gradient of the estimated PDF with *numpy.gradient()*. That's a choice to be eventually discussed.

2.2.5 • Comparison of the computation times of the different modes

In order to quantify the difference of performance of the four previous modes, we took the liberty of comparing their computation time in a relevant example.

The model used is a 1D linear model (see 2.3.1). All the simulations are run on the supercomputer Cori (NERSC).

We took the following simulation parameters :
$\boldsymbol{n = 2\text{x}10^6}$
$m = n$
$\Delta t = 0.01$
$T = 10$
$h = 0.01$ for the 'cooper' and 'subcooper1000' mods

Averaging on 10 iterations, we finally obtain the following mean computation times (in seconds) :

	cooper	subcooper1000	gaussian	fastKDE
pdf	$\sim 7h$	91.0	**35.5**	50.6
after pdf	12.1	11.9	12.7	**11.8**
Total	$\sim 7h$	102.9	**48.2**	62.4

Unsurprisingly, the faster method is the *'gaussian'* mode, i.e. the *quasi-Gaussian FDT*, since the algorithm is just a covariance estimation followed by a matrix inversion and multiplication.
We note a impressive gain switching from *'cooper'* to *'subcooper1000'*, which will be very useful in the following. Finally, we note that the computation times after the estimation of the PDF, are nearly the same for all modes provided that the algorithm is exactly the same after that step.

2.3 Reproduce the results of the original npFDT paper

In order to test the new implementation of the algorithm, let's first compare our results with the results of F. Cooper on two simple linear models.

In the following we present briefly these two models.

2.3.1 • First model: a 1D-linear Model

$$\frac{dx}{dt} = -0.5x + \xi + f(t) \tag{23}$$

With f being a perturbation (we chose $f(t) = 1$ as in [3]), and ξ a Gaussian white noise ($\xi \sim \mathcal{N}(0, \sigma^2)$).

As said before, this case is particularly interesting because it is precisely a case where : 1) the FDT holds perfectly. 2) The Gaussian approximation of the FDT is not anymore an approximation, but is exact. Thus, the Gaussian-FDT should perform better than the non-parametric FDT here.

We note that with a constant f (we take $f = 1$ in the following), this model is a 1D **Ornstein-Uhlenbeck** process. As a reminder, a Ornstein-Uhlenbeck (O.U) process x_t is defined by the following stochastic differential equation:

$$dx_t = -\theta x_t dt + \sigma dW_t \tag{24}$$

where $\theta > 0$ and $\sigma > 0$ are parameters and W_t denotes the Wiener process. An additional drift term is sometimes added:

$$dx_t = \theta\left(\mu - x_t\right) dt + \sigma dW_t \tag{25}$$

where μ is a constant.

The Ornstein-Uhlenbeck process is sometimes also written as a Langevin equation of the form

$$\frac{dx_t}{dt} = -\theta x_t + \sigma\eta(t) \tag{26}$$

where $\eta(t)$, also known as white noise, stands in for the supposed "derivative" dW_t/dt of the Wiener process (although this is strictly formal). Thus, Eq. (23) is just the Langevin equation of a O.U process having $\theta = 1/2$ and $\mu = 2$ (if $f = 1$).

Conditioned on a particular value of x_0, the mean is

$$\mathbb{E}(x_t) = x_0 e^{-\theta t} + \mu\left(1 - e^{-\theta t}\right) \tag{27}$$

and the covariance is

$$\operatorname{cov}(x_s, x_t) = \frac{\sigma^2}{2\theta}\left(e^{-\theta|t-s|} - e^{-\theta(t+s)}\right). \tag{28}$$

For the stationary (unconditioned) process, the mean of x_t is μ, and the covariance of x_s and x_t is $\frac{\sigma^2}{2\theta}e^{-\theta|t-s|}$. Note that we can show further that the invariant measures of the systems (24) and (25) are respectively $\mathcal{N}(0, \frac{\sigma^2}{2\theta})$ and $\mathcal{N}(\mu, \frac{\sigma^2}{2\theta})$.

Therefore, with $f = 1$, the equilibrium state of the disturbed system should converge around the value 2 (so $\langle\delta\mathbf{x}\rangle = 2$). More precisely, the equilibrium distribution is $\mathcal{N}(2, \sigma^2)$. Similarly, the equilibrium distribution of the undisturbed system ($f = 0$) is $\mathcal{N}(0, \sigma^2)$.

Besides, to simulate the trajectory on discrete time steps, we choose a classical way of doing it using a *Euler–Maruyama* scheme :

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{B}\mathbf{x}(t)\Delta t + \mathbf{l}\boldsymbol{\xi}\sqrt{\Delta t} + \mathbf{f}\Delta t \tag{29}$$

in which $\boldsymbol{\xi}$ is a Gaussian distributed random vector with covariance $\mathbf{Q}$. We choose to restrict ourselves to the case $\mathbf{Q} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Knowing that we can compute the explicit solution, we do not need such a numerical scheme here, but we decide to keep it because other models that are not necessarily solvable will then be tested.

Finally, let's make a quick recap of all the parameters we introduced so far :

- n the number of samples in the simulation of a trajectory of the undisturbed system.
- m the number of samples in the time average. We choose $m = n$.
- Δt the discretization time step (trajectory and integral). We choose $\Delta t = 0.01$ as in [3].
- σ the standard deviation of the white noise ("the amplitude of the noise").
- T the upper limit of the truncation of the integral (3).

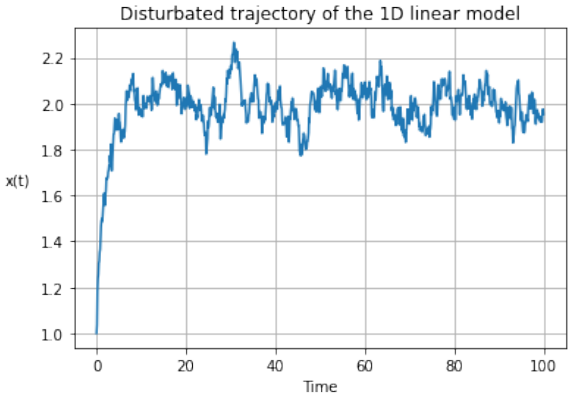

Figure 2: Example of a trajectory of the disturbed system ($f = 1$) for the model (23). Parameters : $n = 10^4$, $\Delta t = 0.01$, $\sigma = 0.1$. After it reaches the equilibrium value of 2, the system oscillate around this value.

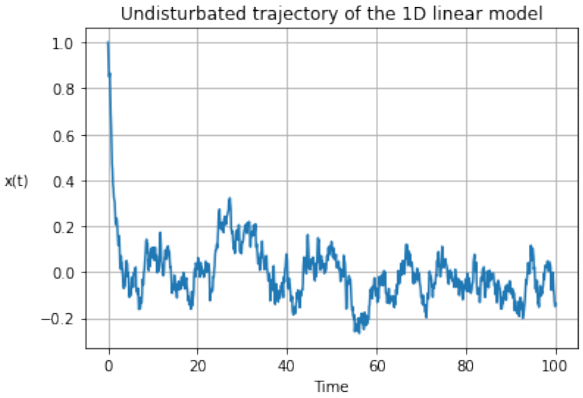

Figure 3: Example of a trajectory of the undisturbed system ($f = 0$) for the model (23). Parameters : $n = 10^4$, $\Delta t = 0.01$, $\sigma = 0.1$. After it reaches the equilibrium value of 0, the system oscillate around this value.

2.3.2 • Second model: a 3D-linear Model

To test for the first time the behavior of our estimator on a multivariate system, we now consider a three-dimensional linear model as in [3].

The model is the following :

$$\frac{d\mathbf{x}}{dt} = \mathbf{B}\mathbf{x} + \mathbf{f}(t) + \xi \tag{30}$$

with $\mathbf{B} = \begin{pmatrix} -6.8634 & 4.0196 & 2.0789 \\ 2.1943 & -92.5446 & 4.1803 \\ 4.6435 & 5.6232 & -77.5484 \end{pmatrix} \times 10^{-3}$ chosen randomly subject to the constraint that the system is stable (see 3.2 for more details), and $\xi \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3)$.

The constant forcing $\mathbf{f}$ was taken to be

$$\mathbf{f}(t) = \mathbf{f} = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} 3.3143 \\ 0.2447 \\ 1.0769 \end{pmatrix} \times 10^{-2}.$$

The same results on *Ornstein-Uhlenbeck* processes can still be applied. The theoretical target response can thus be computed :

$$\langle \delta x \rangle = \begin{pmatrix} 5.0644 \\ 0.1670 \\ 0.4542 \end{pmatrix}.$$

2.3.3 • First results

Our first results on the 1D model (2.3.1) were quite encouraging, with a clear convergence around the target value of 2. However, a really high number of samples (to the order 10^6) was required to obtain a reasonably low variance (of order 10^{-2}). With the same parameters σ, T and Δt, Cooper ([3]) was able to obtain the same variance but with a significantly fewer number of samples ($n = 6,55 \times 10^4$).

Before trying to reproduce some figures of [3] and comparing the different modes, we had to see the influence of some key simulations parameters such as the amplitude of the noise σ or the upper limit of the integral T. The next two graphs (4 and 5) shows such influences for the 1D linear model. In these figures, we only display the results of the 'gaussian' mode, but the same results were obtained for the three other modes.

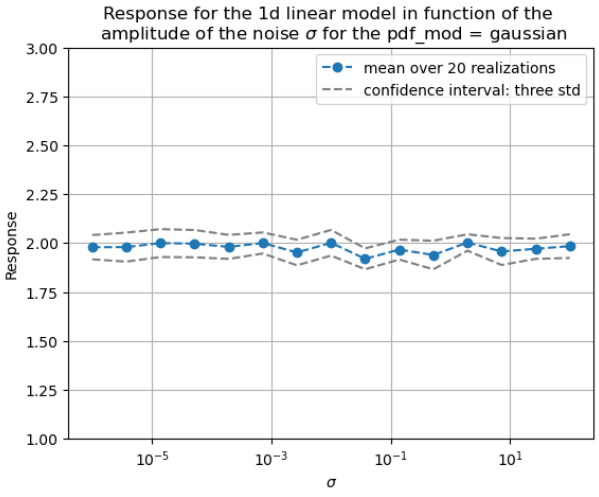

Figure 4: Estimator in function of the amplitude of the noise σ for the pdf-mod = 'gaussian', for the 1D linear model. As explained in 2.3.1, the target value is 2. Parameters : $K = 20$, $n = 2 \times 10^6$, $m = n$, $\Delta t = 0.01$, $T = 10$. We see that σ has no influence on the quality of the estimator.

The first graph (Fig. 4) demonstrates that the amplitude of the noise σ demonstrates no influences neither on the estimator nor on its standard deviation, so it can't explain our no variance problem. On

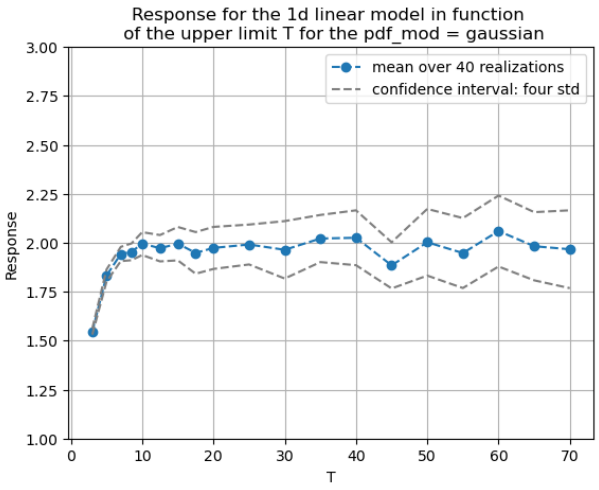

Figure 5: Estimator in function of the upper limit T for the pdf-mod = 'gaussian', for the 1D linear model. Parameters : $K = 40$, $n = 2 \times 10^6$, $m = n$, $\Delta t = 0.01$, $\sigma = 0.1$. This second graph shows that the estimator increase significantly with T. This was at first quite disturbing because our estimator is supposed to get better as T increases (since the theoretical formula takes $T \to \infty$).

the contrary, Fig. 5 shows a clear influence of T : the variance of the estimator increase significantly with T. This result seems quite surprising knowing that our estimator is supposed to get better as T increases (indeed the theoretical formula (Eq. 3) takes $T \to \infty$).

After investigating, it seems that the problem is due to the function we integrate in Eq. (9), namely $\hat{\mathbf{\Lambda}}$, or more precisely the bias-corrected matrix $\hat{\hat{\mathbf{\Lambda}}}$ (see Remark 5). As seen in Fig. 6, after the function decreases exponentially with $s\Delta t$ to zero (according to Eq. (28)), it remains some random noise asymptotically. Summing more and more random noise by increasing the upper limit of the integral appear to increase the variance.

In the following we present how we overcame this issue by adding a correction in the sampling method, and how we successively reproduced some results of Cooper [3] for both the 1D and 3D linear models. We also compare the four modes of estimating the PDF on the 1D linear model.

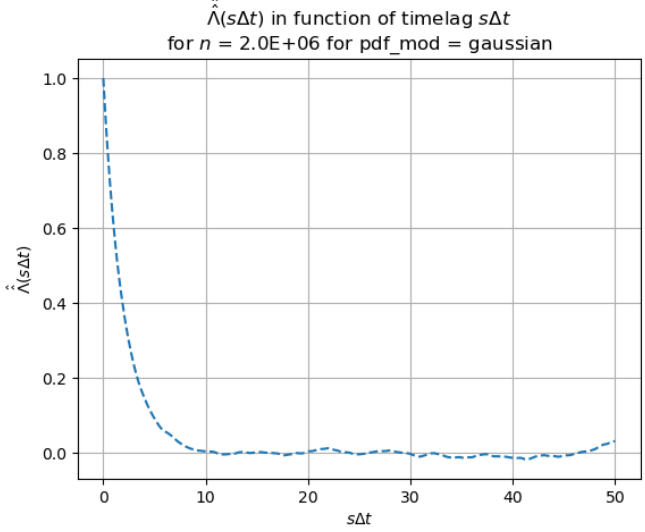

Figure 6: $\hat{\hat{\mathbf{\Lambda}}}$ in function $s\Delta t$ for the pdf-mod = 'gaussian', for the 1D linear model. Parameters : $K = 1$, $n = 2 \times 10^6$, $m = n$, $\Delta t = 0.01$, $\sigma = 0.1$. This graph is proportional to the autocorrelation plot of the data (because the true density is here Gaussian, see Eq. (5)). We observe that the convergence to zero is not perfect, since random noise persists.

2.3.4 • Results after the sub-sampling correction

We note that Eq. (8) assumes that all sample points are used in the estimation of $\nabla\rho/\rho$ (i.e., the summations in the numerator and denominator of the bracketed expression are over all sample points). But given that it is computationally expensive to compute the double sum appearing in Eq. (8), it may be optimal only to use a sub-sample for summation in the expression within braces. Another way of saying this is that, as written, Eq. (8) assumes that the time intervals for which the lag correlations are evaluated are multiples of the time intervals of sampling used to estimate $\nabla\rho/\rho$. But these time intervals could be multiples of some fraction of the latter sampling interval.

Indeed, since introducing such sub-sampling also results in having less points to integrate in the final integral, this would significantly reduce the variance (for equal number of samples n). As we will see, this would greatly reduce the computation time, as less samples are used in both the PDF estimation and the integral, while not reducing the accuracy of the estimation. Thus, our solution is the following : 1) We simulate the trajectory as before with $\Delta t = 0.01$, 2) We only keep one point out of 100. Thus, the new time interval between two samples X_i and X_{i+1} is now $\Delta t' = 100\Delta t = 1$.

We can see the benefits of our correction in Fig. 7, in which we tried to reproduce the first results of [3] for the 3D linear model, for the quasi-Gaussian FDT (the *'gaussian'* mode). In Fig. 10, we will also see that we obtain a much more smooth curve for the estimation of $\mathbf{\Lambda}$ after the sub-sampling correction (compared to Fig. 6). After that, all the results shown are made with the sub-sampling correction.

Then, we tried to reproduce another important result of Cooper [3], showing the influence of the bandwidth parameter h in the case where we do not make any correction of the bias (contrary to Remark 5). The results are shown in Fig. 8. Once again our results are coherent with those of Cooper

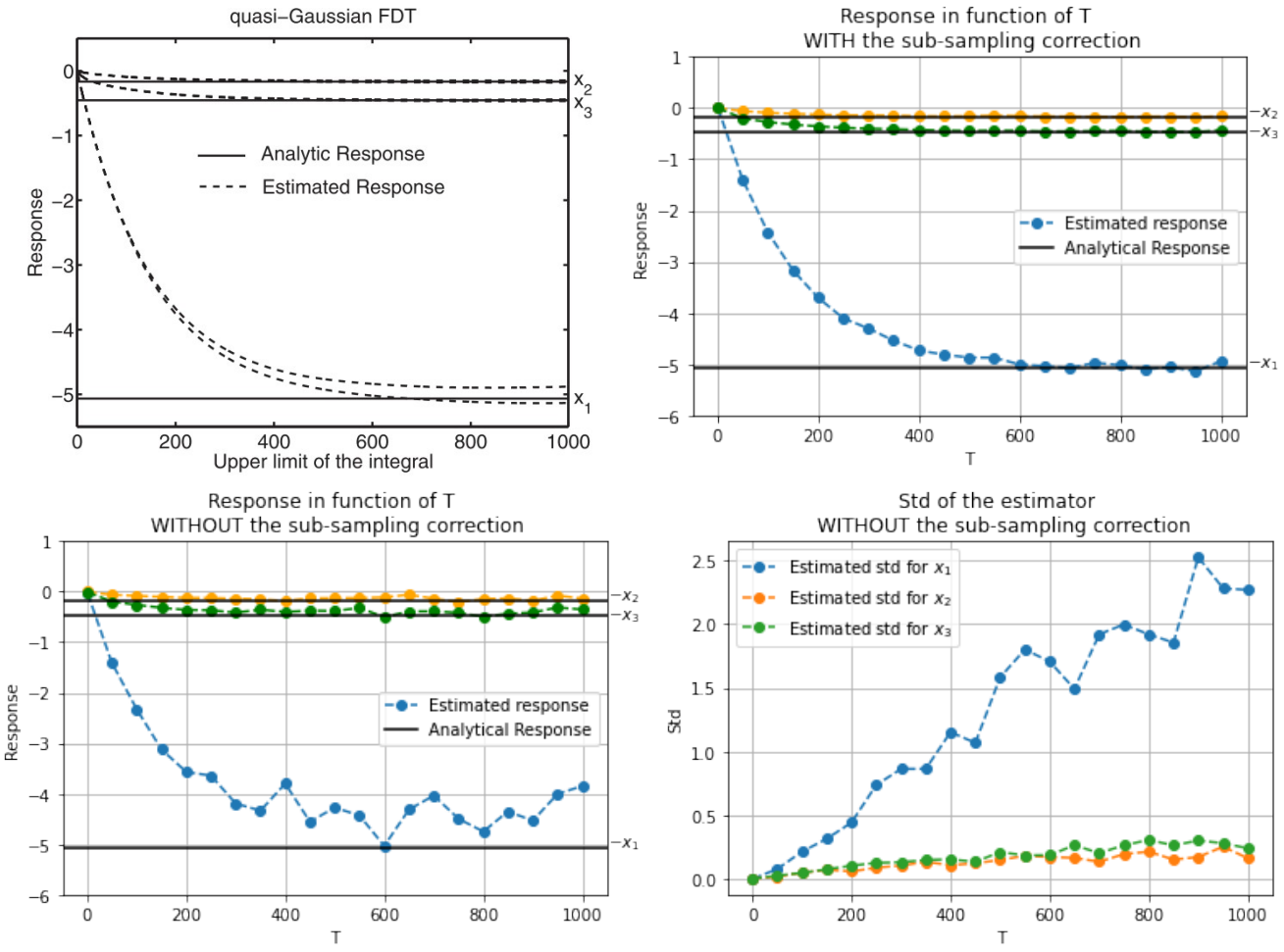

Figure 7: Estimator and its standard deviation in function of the upper limit T for the pdf-mod = 'gaussian', for the 3D linear model. Upper left : result of [3]. Upper right : our results after the sub-sampling correction. Lower left: our results before the sub-sampling correction. Lower right: Standard deviation of the estimator in each coordinate, before the sub-sampling correction. Parameters : $n = 1 \times 10^6$, $m = n$, $K = 20$, $\Delta t = 0.01$ without sub-sampling, $\Delta t = 1$ with sub-sampling, $\sigma = 0.1$. The upper figures show that we succeed in reproducing the results of [3] after the introduction of the sub-sampling correction. We see on the lower figures that without the sub-sampling correction the relative standard deviation of the estimator is increasing greatly with T.

[3]. After that, all the results shown are made with the bias correction.

Finally, we compared our four modes for computing the PDF (namely *'gaussian'*, *'cooper'*, *'sub-cooper1000'* and *'fastKDE'*) still on the simplest model possible : the 1D linear model. What we show is that these different modes have quite similar performance and behavior with respect to the number of samples n used for the estimation. This is quite promising for the rest of the project, since we don't lose too much in accuracy by replacing the Gaussian density by its non-parametric estimation. The results are shown in Fig. 9.

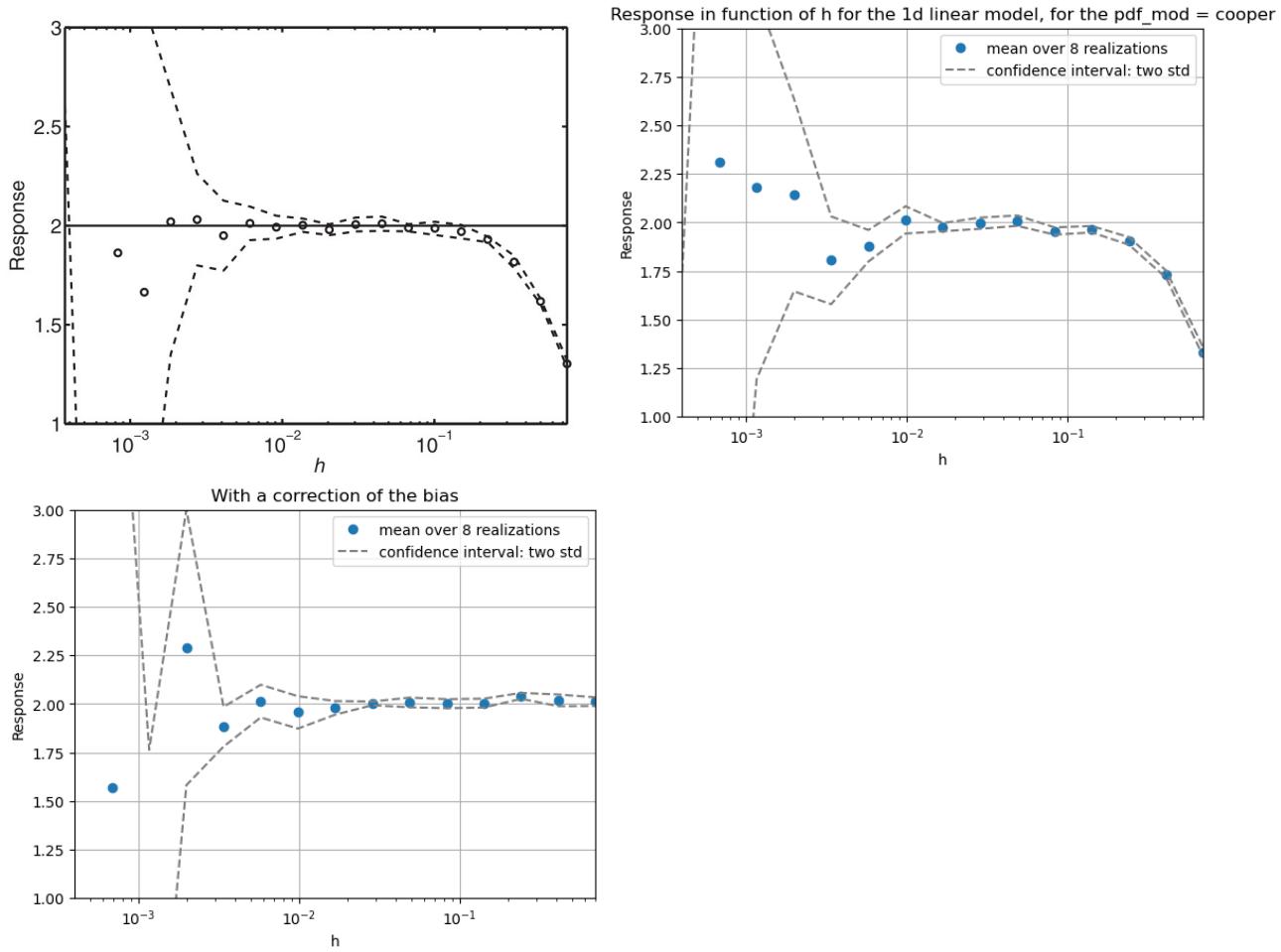

Figure 8: Estimator in function of the bandwidth parameter h for the pdf_mod = 'cooper', for the 1D linear model. Upper left : result of [3], without the bias correction. Upper right : my results, without the bias correction. Lower left: my results, with the bias correction. Parameters : $K = 20$, $n = 6,55 \times 10^4$, $m = n$, $\Delta t = 1$, $T = 10$, $\sigma = 0.1$. The upper figures show that we successfully reproduced the result of [3]. This illustrates the famous bias-variance trade-off : for small values of h we observe a large variance and in theory small bias, and for larger values of h we clearly note the apparition of a bias while the variance is decreasing. Thus, there is a range of optimal values of h that we have to identify carefully (here between 10^{-2} and 10^{-1}). This result is coherent with 1.2.2. The lower figure shows that, as expected, the bias related to large values of h disappear with the correction of the bias, allowing to choose h less carefully provided that it is large enough.

In Fig. 9, we also see that with our approximation of the non-parametric FDT - the *'subcooper'* mode - the potential reduction of performance of the estimator is not detectable for the 1D linear model. This is a good sign for the relevance of our heuristic.

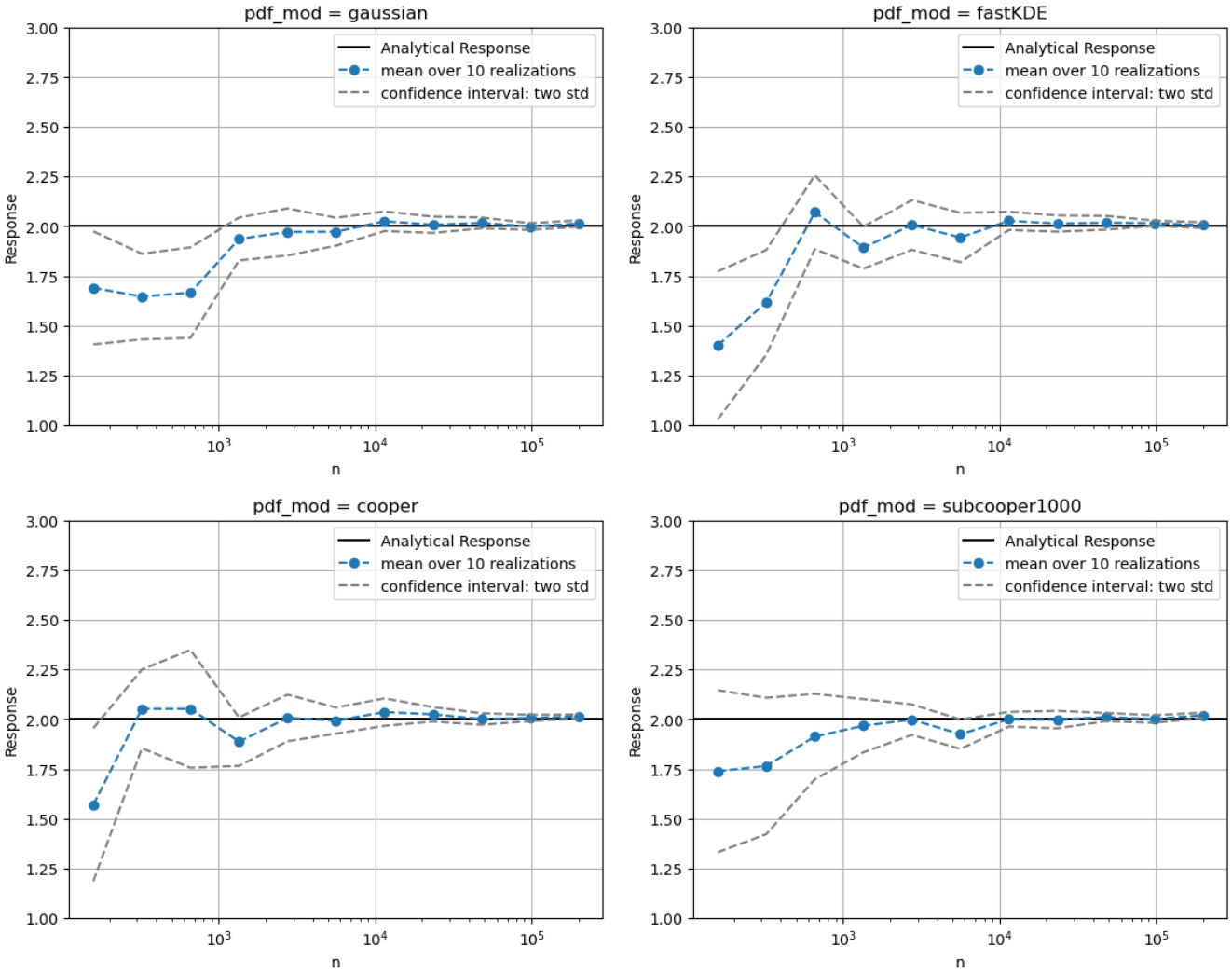

Figure 9: Mean estimated response in function of n for all the pdf_mod, for the 1D linear model. Parameters : $K = 10$, $m = n$, $\Delta t = 1$, $T = 10$, $\sigma = 0.1$, $h = 0.1$ (for the modes 'cooper' and 'subcooper1000' with the bias correction). It shows that these four modes have a similar behavior and performance with respect to n.

3
THE DIMENSIONALITY TEST

Since we now have some guaranties that our algorithm is working (at least on linear models), we can focus on our main objective i.e observe the behavior of the estimator while increasing the dimension of the dynamical systems. As mentioned, we will restrict our study to linear models.

Let's first consider a general definition of multivariate O.U processes.

3.1 Ornstein-Uhlenbeck process in multi-dimension

A multi-dimensional version of the Ornstein-Uhlenbeck process, denoted by the d-dimensional vector $\mathbf{x}_t$, can be defined from

$$d\mathbf{x}_t = -\beta \mathbf{x}_t dt + \sigma d\mathbf{W}_t \tag{31}$$

where $\mathbf{W}_t$ is an d-dimensional Wiener process, and β and σ are constant $d \times d$ matrices.
The solution is

$$\mathbf{x}_t = e^{-\beta t}\mathbf{x}_0 + \int_0^t e^{-\beta(t-t')}\sigma d\mathbf{W}_{t'} \tag{32}$$

and the mean is

$$\mathrm{E}\left(\mathbf{x}_t\right) = e^{-\beta t}\mathrm{E}\left(\mathbf{x}_0\right) \tag{33}$$

These expressions make use of the matrix exponential.

3.2 Generate multi-dimensional linear models

We try to study several linear models with several dimensions. So we need a way to generate linear models in order to compare them meaningfully.
The models are still of the form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{B}\mathbf{x} + \mathbf{f}(t) + \xi \tag{34}$$

with $\mathbf{B}$ a $d \times d$ matrix chosen to guarantee the stability of the system, ξ a d-dimensional Gaussian white noise ($\xi \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$), and $\mathbf{f}$ a constant perturbation (i.e a d-dimensional vector). As previously, Eq. (34) is the Langevin representation of a multivariate O.U process.

Two questions arise at this point :

1. What means that the dynamical is "stable", and how to guarantee it?
2. How to choose $\mathbf{B}$ (and $\mathbf{f}$) for any number of dimensions in order to compare efficiently all the models?

In our case, we say that our linear dynamical system is *stable* if and only if

$$\lim_{t \to +\infty} \mathbf{E}(\mathbf{x}(t)) \text{ exists and is finite.}$$

One way of ensuring that is asking for $\mathbf{B}$ to be diagonalizable and that all is eigenvalues are strictly negative. Indeed, considering the Eq. (33), we note that for such $\mathbf{B}$ (and considering $\mathbf{f} = 0$ for the sake of simplicity) we have :

$$\mathrm{E}(\mathbf{x}(\mathbf{t})) = e^{\mathbf{B}t}\mathrm{E}\left(\mathbf{x}_0\right) = \mathbf{P}\, diag(e^{\lambda_1 t}, ..., e^{\lambda_d t})\, \mathbf{P}^{-1}\mathrm{E}\left(\mathbf{x}_0\right) \xrightarrow[t \to +\infty]{} 0$$

with $\lambda_1, ..., \lambda_d < 0$ the eigenvalues of $\mathbf{B}$, for some $\mathbf{P} \in \mathrm{GL}_d(\mathbb{R})$.

For the second question, the answer is a bit more tricky. As for each of the models we have to set the value of the upper limit T in the truncation of the integral (9), we want a way to generate

models that "behave" more or less the same regarding the decay of the correlations (see Fig. 6 for an example). The 1D and the 3D model presented above, for instance, have very different decorrelation times, more precisely setting the value $T = 10$ is sufficient for the 1D model (see Fig. 6) while we have to go above the value 600 for the 3D model (see Fig. 7).

Hopefully, the Eq. (28) shows that the correlations decrease with time lag τ at the "rate" $e^{\mathbf{B}\tau}$. This fact suggests a way of controlling the generation of the systems in order that we can safely set the same value of the upper limit T for all models.

Finally, we want to generate models randomly so as to average the results on several different models for each dimension.

Thus, our algorithm for generating a linear model in dimension d is the following :

1. First, we control the decorrelation timescale (and so the required upper limit of the integral T) by choosing the eigenvalues of $\mathbf{B}$ near a certain value. More precisely, we want to keep the same behavior as model 2.3.1, so we generate $(\lambda_1, ..., \lambda_d)$, the eigenvalues of $\mathbf{B}$, near the value $-1/2$ according to the distribution $\mathcal{U}_{[-0.7,-0.3]^d}$.
2. Then, we explicitly generate the random matrix $\mathbf{B}$ with eigenvalues $(\lambda_1, ..., \lambda_d)$ according to the following algorithm:
   - We first generate a random matrix $\mathbf{A}$ (according to $\mathcal{N}(0, \mathbf{I}_{d\times d})$).
   - Then, we compute the QR-decomposition of $\mathbf{A}$: $\mathbf{A} = \mathbf{QR}$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is an upper triangular matrix.
   - Finally, we define $\mathbf{B}$ as $\mathbf{B} := \mathbf{Q}\, diag(\lambda_1, ..., \lambda_d)\, \mathbf{Q}^{-1}$
   - What we obtain is a negative-definite random matrix with eigenvalues $(\lambda_1, ..., \lambda_d)$.
3. Finally, we set the value of the perturbation $\mathbf{f}$ to $(1, ..., 1)$, in order to obtain responses for each coordinates near the value 2 (as in 2.3.1).

According to Fig. 10 (and since we obtain almost the exact same curve for all the models generated with our algorithm for any number of dimensions), we set the value of the upper limit of the integral T to the value 20 in the following simulations, without taking any risk of "forgetting" a non-zero part of the curve.

3.3 Result

We generate models up to the dimension 30, averaging the results on 5 different sets of linear models. After the dimension 30, the computational time gets too large for our purpose, but it can still be done if needed.

To evaluate the performance of our estimator we need to average the error on all the coordinates and all the different sets of models. We also need to normalize these quantities. Therefore, we define the two following metrics :

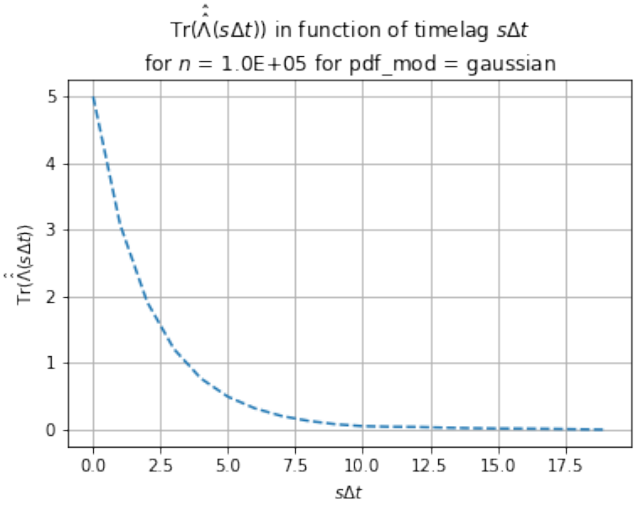

Figure 10: Trace of the estimated matrix $\hat{\hat{\mathbf{\Lambda}}}$ in function of $s\Delta t$, for a random linear model in dimension 5 generated with our algorithm. Parameters : $n = 1 \times 10^5$, $m = n$, $K = 1$, $\Delta t = 1$, $T = 20$, $\sigma = 0.1$, pdf-mod='gaussian'. We obtain almost the exact same curve for all the models generated with our algorithm for any number of dimensions. This allows us to set the value of the upper limit of the integral T to the value 20, without taking any risk of forgetting a non-zero part of the curve.

First, our error metric is here called *mean relative error* and defined as

$$\bar{\text{RE}} := \left\langle \left\langle \frac{|r_{th} - \bar{r}_{est}|}{|r_{th}|} \right\rangle_{dim} \right\rangle_{iter} \tag{35}$$

where r_{th} is the vector of theoretical response for each coordinate, $\bar{r}_{est}$ is the estimated response vector averaged on K realizations, $\langle . \rangle_{dim}$ denotes the mean over all the coordinates, and $\langle . \rangle_{iter}$ is the mean over all the n_{iter} iterations (i.e over the different sets of models).

Then, our metric to evaluate the standard deviation of our estimator is here called *mean relative standard deviation* and defined as

$$\bar{\text{RS}} := \left\langle \left\langle \frac{\bar{s_{td}}}{|r_{th}|} \right\rangle_{dim} \right\rangle_{iter} \tag{36}$$

where $\bar{s_{td}}$ is the estimated standard deviation vector (for each coordinate) computed on K realizations.

What we want is to precisely isolate the effect of the number of dimensions in the performance of the estimator. Therefore, we fix all other simulation parameters in the calculations. In particular, the simulations are made for a fixed number of sample n equal to 5×10^4.

We compared the three PDF-modes that are relatively fast, namely 'fastKDE', 'gaussian' and 'subcooper'. Regarding the selection of h for the mode 'subcooper', we have seen in 1.2.2 that the optimal value of the bandwidth parameter h_{opt} increase with d at the rate $n^{-(1/d^4)}$. This suggests taking larger values of h for larger dimensions in order for the variance to be low enough. Moreover, as suggested by Fig. 8 (lower figure), we take no risk in choosing very large values of h thanks to the

bias correction. Thus, we fixed the value of h at 2 in all the simulations.

We present our final results in Fig. 11.

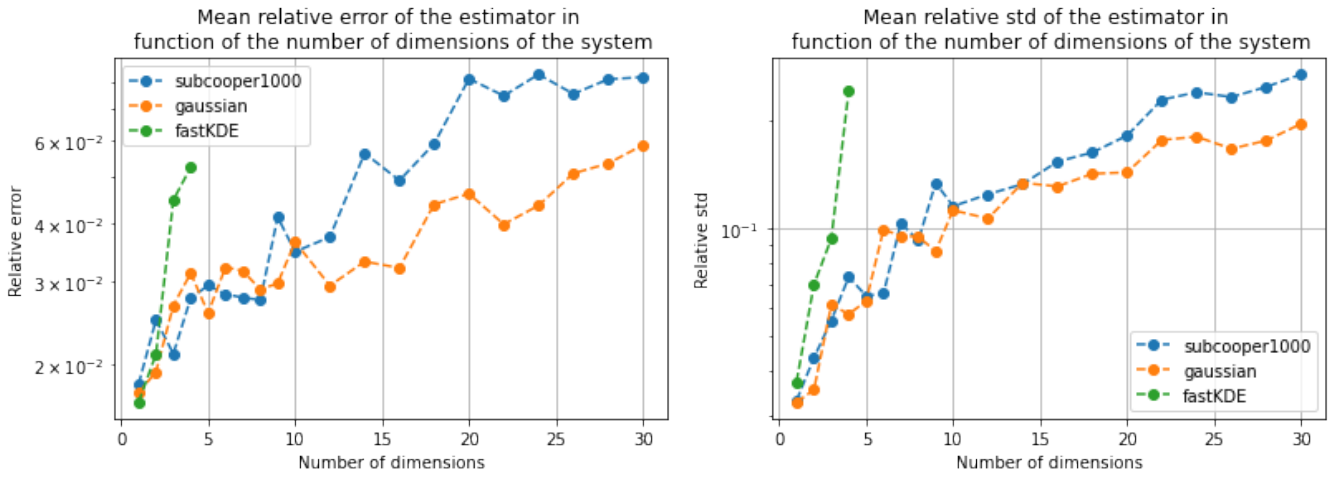

Figure 11: Mean relative error and standard deviation of the estimator in function of the number of dimensions of the system for the pdf modes 'fastKDE', 'subcooper1000' and 'gaussian'. Params : $n_{iter} = 5$, $n = 5 \times 10^4$, $m = n$, $K = 10$, $\Delta t = 1$, $T = 20$, $\sigma = 0.1$, $h = 2$.

What we can learn from these simulations for the 'subcooper' mode, is that the error increases by a factor 4 from dimension 1 to dimension 30 – which is not too problematic since it remains quite low – and the standard deviation of the estimator increase by a factor 10 from dimension 1 to dimension 30 – here it's a bit more problematic but manageable by increasing the number of samples for higher dimensions.

Also, the mode 'fastKDE' appeared to be useless after the dimension 4 as both the computation time and the error diverge for higher dimensions. This is consistent with the exponential dependency in d for the scaling of the algorithm, as presented in 2.2.4.

Finally, an important insight of theses curves is that we don't loose too much by replacing the Gaussian density (the 'gaussian' mode) by its non-parametric estimation with 'subcooper' since both the error and the variance of the estimator are not much bigger for 'subcooper' than for 'gaussian'. As a reminder, the 'gaussian' mode is here supposed to perform optimally because the true density is indeed Gaussian, but obviously the 'subcooper' mode will outperform the quasi-Gaussian FDT in any non-Gaussian case as shown in [3].

Remark 6 *These curves are made with the mode 'subcooper1000', but the accuracy of the estimator obviously increases if we take more points in the sub-sample (like 'subcooper1e4' or 'subcooper5e4' etc...).*

Thus, the mode 'subcooper' appeared to be the most serious candidate for our final objective of estimating the *equilibrium climate sensitivity* with satellites images.

4
CONCLUSION AND HORIZON

Throughout this exciting project, we addressed a small part of a vast enterprise. We were able to successfully reproduce the result of Fenwick Cooper [3] regarding the *non-parametric fluctuation-dissipation theorem* and to propose **a new framework** for the implementation of the algorithm. Moreover, we were able to quantify the performance degradation of the estimation procedure with dimension, restricting our study to linear models.

We were capable to extend our study up to the **dimension 30**, and we did not observe any dramatic performance degradation for such dimensions. We have also exhibited a potential candidate for conducting the next simulations: **the *'subcooper'* mode**. Beside, since in all likelihood we will be able to reduce the number of dimensions of the satellite images (that we planned to work with) down to 80 dimensions, we are rather optimistic regarding the relevance of our procedure to address this problem.

However, many questions remain open. Indeed, the next studies will have to address the question of *non-linearity* and *non-equilibrium*. One can for instance see the work of A. Sarracino [7] for a fluctuation-dissipation relation in non-equilibrium systems. Also, we will have to test our procedure on GCMs in order to know to what extent we could trust the predictions of the algorithm. Indeed, we will definitely need a way to precisely **quantify the uncertainty** of our estimate otherwise the whole procedure will be useless. Any theoretical progress will obviously be welcome.

Moreover, some additional tests on our work would be interesting to be carried out. For instance, a simple thing we could think of would be to change the distribution of the white noise (while keeping a linear model) to study *non-Gaussianity*. Also, other kernels, such as the *Epanechnikov* kernel suggested by F. Cooper, should be tested for the KDEs.
Finally, one could think of a method to automatically choose T, the upper limit of the integral in Eq. (3), for arbitrary distribution in multiple dimensions. In this extend, Travis O'Brien suggested using a method based on selecting a subset of contiguous hypervolumes of some above-threshold values as done in [6] for a different purpose. Any work in this direction could be interesting.

Finally, it is certain that the curse of dimensionality will eventually pose a real problem in the PDF estimation of high-dimensional dynamical systems (see the results of Stone in [10]). In this extend, any progress in the field of non-parametric high-dimensional PDF estimation would be extremely useful for our final goal.

LIST OF FIGURES

REFERENCES

[1] Alberto Bernacchia and Simone Pigolotti. Self-consistent method for density estimation. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 73:407, 06 2011.

[2] I. Cionni, G. Visconti, and F. Sassi. Fluctuation dissipation theorem in a general circulation model. *Geophysical Research Letters*, 31(9), 2004.

[3] Fenwick C Cooper and Peter H Haynes. Climate sensitivity via a nonparametric fluctuation–dissipation theorem. *Journal of the Atmospheric Sciences*, 68(5):937–953, 2011.

[4] Richard Goody, James Anderson, and Gerald North. Testing climate models: An approach. *Bulletin of the American Meteorological Society*, 79(11):2541–2549, 1998.

[5] C. E. Leith. Climate Response and Fluctuation Dissipation. *Journal of Atmospheric Sciences*, 32(10):2022–2026, October 1975.

[6] Travis A. O'Brien, Karthik Kashinath, Nicholas R. Cavanaugh, William D. Collins, and John P. O'Brien. A fast and objective multidimensional kernel density estimation method: fastkde. *Computational Statistics Data Analysis*, 101:148–160, 2016.

[7] A. Sarracino and A. Vulpiani. On the fluctuation-dissipation relation in non-equilibrium and non-hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(8):083132, aug 2019.

[8] David W Scott. The curse of dimensionality and dimension reduction. *Multivariate Density Estimation: Theory, Practice, and Visualization*, 1:217–40, 2008.

[9] B. W. Silverman. *Density estimation for statistics and data analysis / B.W. Silverman*. Chapman and Hall London ; New York, 1986.

[10] Charles J Stone. Optimal rates of convergence for nonparametric estimators. *The annals of Statistics*, pages 1348–1360, 1980.