Parameter identification through gradient flow on latent variables

Muriel Boulakia, Haibo Liu, Damiano Lombardi

Parameter identification through gradient flow on latent variables

Muriel Boulakia
Université Paris-Saclay, UVSQ, CNRS
Laboratoire de Mathématiques de Versailles, 78000, Versailles, France
muriel.boulakia@uvsq.fr

Haibo Liu
Sorbonne Université, Inria, CNRS
Laboratoire Jacques-Louis Lions (LJLL), Paris, France
NOTOCORD part of INSTEM
haibo.liu@instem.com

Damiano Lombardi
Sorbonne Université, Inria, CNRS
Laboratoire Jacques-Louis Lions (LJLL), Paris, France
damiano.lombardi@inria.fr

December 26, 2023

Abstract

In this article, we consider a system of parametric ODEs which involves unknown parameters and we seek to identify the values of the parameters associated to a given measurement. To do so, we place ourselves within the fairly usual framework that this single measurement is in fact taken from a population of data and we therefore want to take advantage of the statistical knowledge about the population to regularize the classical minimization problem associated to our identification problem. In the method that we propose and that we call the Latent Variable Gradient Flow method, the data set is represented by an autoencoder neural network which allows to associate to each element of the data set a latent variable. Then, introducing a non-linear mapping between the parameter space and the latent variable space allows to convexify the cost function and to demonstrate convergence properties. These properties are numerically illustrated with different tests on Van der Pol and FitzHugh-Nagumo models.

1 Introduction

In many studies of experimental sciences, such as chemistry, biology or environmental engineering, mathematical models are used to describe the behaviour of the dynamical systems. Those mathematical models consist of systems of ordinary differential equations (ODEs). For the most part, these ODEs contain parameters that are associated to physical, biological, or other properties of the system. To assess the relevance of the model, it is necessary to understand the role of these parameters and to set their values in order to generate signals that are close to reality. In addition to parameters, the system is completed by the data of the initial state whose values may be unknown or uncertain. When faced with an experimental observation, identifying the unknown parameters and the initial state to closely match the experimental results is a key step in the development and understanding of ODEs [32]. However, this identification problem which belongs to the class of "inverse problems" is a complex task, especially when the ODEs system consists of a large number of equations and parameters.

To define the context more precisely, let us introduce some notations which will be defined in more detail later on. Let a system be described by a mathematical model, in which the output u is affected by a certain number of parameters and by the initial conditions. Let us denote by θ the vector of these parameters and initial conditions. To simplify the formulation, in what follows, we will describe in general θ as a model parameters vector, even if it also includes the initial conditions. Then, we introduce the parameter-to-output map φ which associates to the vector θ the output u: $u = \varphi(\theta)$.

The general problem reads as follows: given some measurements u^* of the output, we try to estimate the model parameters θ^* such that $u^* = \varphi(\theta^*)$.

Classically, the estimation of model parameters ([1, 4]) is cast as an optimisation problem. In particular, a cost function is defined based on the data misfit and the parameter θ is estimated by minimising the discrepancy between the model observations $\varphi(\theta)$ and the actual measurement u^* of the system. This formulation often leads to a non-linear non-convex possibly high-dimensional optimisation problem. This could be dealt with in two ways:

1. Local (often gradient based) optimisation methods. These methods include Quasi-Newton methods [15, 19], the Gauss-Newton method (incorporating explicitly derived and efficiently computed sensitivity equations [6]), as well as other gradient-based methods [3]. The local gradient-based optimisation methods might encounter convergence issues when the initial guess is far from the true parameter values [25].

2. Global optimisation approach. For example, commonly utilized stochastic search algorithms include evolutionary computation, adaptive stochastic methods, and clustering methods. Other global optimisation methods are the genetic algorithms, as proposed for instance in [38] for the determination of rate constants in heterogeneous reaction systems, and collocation methods, as proposed for instance in [2] and [7]. In many cases, stochastic search algorithms are used to select good initial guesses for starting either a Quasi-Newton method or a gradient-based type minimisation. Global optimisation methods are often very expensive from a computational point of view and might become prohibitive when the number of parameters is large [22, 41].

Another way to try to solve the problem is purely based on data. Let us assume that, in certain situations, we have examples of pairs of parameters and observations, $\left\{\theta^{(i)}, u^{(i)}\right\}_{1\leq i\leq N_d}$. For example, the work in [18] used a supervised convolutional machine-learning method to learn the physical model parameters. The advantage of this kind of model is that no a priori knowledge about the system is required. The main disadvantage consists in the fact that we often need a large number of training pairs parameters-observations.

Recently, many efforts were made in order to try to exploit both the knowledge coming from the mathematical model and from sets of data, such as physics-informed neural networks (PINNs) mentioned in [16, 20, 24, 39], deep operator network (DeepONet) find in [26] and physics-informed DeepONet (PI-DeepONet) find in [37].

PINNs operate by taking time as input and by training the network to generate corresponding observations of the dynamic system. This training process leverages both the available data and the underlying dynamics of the system. When addressing inverse problems, the PINN's loss function integrates two components: the discrepancies between predicted observations and given data, and the residual of the differential equations, which depends on the parameters. Then, the network optimises the parameters of the network and estimates the parameters of the dynamical system. PINNs might suffer from the so-called spectral bias [31], which refers to the fact that it tends to prioritize learning the low-frequency part of the solution, not rendering properly multiple frequency scales. To overcome this issue, Fourier Features-Neural Networks (FF-NN) have been proposed to solve dynamical systems and perform inverse problems [33]. Another limitation of PINN-based methods is that they often require a large number of iterations and evaluations to converge to an accurate solution. This computational intensity is a drawback, especially when dealing with dynamic systems consisting of a large number of equations and parameters. In the continuity of PINN-based methods, the method that we propose in this paper seeks to leverage both available data and underlining dynamical systems. However, as we will see, our method provides an alternative method of PINN for which the computational demanding part corresponding to the training on the data set is done in a preliminary part.

About the data set, we focus here on a specific situation: we consider that we do not have data composed of pairs of parameters-observations but that we only have observations

$$\left\{u^{(i)}\right\}_{1\leq i\leq N_d}$$

consisting in a collection of measurements taken on a population or a set of experiments and we assume that our (novel) observation u^* comes from the same population. This situation, which is quite common in a number of realistic applications, is by far more difficult than the one in which pairs of parameters-observations are available (as we have no direct information about the parameters anymore).

In this paper, we will propose a new method of parameter estimation which relies on a regularisation of a classical parameter estimation by means of the data set. A representation of the data set by an autoencoder will allow to describe an element of the data set by a latent variable. Then, the method performs a non-linear mapping between the parameter space, where the cost function associated with the parameter estimation is non-convex, and the latent variable space, where the cost function is convex. Since our method can be described as a gradient flow for the latent variables of the autoencoder, it

will be called the Latent Variables Gradient Flow (LVGF) method.

In addition to the presentation and study of the LVGF method, this paper also presents a novel method to estimate the intrinsic dimension of a data set $U = \left\{u^{(i)}\right\}_{1\leq i\leq N_d}$. Identifying this intrinsic dimension is an important point in the implementation of our method since it allows to set the value of the latent dimension of the autoencoder trained on the data set. The proposed method will be based on the use of autoencoders and a criterion based on the approximation of the Lipschitz constant of the encoding and decoding maps. For an ODE system satisfying observability and identifiability properties, this intrinsic dimension is related to the dimension of the set of parameters which are varying in the population.

The structure of the work is the following: in Section 2, we will state the parameter identification problem, present our parameter estimation method which we call the LVGF method, and study the convergence properties of this method. Moving on to Section 3, we will show some theoretical results and the methodology of using an autoencoder to perform intrinsic dimension estimation which is a part of the parameter estimation problem. Section 4 will be dedicated to demonstrating the performance of our proposed intrinsic dimension estimation method using two example dynamical systems. At last, in Section 5, we will assess the LVGF method's ability to solve the parameter estimation problem on the dynamical systems presented in the first section and compare its performance with the classical minimization method.

2 Parameter estimation

2.1 Problem statement

Let us consider a system of ODEs:

$$\begin{cases} \dot{x}(t) = f(x(t), \zeta), \, \forall \, t \in (0, T), \\ x(0) = x_0 \end{cases} \tag{2.1}$$

where $x : [0, T] \to \mathbb{R}^N$ is a real vector valued function, $\zeta \in \mathbb{R}^q$ represents a vector of constant parameters and $x_0 \in \mathbb{R}^N$ is the initial state. We assume that, for every $x_0 \in \mathbb{R}^N$ and $\zeta \in \mathbb{R}^q$, system (2.1) admits a unique solution x in $C^1([0, T])^N$.

For this system, we consider that the parameter values and the initial condition are unknown or uncertain and we are interested in the simultaneous identification of parameters and initial data. We denote by $\theta = (x_0, \zeta) \in \mathbb{R}^m$ with $m = N + q$ the vector that we want to identify and look for θ in a subset of $\mathbb{R}^m$ that we denote by Θ.

The overall goal of our study is to identify the value of $\theta \in \Theta \subset \mathbb{R}^m$, by exploiting some discrete measurements that we denote by $u \in \mathbb{R}^n$ of the solution x of system (2.1). We denote by H the observation operator that models the measurement procedure:

$$H : C^1([0, T])^N \to \mathbb{R}^n. \tag{2.2}$$

For the sake of simplicity in the presentation, we assume that H only gives one scalar information (for instance, one component of the vector valued function x) discretized in time.

In addition, we define the map which associates the measurement to the unknown parameters and initial data:

$$\begin{array}{rcl} \varphi : \mathbb{R}^m & \to & \mathbb{R}^n \\ \theta & \to & H(x(\theta, \cdot)) \end{array} \tag{2.3}$$

where $x(\theta, \cdot)$ is the solution of system (2.1) associated to $\theta = (x_0, \zeta)$.

The classical deterministic identification problem can be formulated in the following way: we assume that a datum $u^* \in \mathbb{R}^n$ is given and we would like to identify $\theta^* \in \Theta \subset \mathbb{R}^m$ such that

$$\varphi(\theta^*) = u^*. \tag{2.4}$$

Most of the time, this problem is numerically studied by reformulating it as an optimisation problem. In a classical way, the functional to minimize can be for instance of the form:

$$J(\theta) = \|\varphi(\theta) - u^*\|_n^2 \tag{2.5}$$

where $\|\cdot\|_n$ corresponds to the Euclidean norm in $\mathbb{R}^n$.

Such a problem is known to be cumbersome in a number of situations, due to the fact that it is usually non-linear and non-convex. We will give a practical example of those situations in the end of this section.

In the present work, we consider that, in addition to the map φ and the datum u^*, we have access to a set of $N_d \in \mathbb{N}^*$ data, denoted $U = \left\{u^{(i)}\right\}_{1 \leq i \leq N_d} \in \mathbb{R}^n$ which u^* is a part. We assume that these data come from a population (of individuals, or experiments), characterised by a certain (unknown) distribution of parameters θ. This setting is justified by the fact that it is indeed common to have access not to a single measurement but to a large number of data. Therefore, our objective is to study how we can take advantage of this whole set of measurements U in the identification of parameters associated to a single measurement u^* and if adding information coming from the knowledge of the set U can allow to construct a regularisation for the parameter estimation problem. This idea will be the starting point of the LVGF method presented in 2.3.1.

In what follows, we will illustrate our approach on two ODE systems.

First, we will consider Van der Pol model, a model proposed in the 1920s by Van der Pol [34] to represent the oscillations in vacuum tube circuits. It is given by the following system:

$$\begin{cases} \dot{x} = v, \\ \dot{v} = \mu(1 - x^2)v - x, \\ (x, v)(0) = (x_0, v_0). \end{cases} \tag{2.6}$$

In this equation, $\mu > 0$ is a fixed parameter which reflects the degree of nonlinearity of this system, x_0 is the initial position and v_0 is the initial velocity. For this model, we assume that we measure the state variable x on the whole time interval and we are interested in the identification of $\theta = (x_0, v_0, \mu)$.

Second, we will consider FitzHugh-Nagumo model which describes the dynamics of a spiking neuron. It is given by

$$\begin{cases} \dot{v} = v - \frac{v^3}{3} - w + I_{ext}, \\ \dot{w} = \frac{1}{\tau}(v + a - bw), \\ (v, w)(0) = (v_0, w_0) \end{cases} \tag{2.7}$$

where v corresponds to the membrane potential, w to the recovery variable and I_{ext} to a stimulus current. For this model, we assume that we only observe the state variable v and that the value of I_{ext} is known. For this problem, we are interested in the identification of $\theta = (v_0, w_0, a, b, \tau)$.

2.2 Identifiability and observability of the inverse problem

Before testing numerical methods for the resolution of the identification problem (2.4), it is essential to ensure the uniqueness of a solution to (2.4) or, in other words, the injectivity of φ in Θ. Not having this type of theoretical property can in fact explain numerical convergence issues independently of the numerical method chosen.

Moreover, contrary to Tykhonov methods that can overcome a lack of uniqueness in the identification problem through the add of a prior, in our case the only information that we add to regularize our minimization problem comes from the knowledge of a data set. Thus, the underdetermined nature of the problem would remain despite the regularization process.

Since θ is composed both of model parameters and initial data, this identification property is related to two distinct notions in inverse problems: first, the notion of identifiability which refers to the fact that model parameters are uniquely determined by the output and, second, the notion of observability which refers to the fact that the initial conditions are uniquely determined by the output.

Let us notice that the identifiability and observability of the problem introduced above can be reduced to the observability of the following augmented system (as presented for instance in [13] and [27]):

$$\begin{cases} \dot{X}(t) = F(X(t)), \, \forall \, t \in (0, T), \\ X(0) = (x_0, \zeta) = \theta \end{cases} \tag{2.8}$$

where θ is taken in Θ, $X = (x, \zeta) : [0, T] \to \mathbb{R}^m$ is a regular function and F is given by

$$\begin{array}{rcl} F : \mathbb{R}^m & \to & \mathbb{R}^m \\ (x, \zeta) & \to & (f(x, \zeta), 0). \end{array} \tag{2.9}$$

Contrary to the observation operator given by (2.2) and adapted to the numerical framework where the solution is discretized in time, we consider in the theoretical framework that we have access to measurements at each time in $[0, T]$ and we denote by

$$h : C^1([0, T])^N \to C^1([0, T]) \tag{2.10}$$

the observation map which associates the time-dependent measurement to the solution X of the ODE system (2.8).

Numerous works address the question of identifiability and observability for ODE systems and there is a wide variety of methods for doing so. About ODEs modelling, we can quote [12], [35], [5] and [36] among many references. Following these last two references, we are interested by the structural observability property for system (2.8) (or by the structural identifiability and observability property for system (2.1)) which means that, for almost all initial conditions $X_1(0)$ and $X_2(0)$ in Θ

$$h(X_1) = h(X_2) \text{ in } [0,T] \Rightarrow X_1(0) = X_2(0).$$

For the two ODE models presented in the previous section, the we have the following results:

Proposition 2.1. *If we measure the variable x, Van der Pol model (2.6) is structurally observable and identifiable in the variables (x_0, v_0, μ), in the sense that (x_0, v_0, μ) is uniquely determined from the measurement of the function x in $[0,T]$.*

Proposition 2.2. *If we measure the variable v, FitzHugh-Nagumo model (2.7) is structurally observable and identifiable in the variables (v_0, w_0, a, b, τ), in the sense that (v_0, w_0, a, b, τ) is uniquely determined from the measurement of the function v in $[0,T]$.*

The proofs of these results are presented in Appendix A.

2.3 Latent Variables Gradient Flow (LVGF) for parameter estimation

2.3.1 Description of the LVGF method

The first ingredient of the proposed LVGF method relies on the construction of an autoencoder approximating the data set U. It consists in an encoder map $\Psi : \mathbb{R}^n \to \mathbb{R}^p$ which compresses an input data $u \in U$ into a latent variable $\alpha = \Psi(u) \in \mathbb{R}^p$ with $p < n$ and a decoder map $\Psi^\dagger : \mathbb{R}^p \to \mathbb{R}^n$ which recreates an approximation of the input data from the latent variable. Thus, in the data set U, which we can think about as a sampling of an embedded submanifold $\mathcal{U} \subset \mathbb{R}^n$, $\Psi^\dagger \circ \Psi(u)$ is a good approximation of u. The set-up of the autoencoder will be specified in Section 3.1. Let us notice that the choice of the hyperparameter p is related to the notion of intrinsic dimension of $\mathcal{U}$ and, by using some recent results in approximation theory, the estimation of the intrinsic dimension will be discussed in Section 3.2.

The rationale behind the introduction of the autoencoder is the following: the latent variables $\alpha \in \mathbb{R}^p$ are, at the same time, observable and related to the parameters $\theta \in \Theta$. Indeed, given $u^* = \varphi(\theta^*) \in \mathcal{U}$, thanks to the encoder Ψ, we can easily compute the latent variable α^* associated with it, $\alpha^* = \Psi(u^*)$. Furthermore, we are aware of the relationship between θ^* and α^* which is defined as follows:

$$\Psi \circ \varphi(\theta^*) = \alpha^*.$$

By leveraging these two observations, we can introduce a method in which, by trying to match the latent variables values α^*, we estimate θ^*.

The way in which we try to reach this goal is by introducing a gradient flow in the latent variable α, which motivates the method name: Latent Variable Gradient Flow (LVGF).

In what follows, we use the notation $g = \Psi \circ \varphi : \mathbb{R}^m \to \mathbb{R}^p$ and assume that

$$g \in C^{1,\beta}(\Theta), \text{ for some } \beta > 0 \text{ and } \operatorname{rank} \nabla g(\theta) = p, \forall\, \theta \in \Theta. \tag{2.11}$$

Under these hypotheses, we can consider the generalized inverse (also called the Moore-Penrose pseudoinverse [28]) of $\nabla g(\theta)$ defined for a matrix M, by $M^\dagger := M^T(MM^T)^{-1}$. Moreover, we assume that there exists a constant $\overline{C} > 0$ such that, for all $\theta \in \Theta$

$$\|\nabla g(\theta)^\dagger\| \leq \overline{C}. \tag{2.12}$$

The LVGF algorithm is based on the following iterative process: let a step $s > 0$ and an initial value $\theta_0 \in \Theta$ be given. We denote by α_0 the initial value associated to θ_0, that is $\alpha_0 = g(\theta_0)$. Then, we define the sequence $(\theta_k)_{k\in\mathbb{N}}$ iteratively by: for $k \in \mathbb{N}$

$$\left\{ \begin{array}{ll} \Delta\theta_k & = sM_k^\dagger(\alpha^* - \alpha_k) \text{ where } M_k = \nabla g(\theta_k) \\ \theta_{k+1} & = \theta_k + \Delta\theta_k \\ \alpha_{k+1} & = g(\theta_{k+1}). \end{array} \right. \tag{2.13}$$

By definition, $\Delta\theta_k$ is thus the unique vector of smallest norm solution in $\mathbb{R}^m$ of the system

$$M_k \Delta\theta_k = s(\alpha^* - \alpha_k).$$

2.3.2 Convergence of the LVGF method

In the following proposition, we prove a convergence result on the sequence (θ_k) in the specific case where the parameter space is the full space $\mathbb{R}^m$.

Proposition 2.3. *We assume that $\Theta = \mathbb{R}^m$ and that g satisfies the hypotheses (2.11)-(2.12). Then, for $s > 0$ small enough, the sequence $(\theta_k)_{k\in\mathbb{N}}$ defined by (2.13) converges to some $\tilde{\theta} \in \mathbb{R}^m$ which satisfies*

$$\alpha^* = g(\tilde{\theta}). \tag{2.14}$$

Moreover, there exists a constant $C > 0$ depending on $\alpha^* - \alpha_0$ and $\overline{C}$ such that

$$\|\tilde{\theta} - \theta_k\|_m \leq \frac{C}{s}\big(1 - \frac{s}{2}\big)^k$$

Proof. Let $k \in \mathbb{N}$ be given. According to (2.13), we have

$$\alpha^* - \alpha_{k+1} = \alpha^* - g(\theta_k + \Delta\theta_k).$$

Since $g \in C^{1,\beta}(\Theta)$, we can write

$$g(\theta_k + \Delta\theta_k) = g(\theta_k) + M_k\Delta\theta_k + \xi_k = \alpha_k + s(\alpha^* - \alpha_k) + \xi_k$$

where $\xi_k \in \mathbb{R}^p$ satisfies

$$\|\xi_k\|_p \leq \tilde{C}\|\Delta\theta_k\|_m^{1+\beta} \leq \tilde{C}\overline{C}^{1+\beta}s^{1+\beta}\|\alpha^* - \alpha_k\|_p^{1+\beta}$$

where we have used (2.12) and denoted by $\tilde{C}$ the Hölderian constant of ∇g. This implies that

$$\begin{aligned} \|\alpha^* - \alpha_{k+1}\|_p &= \|(1-s)(\alpha^* - \alpha_k) - \xi_k\|_p \\ &\leq (1-s)\|\alpha^* - \alpha_k\|_p + \tilde{C}\overline{C}^{1+\beta}s^{1+\beta}\|\alpha^* - \alpha_k\|_p^{1+\beta} \end{aligned}$$

Assume now that the step $s \in]0,1[$ is chosen such that

$$\tilde{C}\overline{C}^{1+\beta} s^{\beta} \|\alpha^* - \alpha_0\|_p^{\beta} \leq \frac{1}{2}.$$

Then, an argument by induction allows us to deduce from the previous inequality that, for all $k \in \mathbb{N}$

$$\|\alpha^* - \alpha_k\|_p \leq \|\alpha^* - \alpha_0\|_p.$$

Therefore, we deduce that, for all $k \in \mathbb{N}$

$$\|\alpha^* - \alpha_{k+1}\|_p \leq \big(1 - \frac{s}{2}\big)\|\alpha^* - \alpha_k\|_p.$$

So the sequence $(\alpha_k)_{k\in\mathbb{N}}$ linearly converges to α^* and

$$\|\alpha^* - \alpha_k\|_p \leq \big(1 - \frac{s}{2}\big)^k \|\alpha^* - \alpha_0\|_p.$$

This implies that the sequence $(\theta_k)_{k\in\mathbb{N}}$ satisfies

$$\|\theta_{k+1} - \theta_k\|_m = \|\Delta\theta_k\|_m \leq C\|\alpha^* - \alpha_k\|_p \leq C\big(1 - \frac{s}{2}\big)^k \|\alpha^* - \alpha_0\|_p.$$

In particular, since $(\theta_k)_{k\in N}$ is a Cauchy sequence, it converges to some $\tilde{\theta}$. Moreover, we have

$$\|\tilde{\theta} - \theta_k\|_m \leq \frac{2C}{s}\big(1 - \frac{s}{2}\big)^k \|\alpha^* - \alpha_0\|_p$$

At last, passing to the limit in the expression $\alpha_k = g(\theta_k)$, we deduce the formula (2.14). □

Corollary 2.4. *Under the same hypotheses as Proposition 2.3, we assume in addition that $\Psi : \mathcal{U} \to \mathbb{R}^p$ and $\varphi : \mathbb{R}^m \to \mathbb{R}^n$ are one-to-one and that $\tilde{\theta} \in \varphi^{-1}(\mathcal{U})$. Then, for $s > 0$ small enough, the sequence $(\theta_k)_{k\in\mathbb{N}}$ defined by (2.13) linearly converges to $\theta^* \in \mathbb{R}^m$.*

This corollary directly comes from the fact that, by definition of α^* and according to equation (2.14), the limit $\tilde{\theta}$ of the sequence $(\theta_k)_{k\in\mathbb{N}}$ satisfies:

$$\Psi \circ \varphi(\tilde{\theta}) = \Psi \circ \varphi(\theta^*).$$

Thus, under the hypotheses of the corollary, $\tilde{\theta} = \theta^*$.

Let us add a few comments on the additional assumptions of Corollary 2.4. First, we remark that the hypothesis that $\Psi : \mathcal{U} \to \mathbb{R}^p$ is one-to-one is naturally related to the encoding properties of this function for which a given latent variable can correspond to only one vector in $\mathcal{U}$. The hypothesis of injectivity of $\varphi : \mathbb{R}^m \to \mathbb{R}^n$ is also a natural assumption related to the identifiability and observability discussed in Section 2.2.

Since the algorithm given by equation (2.13) does not allow to ensure that the property $\tilde{\theta} \in \varphi^{-1}(\mathcal{U})$ is satisfied, in practice, we correct it by adding a projection procedure. Let us denote by $\mathcal{P} : \mathbb{R}^m \to \mathbb{R}^m$ a projection operator on $\varphi^{-1}(\mathcal{U})$. For a given $\theta_{-1} \in \mathbb{R}^m$, we compute $\theta_0 = \mathcal{P}(\theta_{-1}) \in \varphi^{-1}(\mathcal{U})$ and replace system (2.13) by

$$\left\{ \begin{array}{ll} \Delta\theta_k & = sM_k^{\dagger}(\alpha^* - \alpha_k) \text{ where } M_k = \nabla g(\theta_k) \\ \theta_{k+1} & = \mathcal{P}(\theta_k + \Delta\theta_k) \\ \alpha_{k+1} & = g(\theta_{k+1}). \end{array} \right. \tag{2.15}$$

Remark 2.5. *In most of the realistic situations, Θ is not the whole space $\mathbb{R}^m$ but a compact set in $\mathbb{R}^m$. The method can be written in a parametric domain (say, a box) containing this domain, and such that θ^* is an interior point of the domain. In the case in which the iterates are all interior points, the result presented holds true without any modification. If some iterates fall out of the domain, they can be reprojected into it. In this case, further investigations would be needed to determine the speed of convergence.*

Remark 2.6. *The LVGF method can be applied also in the case in which $\bar{p} = m$ and it has a remarkable interpretation. Let us assume that a parametric model is identifiable. Let us recall that $g \in C^{1,\beta}(\Theta)$. It follows that $M^\dagger = M^{-1}$, for all $\theta \in \Theta \subseteq \mathbb{R}^m$. Solving the problem with a classical cost function would amount to solve a non-linear, non-convex optimisation problem. The function $g = \psi \circ \varphi$ can be seen as a change of variables making it possible to cast the parameter estimation problem as the optimisation of the convex function $J(\alpha) = \|\alpha^* - \alpha\|_m^2$.*

Remark 2.7. *The LVGF method comes with a way to verify, a posteriori, whether the model solution $\varphi(\theta)$ does not belong to the population. This can be done by exploiting the decoder $\psi^\dagger$. Let us consider: $\|\psi^\dagger \circ \psi \circ \varphi(\theta) - \varphi(\theta)\|_n^2$. If this quantity is significantly larger than the error obtained on the validation set when training the autoencoder, it implies that the obtained solution is far from the ones belonging to the population.*

2.3.3 Numerical implementation

For the numerical implementation of this method, it is necessary to describe how the projection on $\varphi^{-1}(\mathcal{U})$ is defined and computed. By noticing that the elements θ which belong to $\varphi^{-1}(\mathcal{U})$ satisfy

$$\Psi^\dagger \circ \Psi \circ \varphi(\theta) - \varphi(\theta) = 0,$$

we introduce the functional

$$E(\theta) = ||\Psi^\dagger \circ \Psi \circ \varphi(\theta) - \varphi(\theta)||_n^2 \tag{2.16}$$

and define the projection $\mathcal{P}(\theta)$ as the solution of the minimisation of E starting from the initial data θ. The minimisation of the function E is achieved by using the gradient descent method.

In practice, to initialize the sequence $(\theta_k)_{k\in\mathbb{N}}$ with a θ_0 such that $\theta_0 \in \varphi^{-1}(\mathcal{U})$, the gradient descent method may fail if we start with a value far from the submanifold $\varphi^{-1}(\mathcal{U})$. So, we consider different initial values for θ_0, run for each initial value the gradient descent method and keep a θ_0 for which the optimisation procedure gives a value of E close to zero. In addition, we do not run the projection step at each iteration but only if θ_k begins to move away from $\varphi^{-1}(\mathcal{U})$ which is tested by comparing the value $E(\theta_k)$ to a threshold value. The details of the LVGF method are shown in Algorithm 1.

2.4 A numerical illustration of LVGF

In this section, a first numerical illustration of the LVGF method is proposed for the identification of a two-dimensional parameter vector. Moreover, our method is compared to a gradient descent method applied to the minimisation of the classical functional (2.5).

Algorithm 1 The LVGF algorithm

Input: N_{max} = maximum number of iterations; σ = tolerance for $||\alpha^* - \alpha||_p$; $S = [s_1, s_2, ..., s_n]_{1 \leq i \leq n}$
start with a random initial guess θ_0
if $E(\theta_0) > \epsilon$ **then**
  run the projection $\theta = \mathcal{P}(\theta_0)$
else
  $\theta = \theta_0$
end if
compute $\alpha = \Psi(\varphi(\theta))$
while $j \leq N_{max}$ and $||\alpha^* - \alpha||_p \geq \sigma$ **do**
  compute $M = [\nabla\Psi(\varphi(\theta))][\nabla\varphi(\theta)]$
  compute $\Delta\theta = [M]^\dagger(\alpha^* - \alpha)$
  select $1 \leq i \leq n$ the minimum point of $\{\|\alpha^* - \Psi(\varphi(\theta + s_i\Delta\theta))\|_p,\ 1 \leq i \leq n\}$
  update $\theta = \theta + s_i\Delta\theta$
  if $E(\theta) > \epsilon$ **then**
    run the projection $\theta = \mathcal{P}(\theta)$
  end if
  compute $\alpha = \Psi(\varphi(\theta))$
end while
return the value of the last iteration of θ

We consider Van der Pol model (2.6) and are interested by identifying $\theta = (v_0, \mu)$ whereas x_0 is assumed to be known and its value is given by $x_0 = 0.5$. We consider a measurement u^* defined by $u^* = \varphi(\theta^*)$ with $\theta^* = (0.5, 0.5)$ and our objective is to identify this value θ^*. In addition to u^*, we consider that we have access to a data set U (contain $N_d = 1200$ simulations) which has been generated in a preliminary step by varying the values of μ, randomly drawn from a uniform distribution in $[0.05, 2.0]$ and setting $v_0 = 0.5$. So in this case, the intrinsic dimension of the submanifold $\mathcal{U}$ is equal to 1. The latent dimension of the autoencoder trained on this data set is fixed to $p = 1$ (this dimension can also be rediscovered following the method presented in Section 3.2). We used AE1 (mentioned in Table 1) to approximate the data set U. There are 1000 samples used in the training set and 200 in the validation set. Moreover, the parameters given as input of Algorithm 1 have been fixed to $N_{max} = 2000$, $\sigma = 10^{-4}$ and $S = [0.0005, 0.0025, 0.0125, 0.0625]$.

In addition to the LVGF method, we have implemented the gradient descent method applied to the minimisation of the classical functional (2.5) in order to compare both methods. In Figure 1, we plot the contour of the classical cost function given by equation (2.5). We observe that the function is non convex and presents local minima in addition to the global minimum at $\theta^* = (0.5, 0.5)$ represented by a red sign "+". The different iterations for the classical gradient descent method applied to equation (2.5) and for the LVGF method have been also depicted starting with the same initialisation $\theta_0 = (0.8, 1.9)$.

It is interesting to observe that the gradient descent method converges as expected to a local minimiser whereas the LVGF method is able to go against the direction of the steepest slope in order to change valley and reach the global minimiser. Thus the information coming from the data set and added through the use of the latent variable

made it possible to correct the convexity defect of the classical functions and to identify the right value of the unknown parameters.

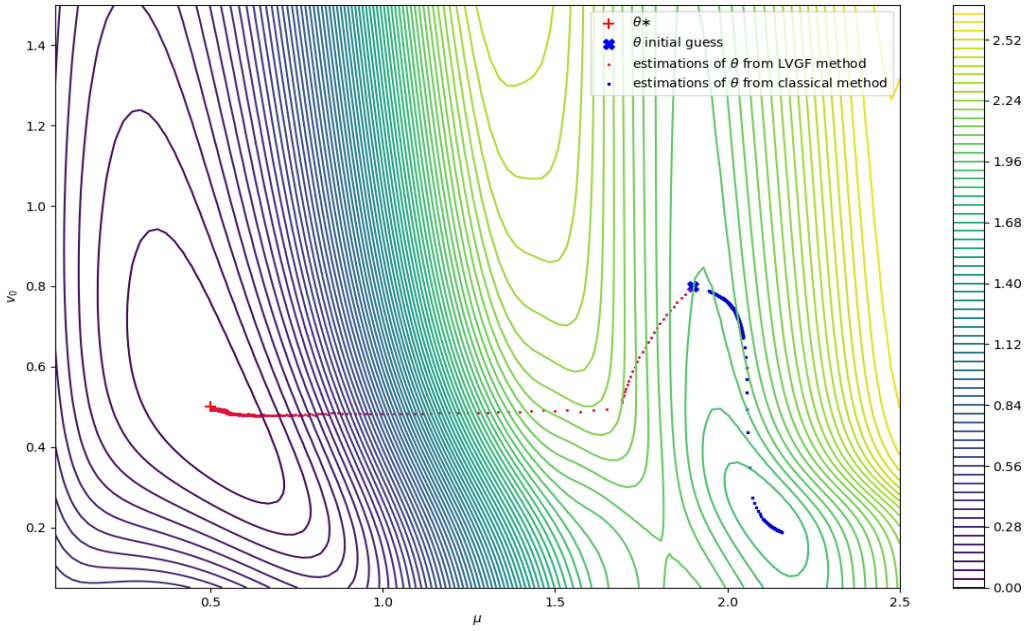


Figure 1: Contour plot of the classical function $J(\theta) = \|u^* - \varphi(\theta)\|_n^2$ with $\theta = (v_0, \mu)$: comparison between the iterations of the classical gradient descent method (blue dots) and the LVGF steps (red dots) with the same initial guess (blue cross). The value of the true parameter, which is the minimum of J is $\theta^* = (0.5, 0.5)$, marked with a red cross.

Further numerical tests to better assess the performances of LVGF will be presented in Section 5.

3 Lipschitz-stable autoencoders and intrinsic dimension

As presented in Section 2.3.1, our LVGF method relies on a description of the data set $U = \{u^{(i)}\}_{1 \leq i \leq N_d}$ by an autoencoder. These neural networks are dimensionality reduction algorithms which learn two functions called encoder and decoder. The encoder function associates to an input vector in $\mathbb{R}^n$ a vector called a code or a latent variable of smaller dimension p.

It is important to notice that the dimension p is an hyperparameter that has to be set before training the autoencoder even though, in general, the intrinsic dimension of a data set is unknown. That is why the first question that naturally arises in the representation of a data set by an autoencoder is to understand how to settle the dimension of the latent variable. In this section, we propose a criterion based on the notion of stable manifold width to estimate the intrinsic dimension of the data set. This criterion is a consequence

of the result given by Corollary 3.2 which highlights the fact that, if the dimension of the latent variable is smaller than the intrinsic dimension, the product of the Lipschitz constants of the encoder and decoder functions will blow up.

3.1 Presentation of the autoencoder

In this section, we describe the autoencoder neural network and give some details on the implementation of the autoencoder that we have considered in this work.

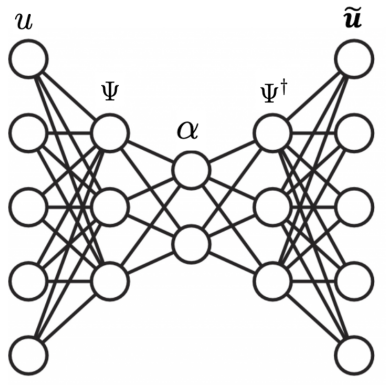


Figure 2: autoencoder systemic structure.

As described in [14] and [29], an autoencoder is a feed-forward neural network with a systemic structure that is composed of multiple hidden layers which include an encoding and a decoding part. The input and output layers have the same size as shown in Figure 2. The encoder compresses the input data ($u \in \mathbb{R}^n$) to some latent representation ($\alpha \in \mathbb{R}^p$):

$$\Psi : \begin{cases} \mathcal{U} \longrightarrow \mathbb{R}^p \\ u \longmapsto \alpha := \Psi(u) \end{cases}$$

whereas the decoder decodes the compressed latent representation α and reconstructs the input in $\tilde{u} \in \mathbb{R}^n$:

$$\Psi^\dagger : \begin{cases} \mathbb{R}^p \longrightarrow \mathbb{R}^n \\ \alpha \longmapsto \tilde{u} := \Psi^\dagger(\alpha). \end{cases}$$

Among the key aspects that have to be set in order to define an autoencoder, we need to specify the loss function to be minimised. In our paper, we have considered the following loss function:

$$\frac{1}{N_d} \sum_{i=1}^{N_d} \| u^{(i)} - \Psi^\dagger\big(\Psi(u^{(i)})\big) \|_n^2 \tag{3.1}$$

where $U = \left\{u^{(i)}\right\}_{1 \leq i \leq N_d}$ corresponds to the data set. In addition, we chose the exponential linear unit (ELU) [10] as the activation function:

$$ELU(x) = \begin{cases} x & \text{if } x \geq 0 \\ \rho(\exp(x) - 1) & \text{otherwise} \end{cases} \tag{3.2}$$

where $\rho > 0$ is a constant number (we have taken $\rho = 1.0$ in our numerical tests). This activation function is differentiable, a property which will allow to estimate the Lipschitz

constants associated to the encoder and decoder functions. The weights and biases of the hidden layers were optimised by using an Adam optimiser [40].

In the numerical tests described in Sections 4 and 5, different architectures will be considered and, for each of them, we will specify the values of the hyperparameters (numbers of hidden layers, number of neurons per layer).

3.2 A criterion based on stable manifold widths to determine the latent variable dimension

In this section, we consider $\mathcal{U} \subset \mathbb{R}^n$ a compact embedded submanifold of topological dimension $\overline{p}$ and we propose a criterion based on autoencoders to identify the dimension $\overline{p}$. We refer to [8] and [17] for an overview of the intrinsic dimension estimation methods, the methods aim to project the original data $\mathcal{U} \subset \mathbb{R}^n$ to a lower M-dimensional submanifold of $\mathbb{R}^n$ that $M < n$ in a way that we will not lose the information of the original data. When the minimum M-dimension is necessary to represent the observed properties in $\mathcal{U}$, M can be called intrinsic dimension.

Having in mind autoencoder applications, we are interested in nonlinear methods of approximation of $\mathcal{U}$ depending on p parameters and built on two Lipschitz mappings that correspond to the encoder map and the decoder map. Following [11], we then introduce the quantity: for $p \in \mathbb{N}$ and for given constants $\gamma > 0$ and $\gamma^\dagger > 0$,

$$\delta_{p,\gamma,\gamma^\dagger}(\mathcal{U}) = \inf_{\Psi,\, \Psi^\dagger} \sup_{u \in \mathcal{U}} \|u - \Psi^\dagger \circ \Psi(u)\|_n$$

where the infimum is taken over the γ Lipschitz functions $\Psi : \mathcal{U} \to \mathbb{R}^p$ and the $\gamma^\dagger$ Lipschitz functions $\Psi^\dagger : \mathbb{R}^p \to \mathbb{R}^n$.

This definition is introduced in [11] under the name of *stable manifold width* with the slight nuance that the definition of [11] also involves the infimum over all the norms in $\mathbb{R}^p$. As explained in [11], the concept of stable manifold width compared to the concept of manifold width (where the functions Ψ and $\Psi^\dagger$ are only assumed to be continuous) is motivated by the fact that the Lipschitz regularity allows to explicitly control perturbations coming from noise or numerical approximation.

In the following proposition, we are interested by the case where $p < \overline{p}$. This corresponds to the case where the dimension of the latent representation space is smaller than the topological dimension of the submanifold that the autoencoder has to approximate. Under this assumption, the proposition gives a bound from below for the stable manifold width.

Proposition 3.1. *Let $\mathcal{U}$ be a compact embedded submanifold of $\mathbb{R}^n$ of topological dimension $\overline{p}$. We assume that $p < \overline{p}$. Then, there exists a constant $C > 0$ which only depends on $\mathcal{U}$ such that*

$$\delta_{p,\gamma,\gamma^\dagger}(\mathcal{U}) \geq \left(\frac{C}{\gamma^\dagger \gamma}\right)^{p/(\overline{p}-p)} R \tag{3.3}$$

where $R > 0$ is such that $\mathcal{U} \subset B_n(0, R)$.

For $m \in \mathbb{N}^*$, we have denoted by $B_m(y, r)$ with $y \in \mathbb{R}^m$ and $r > 0$ the closed ball of center y and radius r.

Proof. The proof of this proposition will rely on a result given in [30]. This paper introduces and studies the notion of *Lipschitz width*, a concept close to *stable manifold width*. Let ϵ be such that $\epsilon > \delta_{p,\gamma,\gamma^\dagger}(\mathcal{U})$. Then, there exist a γ Lipschitz function $\Psi : \mathcal{U} \to \mathbb{R}^p$ and a $\gamma^\dagger$ Lipschitz function $\Psi^\dagger : \mathbb{R}^p \to \mathbb{R}^n$ such that

$$\sup_{u\in\mathcal{U}} \|u - \Psi^\dagger \circ \Psi(u)\|_n \leq \epsilon.$$

Since Ψ is γ Lipschitz and $\mathcal{U} \subset B_n(0, R)$, we have $\Psi(\mathcal{U}) \subset B_p(\Psi(0), \gamma R)$. Thus, we get that

$$\inf_{\Psi_p^\dagger} \sup_{u\in\mathcal{U}} \inf_{\alpha\in B_p(\Psi(0),\gamma R)} \|u - \Psi_p^\dagger(\alpha)\|_n \leq \epsilon$$

where the infimum is taken over all the maps $\Psi_p^\dagger : (B_p(\Psi(0), \gamma R), \|\cdot\|_p) \to \mathbb{R}^n$ such that

$$\sup_{\alpha_1,\alpha_2\in B_p(\Psi(0),\gamma R)} \frac{\|\Psi_p^\dagger(\alpha_1) - \Psi_p^\dagger(\alpha_2)\|_n}{\|\alpha_1 - \alpha_2\|_p} \leq \gamma^\dagger$$

Let us now introduce the scaled norm $\|\cdot\|_{p,s}$ in $\mathbb{R}^p$ defined by $\|\alpha\|_{p,s} = \frac{1}{\gamma R}\|\alpha\|_p$. Then we have

$$\inf_{\Psi_p^\dagger} \sup_{u\in\mathcal{U}} \inf_{\alpha\in B_{p,s}(\Psi(0),1)} \|u - \Psi_p^\dagger(\alpha)\|_n \leq \epsilon$$

where the infimum is taken over all the maps $\Psi_p^\dagger : (B_{p,s}(\Psi(0), 1), \|\cdot\|_{p,s}) \to \mathbb{R}^n$ such that

$$\sup_{\alpha_1,\alpha_2\in B_{p,s}(\Psi(0),1)} \frac{\|\Psi_p^\dagger(\alpha_1) - \Psi_p^\dagger(\alpha_2)\|_n}{\|\alpha_1 - \alpha_2\|_{p,s}} \leq \gamma^\dagger\gamma R.$$

This property coincides with the fact that the fixed Lipschitz width of $\mathcal{U}$ associated to the norm $\|\cdot\|_{p,s}$ for the Lipschitz constant $\gamma^\dagger\gamma R$ is smaller than ϵ. Now, let us introduce the Lipschitz width of $\mathcal{U}$ for the Lipschitz constant $\gamma^\dagger\gamma R$ which is defined as the infimum of the fixed Lipschitz widths over all the norms in $\mathbb{R}^p$ and denote it by $d_{p,\gamma^\dagger\gamma R}(\mathcal{U})$. We thus have the following property:

$$d_{p,\gamma^\dagger\gamma R}(\mathcal{U}) < \epsilon.$$

Thus according to Proposition 3.5 in [30], this implies that we have an upper bound on the Lipschitz constant which is given by:

$$\gamma^\dagger\gamma R \geq \frac{1}{3}\epsilon N_{2\epsilon}^{1/p}(\mathcal{U})$$

where $N_\epsilon(\mathcal{U})$ is the $\epsilon-$covering number of $\mathcal{U}$. Since $\mathcal{U} \subset B_n(0, R)$ is a compact embedded submanifold of dimension $\overline{p}$, we have that

$$N_\epsilon(\mathcal{U}) \geq C\frac{R^{\overline{p}}}{\epsilon^{\overline{p}}},$$

where C depends on the Lipschitz constants of the local maps of the finite atlas describing $\mathcal{U}$. Thus, we get that

$$\gamma^\dagger\gamma \geq CR^{\overline{p}/p-1}\epsilon^{1-\overline{p}/p}$$

which implies that

$$\epsilon \geq \left(\frac{C}{\gamma^\dagger\gamma}\right)^{p/(\overline{p}-p)} R$$

for every $\epsilon > \delta_{p,\gamma,\gamma^\dagger}(\mathcal{U})$. This property allows us to conclude the proof. □

In the following corollary, we give another formulation of the previous proposition which will be more useful for our numerical tests.

Corollary 3.2. *Under the same notations and hypotheses as in Proposition 3.1, we assume that there exist a γ Lipschitz function $\Psi : \mathcal{U} \to \mathbb{R}^p$ and a $\gamma^\dagger$ Lipschitz function $\Psi^\dagger : \mathbb{R}^p \to \mathbb{R}^n$ such that*

$$\sup_{u \in \mathcal{U}} \|u - \Psi^\dagger \circ \Psi(u)\|_n \leq \epsilon.$$

Then, there exists a constant $C > 0$ which only depends on $\mathcal{U}$ such that

$$\gamma^\dagger \gamma \geq C R^{\overline{p}/p - 1} \epsilon^{1 - \overline{p}/p}. \tag{3.4}$$

This result can be interpreted in the following way: if the dimension p is smaller than the topological dimension of $\mathcal{U}$, then getting an accurate approximation of the elements of $\mathcal{U}$ with the latent dimension p may be achieved only if the Lipschitz constants of the approximation mappings are sufficiently large. It is related to the fact that, as long as the latent dimension is smaller than the intrinsic dimension, the encoder and decoder functions have to compensate this with strong variations, in the same vein as the space-filling curves. Our criterion thus relies on the representation of the variation of the product of the Lipschitz constants of the encoder and decoder functions with respect to the latent dimension p at a given accuracy ϵ. Its relevance will be illustrated in Section 4.

3.3 Estimation of the Lipschitz constants

The criterion that we propose to estimate the intrinsic dimension relies on the inequality (3.4) and thus it is necessary to estimate the Lipschitz constants of the encoder Ψ and the decoder $\Psi^\dagger$. In this section, we explain how this is numerically achieved.

Since the activation function used to build these functions is the ELU, Ψ and $\Psi^\dagger$ are regular functions, in particular they are C^1 functions. Let us introduce the gradient of the function $\Psi : \mathcal{U} \to \mathbb{R}^p$ and denote it by $\nabla\Psi$:

$$\nabla\Psi : \begin{cases} \mathcal{U} \longrightarrow \mathbb{R}^{p \times n} \\ u \longmapsto \nabla\Psi(u) \end{cases}$$

Similarly, we denote by $\nabla\Psi^\dagger$ the gradient of $\Psi^\dagger : \mathbb{R}^p \to \mathbb{R}^n$:

$$\nabla\Psi^\dagger : \begin{cases} \mathbb{R}^p \longrightarrow \mathbb{R}^{n \times p} \\ \alpha \longmapsto \nabla\Psi^\dagger(\alpha) \end{cases}$$

The estimation of the product $\gamma^\dagger\gamma$ of the Lipschitz constants relies on the following lemma whose proof uses classical arguments.

Lemma 3.3. *For a function $f : \mathcal{U} \subset \mathbb{R}^n \to \mathbb{R}^n$ of class C^1, we have the following estimate: for all $u^{(1)}$, $u^{(2)} \in \mathcal{U}$,*

$$\frac{\|f(u^{(1)}) - f(u^{(2)})\|_{l^{q,n}}}{\|u^{(1)} - u^{(2)}\|_{l^{q,n}}} \leq \left[\sum_{k=1}^{n} \left(\sum_{j=1}^{n} \sup_{u \in \mathcal{U}} \left| \frac{\partial f_k}{\partial u_j}(u) \right|^z \right)^{\frac{q}{z}} \right]^{\frac{1}{q}}$$

where $z, q \in]1, +\infty[$ are such that $\dfrac{1}{z} + \dfrac{1}{q} = 1$.

In particular, if we set $z = q = 2$, we get that, for all $u^{(1)}$, $u^{(2)} \in \mathcal{U}$,

$$\frac{\|f(u^{(1)}) - f(u^{(2)})\|_n}{\|u^{(1)} - u^{(2)}\|_n} \leq \left(\sum_{j,k=1}^{n} \sup_{u \in \mathcal{U}} \left| \frac{\partial f_k}{\partial u_j}(u) \right|^2 \right)^{\frac{1}{2}}$$

Thus, if we apply this estimate to $f = \Psi^\dagger \circ \Psi$, we can have an approximation of a lower bound of $\gamma^\dagger \gamma$ by computing

$$\left(\sum_{j,k=1}^{n} \sum_{j=1}^{n} \max_{1 \leq m \leq N} \left| \sum_{l=1}^{p} \frac{\partial \Psi_k^\dagger}{\partial \alpha_l}(\Psi(u^{(m)})) \frac{\partial \Psi_l}{\partial u_j}(u^{(m)}) \right|^2 \right)^{\frac{1}{2}}$$

where we have selected N samples $(u^{(m)})_{1 \leq m \leq N}$ to have a discrete approximation of the supremum value of the gradient.

4 Numerical tests of intrinsic dimension estimations

In this section, we will present some numerical tests on the estimation of the intrinsic dimension of the data set by using an autoencoder, as described in Section 3. In particular, we will compare the criterion that we propose which is based on the estimation of the Lipschitz constants with a more classical criterion (we refer to [23] for a presentation of other criteria) based solely on the evolution of the accuracy with respect to the dimension of the latent dimension.

For the tests which we will present hereafter, we consider the architectures of three autoencoders, named AE1, AE2 and AE3, and described in Table 1, in which only half of the architecture is written, since we consider symmetric autoencoders (the encoder and the decoder have the same architectures). The number of layers corresponds to the number of hidden layers.

Each autoencoder is trained for different choices of p, the dimension of the latent layer. Then, the product of the Lipschitz constants $\gamma^\dagger \gamma$ is evaluated following the method presented in Section 3.3 and the error ϵ (which corresponds to the value of the loss function given by (3.1)) is computed in a validation set.

Table 1: Architectures of the autoencoder models

	Number of layers	Number of hidden units
AE1	7	150, 80, 20, p
AE2	11	250, 150, 80, 20, 10, p
AE3	15	250, 180, 120, 80, 50, 30, 15, p

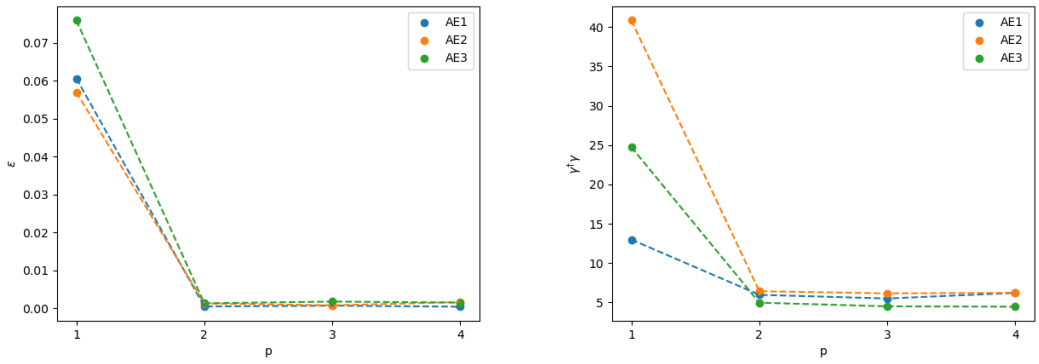
4.1 Tests on Van der Pol model

To build a data set from Van der Pol model (2.6), among the three parameters of the model, we fixed the initial velocity $x_0 = 0.5$ while the parameters v_0 and μ were randomly drawn from a uniform distribution in $[0.05, 1.5]$ and $[0.05, 2.0]$ respectively. Then, for each

parameter value, we computed an approximated solution of the ODE system in $[0, T]$, with $T = 30$ thanks to the Crank-Nicolson scheme with the time step given by $\Delta t = 0.075$. A total of $N_d = 1200$ solutions were generated, of which 1000 were used in the training set and 200 in the validation set.

The training of the autoencoder was performed by optimising the loss function defined in equation (3.1) by using the Adam optimiser and by taking 1000 epochs with a batch size equal to 40. Each training for a given p was repeated 10 times and the results presented hereafter correspond to the average of these 10 tests.

In Figure 3(a), we have represented the variations of the error with respect to p for different architectures. As explained in [23], these curves allow to identify the intrinsic dimension which corresponds to the value of p from which the curve begins to stagnate. In Figure 3(b), we have represented the variations of the products of the Lipschitz constants to apply the criterion that we proposed. For both methods, we can observe a stagnation for $p = 2$. Therefore, for this simple data set, both criteria are able to identify the right intrinsic dimension $\bar{p} = 2$.



(a) ϵ against different number of p

(b) $\gamma^\dagger\gamma$ against different number of p

Figure 3: Intrinsic dimension estimation for Van der Pol model and $\bar{p} = 2$

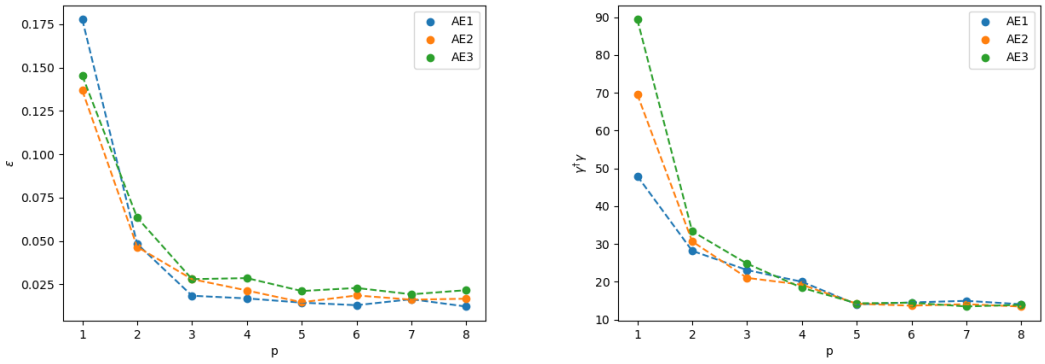
4.2 Tests on the FitzHugh-Nagumo model

In this section, we present the results obtained on the intrinsic dimension estimation for the solutions of the FitzHugh-Nagumo model. We consider here a scenario in which we test the dimension estimation when $\bar{p} = 5$. The numerical approximation of FitzHugh-Nagumo model is carried out by using the Crank-Nicolson scheme on a time interval $(0, T)$ with $T = 200$ and with the time step given by $\Delta t = 0.5$.

To generate the data set, we have kept I_{ext} fixed to the value $I_{ext} = 0.325$ whereas b, τ, a, v_0 and w_0 are drawn from a uniform distribution respectively in the intervals $[0.05, 0.5]$, $[12.0, 13.0]$, $[0.05, 0.5]$, $[0.05, 1.0]$, and $[0.05, 1.0]$. So, in this case, the intrinsic dimension is $\bar{p} = 5$.

If we consider the curve of the accuracy with respect to the latent dimension p, reported in Figure 4(a), we observe a stagnation from $p = 3$, which would lead to a wrong conclusion. Let us mention that the difficulty to identify the latent dimension thanks to this simple and natural criterion has already been highlighted in several papers (we

refer for instance to [23]). In particular, in relatively high dimension, it is often observed that the curve gradually decreases and starts to stagnate before reaching the intrinsic dimension. On the other hand, if we consider the curve of the product of the Lipschitz constants with respect to the latent dimension, reported in Figure 4(b), we can observe that a stagnation only occurs from $p = 5$. Thus the criterion that we propose is able to correctly identify the value of the intrinsic dimension.



(a) ϵ against different number of p (b) $\gamma^\dagger\gamma$ against different number of p

Figure 4: Intrinsic dimension estimation for FitzHugh-Nagumo model and $\bar{p} = 5$

5 Parameter identification with LVGF method

In this section, we will assess the performances of the LVGF method for the identification of parameters in Van der Pol and Fitzhugh-Nagumo models. For each numerical test, we will compare the results given by LVGF method with what we will call *the classical method*: it consists in minimising the classical misfit functional J given by (2.5) thanks to a gradient descent algorithm with a fixed step size.

Before describing each test, let us start with some practical details for the implementation of the LVGF and classical methods. For the training process in the preliminary step of the LVGF method, we have trained the autoencoders in 1000 epochs with 40 batch sizes. In Algorithm 1 which describes LVGF method, the maximum number of iterations has been set to $N_{max} = 2000$, the tolerance for $\|\alpha^* - \alpha_k\|_p$ has been set to $\sigma = 10^{-4}$ and the steps were given by $S = [0.0005, 0.0025, 0.0125, 0.0625]$.

To quantify the errors in the parameter estimation, we evaluated the total error by computing the root relative sum of squares (RRSSE):

$$RRSSE(\theta^*, \theta) = \frac{\|\theta^* - \theta\|_m}{\|\theta^*\|_m}$$

and, to evaluate the errors in each of the parameters, we also computed the following relative error (RE) for each parameter:

$$RE(\theta_i^*, \theta_i) = \frac{|\theta_i^* - \theta_i|}{|\theta_i^*|}, \ 1 \leq i \leq m.$$

At last, the error between the observed signal u^* and the signal associated to the estimated parameters $u = \varphi(\theta)$ is defined as the root relative mean squared error (RRMSE):

$$RRMSE(u^*, u) = \frac{\|u^* - u\|_n}{\|u^*\|_n}.$$

5.1 Parameter identification for Van der Pol model

In this section, we consider Van der Pol model and are interested by the identification of the parameter vector $\theta = (x_0, v_0, \mu)$.

For LVGF method, we have taken the same data set of 1200 samples and the same autoencoder setup as in Section 4.1. In particular, in the data set, θ_1 has a valued fixed to $\theta_1 = 0.5$, whereas θ_2 and θ_3 are respectively drawn from uniform distributions in $[0.05, 1.5]$ and in $[0.05, 2.0]$. The architecture of the autoencoder corresponds to AE1 described in Table 1 and its latent dimension has been identified in Section 4.1 to be equal to $p = 2$ thanks to the criterion presented in Section 3.2.

In order to compare the classical method and the LVGF method, we tested them for 10 random values of θ^* taken in the same range as the data set for θ_2 and θ_3 and in the range of $[0.05, 1.5]$ for θ_1. Moreover, for each θ^*, we performed the parameter estimation methods starting from 20 different values of initial guess taken in the same range.

The results are presented in Tables 2 and 3. Table 2 corresponds to the errors on the parameters and the signals averaged over the 10 different values of θ^* and the 20 different initial guesses whereas Table 3 details the results (still averaged over the initial guesses) obtained for two values of θ^* (corresponding to the best case and the worst case). We can see that the values obtained with LVGF method are significantly closer to the right values than the classical method. With LVGF method, both the averaged RRSSE and RRMSE are about 10 times smaller than the ones obtained by using the classical method. Moreover, even if, for the best case, the classical method gives very accurate results, most of the time, we observe that the LVGF method could provide us estimations with smaller RE and that the classical method converges to a local minima, which prevents it from reaching the right value and leads to significant errors.

Table 2: Comparison between the classical method and LVGF method: averaged results on 10 tests with Van der Pol model

	RE of each estimated θ	RRSSE of the estimated θ	RRMSE of the estimated signal
classical method	(0.327, 0.66, 1.37)	1.412	0.4247
LVGF method	(0.055, 0.095, 0.01)	0.112	0.0231

In order to better assess the performances of the methods and illustrate them, we are going to describe the results in more detail for a case which is representative of what is observed in general. We consider the case where the parameter to recover is given by $\theta^* = (0.5, 0.5, 0.5)$ and where we start from the initial guess $\theta = (0.6615, 1.23, 1.6994)$. The results obtained in this specific case are reported in Table 4. The iterations obtained with the classical method are represented in Figure 5 (a) where, to better visualize them, we have chosen to represent only θ_2 and θ_3. Figure 5 (b) represents the values taken

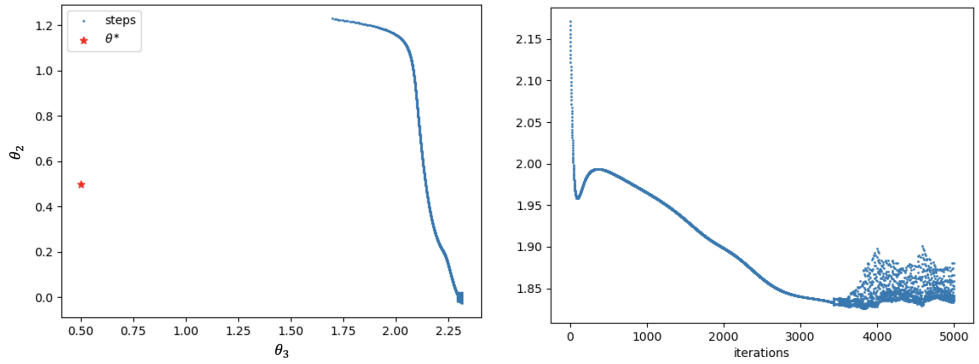
Table 3: Comparison between the classical method and LVGF method for Van der Pol model: detailed results for two specific cases (the best case and the worst case) and averaged results over 10 different parameter values. At each line, the results are averaged over 20 initial guesses.

	Methods	RE of each estimated θ	RRSSE of the estimated θ	RRMSE of the estimated signal
Best case	Classical	(0.0003, 0.0003, 0.0001)	0.0004	0.00008
	LVGF	(0.06, 0.07, 0.009)	0.091	0.0196
Worst case	Classical	(1.29, 1.92, 8.38)	8.928	0.77
	LVGF	(0.08, 0.42, 0.03)	0.46	0.07

by the misfit function J given by (2.5) at the successive iterations. We note that the gradient method experiences convergence issues and that the iterations stagnate to a local minimum which is far away from θ^*.

Table 4: Comparison between the classical method and LVGF method in a specific case for Van der Pol model: parameter to identify $\theta^* = (0.5, 0.5, 0.5)$ and initial guess $\theta = (0.6615, 1.23, 1.6994)$

	estimated θ	RRSSE of the estimated θ	RRMSE of the estimated signal
classical method	$(0.44, -0.16, 2.32)$	3.8666	1.81
LVGF method	$(0.52, 0.52, 0.502)$	0.0541	0.011



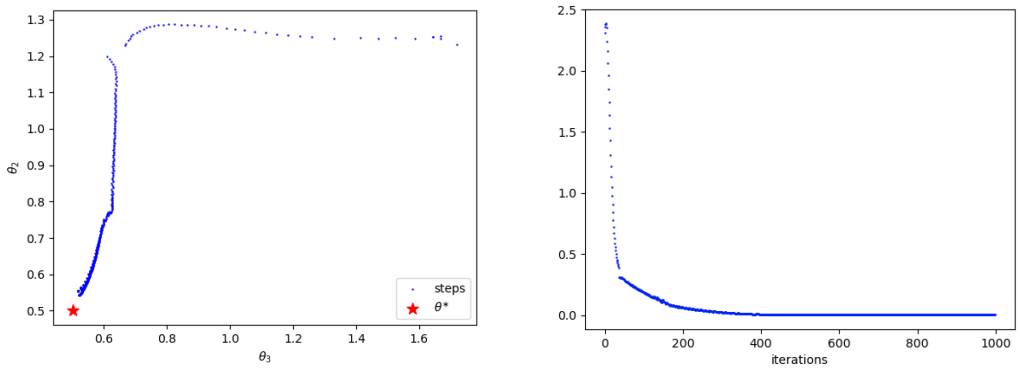
(a) Values of θ_2 and θ_3 at each iteration (b) Values of J at each iteration

Figure 5: Illustration of the test presented in Table 4: representations of the iterations with the classical method

On the other hand, with LVGF method, the iterations converge to the correct values of the parameters, as illustrated in Figure 6. In addition, Figure 7 also depicts the iterations of the latent variable $\alpha = (\alpha_1, \alpha_2)$ and we observe that the descent direction for this variable allows to go in a relatively direct way towards the value α^*.

Still for this specific case, let us finally illustrate the errors on the signal presented in the last column of Table 4. For both methods, Figure 8 presents a comparison of the

signal associated to the retrieved parameter with the true signal u^* associated to θ^*, the value to be identified. The signal corresponding to the initial guess (which is the same for both methods) corresponds to the green curve. We can see that the curve associated to the LVGF method perfectly fits the measured signal whereas the signal reconstructed thanks to the classical method is quite far from the measured signal.



(a) Values of θ_2 and θ_3 at each iteration

(b) Values of J at each iteration

Figure 6: Illustration of the test presented in Table 4: representations of the iterations with LVGF method

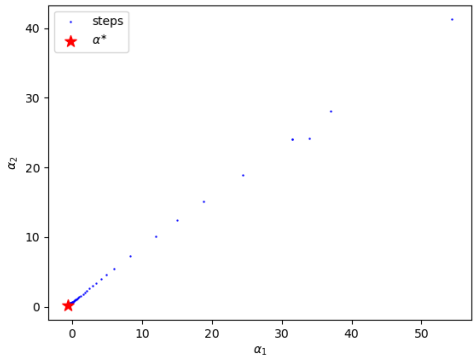
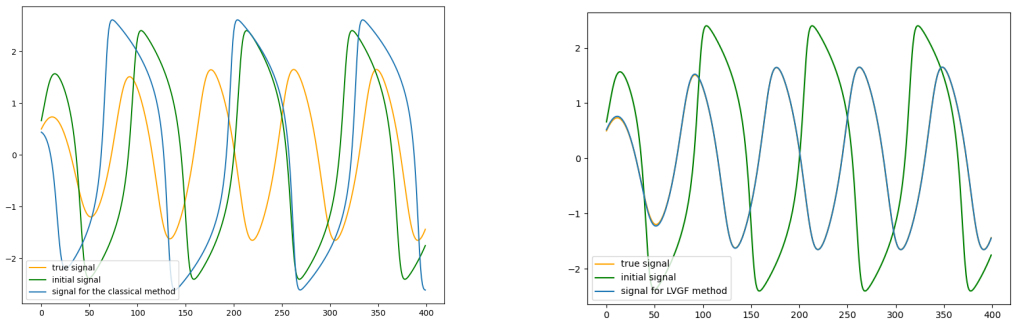


Figure 7: Illustration of the test presented in Table 4: values of α at each iteration

5.2 Parameter identification for FitzHugh-Nagumo model

This section is devoted to a presentation of the numerical results obtained for the identification of parameters in FitzHugh-Nagumo model (2.7). In these tests, the values of I_{ext} and of the initial condition of v are fixed to $I_{ext} = 0.325$ and $v_0 = 1$ and we want to identify the four remaining parameters $\theta = (w_0, a, b, \tau)$.

For LVGF method, we have considered a data set of 1200 samples corresponding to a fixed value of τ given by $\tau = 12.5$ whereas the values of w_0, a and b are respectively drawn from a uniform distribution in $[0.05, 1.0]$, $[0.05, 0.8]$, and $[0.05, 0.8]$. With regard to the setup and training process of the autoencoder, we made the same choices as in the



(a) Signal generated with the parameters obtained with the classical method

(b) Signal generated with the parameters obtained with LVGF method

Figure 8: Illustration of the test presented in Table 4: comparison of the signals with the classical method and with LVGF method (Van der Pol model)

previous section for Van der Pol model, except that the dimension of the latent layer is given by $p = 3$.

To test the performances of the LVGF method and compare them with the classical method, we repeated the same statistical test as for Van der Pol model by considering 10 random values of θ^* and 20 initial guesses for each value of θ^*. We took the parameters in the same range as the data set for w_0, a, b and in the range of $[12, 13]$ for τ.

As in the previous section, the results are given by two tables: Table 5 gives the averaged results whereas Table 6 details the results obtained for two values of θ^* (corresponding to the best case and the worst case) still averaged on the different initial guesses.

As for Van der Pol model, we observe that the relative error made on the parameters is smaller with LVGF method than with the classical method. More precisely, we see that the error is reduced by a factor of 2 or more and that the error on the signal is reduced by a factor of about 7. If we look at the relative error parameter by parameter, we also observe that the error on the reconstruction of the parameter b (third parameter in θ) is relatively large with both methods still with a clear improvement with LVGF method (around 54% with the classical method and 27% for LVGF method). Since this lack of accuracy has a rather low impact on the reconstruction of the signal, this difficulty to identify the parameter b compared to w_0, a and τ is related to the differences of sensitivities of the signal with respect to the parameters: its sensitivity with respect to b is smaller than its sensitivity with respect to the other parameters and so a relatively rough identification still allows to accurately reconstruct the signal.

At last, the results for a specific example corresponding to $\theta^* = (0.5, 0.5, 0.5, 12.5)$ with the initial guess $\theta = (0.4815, 0.223, 0.689, 12.52)$ are detailed in Table 7. In that case, the reconstruction of the signal associated to the parameters is presented in Figure 9. We can see that the signal generated by the parameters obtained with LVGF method perfectly fits the measured signal.

Table 5: Comparison between the classical method and LVGF method: averaged results on 10 tests (FitzHugh-Nagumo model)

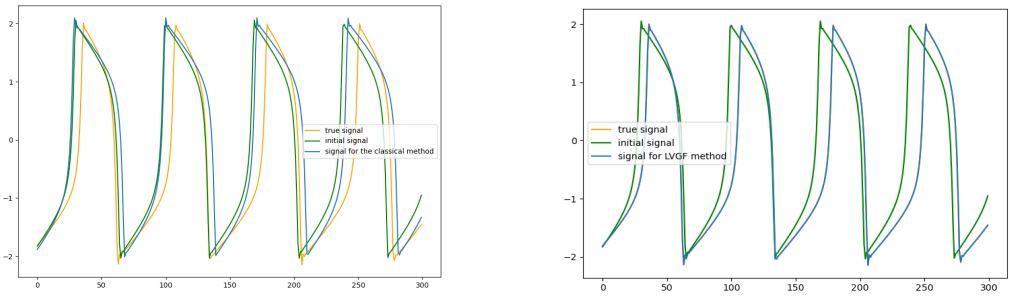
Method	RE of each estimated θ	RRSSE of the estimated θ	RRMSE of the estimated signal
classical method	(0.174, 0.322, 0.544, 0.036)	0.703	0.3662
LVGF method	(0.074, 0.08, 0.27, 0.014)	0.3026	0.0529

Table 6: Comparison between the classical method and LVGF method for FitzHugh-Nagumo model: detailed results for two specific cases (the best case and the worst case) and averaged results over 10 different parameter values. At each line, the results are averaged over the initial guesses.

	Methods	RE of each estimated θ	RRSSE of the estimated θ	RRMSE of the estimated signal
Best case	Classical	(0.06, 0.14, 0.25, 0.02)	0.304	0.152
	LVGF	(0.052, 0.035, 0.16, 0.014)	0.168	0.0142
Worst case	Classical	(0.33, 0.48, 1.74, 0.04)	1.867	0.579
	LVGF	(0.1, 0.11 , 0.47, 0.019)	0.506	0.076

Table 7: Comparison between the classical method and LVGF method in a specific case for FitzHugh-Nagumo model: parameter to identify $\theta^* = (0.5, 0.5, 0.5, 12.5)$ and initial guess $\theta = (0.4815, 0.223, 0.689, 12.52)$

	Estimated θ	RRSSE of estimated θ	RRMSE of estimated signal
classical method	(0.59, 0.05, 0.59, 12.99)	0.9344	0.6695
LVGF method	(0.503, 0.502, 0.508, 12.48)	0.0203	0.00624



(a) Signal generated with the parameters obtained with the classical method

(b) Signal generated with the parameters obtained with LVGF method

Figure 9: Illustration of the tests presented in Table 7: comparison of the signals with the classical method and with LVGF method (FitzHugh-Nagumo model)

6 Conclusion and discussion

In this study, we have presented an innovative method called Latent Variables Gradient Flow (LVGF) which leverages the available data and the underlying dynamics of the system in order to identify parameters of the ODE model. This approach entails a dual-phase process. In the first phase, an autoencoder is trained on the data set in order to represent it in a compressed way. The description of the data set by a latent variable is then exploited in the second phase which corresponds to an optimization method that can be described as a gradient flow for the latent variables.

For this new method, we presented numerical tests and a comparison with the classical method corresponding to a gradient descent method applied to the minimisation of the classical data misfit functional (2.5). A noticeable property of LVGF method is that it is able to avoid convergence towards local minimum points. In this way, LVGF method generally gives much better results than the classical method for the parameters identification problem.

In addition, we have shown that the sequence defined by the LVGF method satisfies convergence properties. These results still need to be extended to the algorithm actually implemented where there is an additional projection step. Nevertheless, our study, even if it is incomplete, allows to understand how the use of the non-linear mapping between the parameter space and the latent variable space latent variable allows to convexify our initial minimization problem.

In a complementary manner to this parameter identification method, we also proposed a new criterion to identify the intrinsic dimension of a sub-manifold thanks to autoencoders. This criterion is justified by a theoretical result based on the notion of stable manifold width. It describes the behaviour of the product of the Lipschitz constants of the encoder and decoder when the latent variable is too small. We have compared our criterion to the classical criterion based on the evolution of the error with respect to the latent dimension and observed that it leads to better results. Indeed, in the case of the classical criterion, the error gradually decreases when the latent dimension increases, making it difficult to pick a single value for the dimension. On the contrary, our criterion clearly highlights two phases in the evolution of the product of the Lipschitz constants and the intrinsic dimension corresponds to the elbow of the curve.

In addition to the theoretical study which must be in-depth, this work opens a certain number of perspectives. First of all, LVGF method has been tested on simple dynamical systems to start assessing its basic properties. We intend to continue exploring its use in more complex situations corresponding to larger dynamical systems that involve a large number of parameters.

At last, the autoencoder used to approximate the available population data is based on neural networks. This is not a necessary choice, and other manifold learning methods could be used ([9]). For instance, Principal Component Analysis (PCA), kernel-PCA, Locally Linear Embedding, Isomap, Laplacian Eigenmaps, Semidefinite Embedding are all methods which could be used in order to build an encoder-decoder pair.

Acknowledgements

We would like to thank the support from INSPIRE European Training Network which receives funding from the EU Horizon 2020 Research and Innovation programme, under the Marie Skłodowska-Curie GA 858070 [21]. We also appreciated all the help from Sylvain Bernasconi and Christophe Bleunven at NOTOCORD, an Instem company.

A Appendix

Proof of Proposition 2.1 (identifiability and observability for Van der Pol model)

We consider two solutions of system (2.6) $(x^{(1)}, v^{(1)})$ and $(x^{(2)}, v^{(2)})$ respectively associated to the set of initial conditions and parameters $X^{(1)} = (x_0^{(1)}, v_0^{(1)}, \mu^{(1)})$ and $X^{(2)} = (x_0^{(2)}, v_0^{(2)}\mu^{(2)})$. We assume that the measurements on these two solutions coincide, that is $x^{(1)} = x^{(2)}$ in $[0, T]$ and we want to prove that $X^{(1)} = X^{(2)}$.

First, by assumption, we immediately have that $x_0^{(1)} = x_0^{(2)}$. Next, according to the first equation of (2.6), we get that $v^{(1)} = v^{(2)}$ in $[0, T]$, so in particular $v_0^{(1)} = v_0^{(2)}$. At last, since $\dot{v}^{(1)} = \dot{v}^{(2)}$ in $[0, T]$, the second equation of (2.6) taken at $t = 0$ gives that

$$\mu^{(1)}(1 - (x_0^{(1)})^2)v_0^{(1)} = \mu^{(2)}(1 - (x_0^{(2)})^2)v_0^{(2)}.$$

Thus, for almost all $X^{(1)}$ and $X^{(2)}$, we deduce that $\mu^{(1)} = \mu^{(2)}$ and we conclude that $X^{(1)} = X^{(2)}$.

Proof of Proposition 2.2 (identifiability and observability for FitzHugh-Nagumo model)

We consider two solutions of system (2.7) $(v^{(1)}, w^{(1)})$ and $(v^{(2)}, w^{(2)})$ respectively associated to the set of initial conditions and parameters $X^{(1)} = (v_0^{(1)}, w_0^{(1)}, a^{(1)}, b^{(1)}, \tau^{(1)})$ and $X^{(2)} = (v_0^{(2)}, w_0^{(2)}, a^{(2)}, b^{(2)}, \tau^{(2)})$. We assume that the measurements on these two solutions coincide, that is $v^{(1)} = v^{(2)}$ in $[0, T]$ and we want to prove that $X^{(1)} = X^{(2)}$.

First, by assumption, we immediately have that $v_0^{(1)} = v_0^{(2)}$. Next, according to the first equation of (2.7), we get that $w^{(1)} = w^{(2)}$ in $[0, T]$, so in particular $w_0^{(1)} = w_0^{(2)}$. Using the second equation of (2.7), we have in $[0, T]$

$$\frac{1}{\tau^{(1)}}(v^{(1)} + a^{(1)} - b^{(1)}w^{(1)}) = \frac{1}{\tau^{(2)}}(v^{(2)} + a^{(2)} - b^{(2)}w^{(2)}) \tag{A.1}$$

and

$$\frac{1}{\tau^{(1)}}(\dot{v}^{(1)} - b^{(1)}\dot{w}^{(1)}) = \frac{1}{\tau^{(2)}}(\dot{v}^{(2)} - b^{(2)}\dot{w}^{(2)}).$$

This implies that

$$\left(\frac{1}{\tau^{(1)}} - \frac{1}{\tau^{(2)}}\right)\dot{v}^{(1)} - \left(\frac{b^{(1)}}{\tau^{(1)}} - \frac{b^{(2)}}{\tau^{(2)}}\right)\dot{w}^{(1)} = 0. \tag{A.2}$$

Let us prove that $\dot{v}^{(1)}$ and $\dot{w}^{(1)}$ are linearly independent. If it does not hold, there exists $\lambda \in \mathbb{R}^*$ such that $\dot{w}^{(1)} = \lambda\dot{v}^{(1)}$. Using system (2.7), this implies that, for all $t \in [0, T]$

$$\frac{1}{\tau^{(1)}}(v^{(1)} + a^{(1)} - b^{(1)}w^{(1)}) = \lambda\left(v^{(1)} - \frac{(v^{(1)})^3}{3} - w^{(1)} + I_{ext}\right)$$

Differentiating this identity, we get that, for all $t \in [0, T]$

$$\frac{1}{\tau^{(1)}}(\dot{v}^{(1)} - b^{(1)}\dot{w}^{(1)}) = \lambda\left(\dot{v}^{(1)} - (v^{(1)})^2\dot{v}^{(1)} - \dot{w}^{(1)}\right)$$

Replacing $\dot{w}^{(1)}$ by $\lambda\dot{v}^{(1)}$, we get

$$\dot{v}^{(1)}\left[\lambda\big(1-(v^{(1)})^2-\lambda\big)-\frac{1}{\tau^{(1)}}(1-b^{(1)}\lambda)\right]=0.$$

So this implies that, for all $t\in[0,T]$, either $\dot{v}^{(1)}(t)=0$ or $(v^{(1)})^2(t)$ is given by a constant. Using that $v^{(1)}$ is a continuous function, this implies that it is a constant function in $[0,T]$ and since $\dot{w}^{(1)}=\lambda\dot{v}^{(1)}$, $w^{(1)}$ is also a constant function in $[0,T]$.

So, if $v^{(1)}$ and $w^{(1)}$ are not constant functions, we get that $\dot{v}^{(1)}$ and $\dot{w}^{(1)}$ are linearly independent and we deduce from (A.2) that

$$\left(\frac{1}{\tau^{(1)}}-\frac{1}{\tau^{(2)}}\right)=0 \text{ and } \left(\frac{b^{(1)}}{\tau^{(1)}}-\frac{b^{(2)}}{\tau^{(2)}}\right)=0.$$

By this way, we get that $b^{(1)}=b^{(2)}$ and $\tau^{(1)}=\tau^{(2)}$. At last, using (A.1), we deduce that $a^{(1)}=a^{(2)}$. So, if $v^{(1)}$ and $w^{(1)}$ are not constant functions (which holds for almost all $X^{(1)}$), we have obtained that $X^{(1)}=X^{(2)}$.

References

[1] R. C. Aster, B. Borchers, and C. H. Thurber. *Parameter estimation and inverse problems*. Elsevier, 2018.

[2] N. Baden and J. Villadsen. "A family of collocation based methods for parameter estimation in differential equations". In: *The Chemical Engineering Journal* 23.1 (1982), pp. 1–13.

[3] Y. Bard. "Comparison of gradient methods for the solution of nonlinear parameter estimation problems". In: *SIAM Journal on Numerical Analysis* 7.1 (1970), pp. 157–186.

[4] J. V. Beck and K. J. Arnold. *Parameter estimation in engineering and science*. James Beck, 1977.

[5] R. Bellman and K. J. Åström. "On structural identifiability". In: *Mathematical biosciences* 7.3-4 (1970), pp. 329–339.

[6] A. Bjorck. *Numerical Methods for Least Squares Problems*. Vol. 51. SIAM, 1996.

[7] B. V. D. Bosch and L. Hellinckx. "A new method for the estimation of parameters in differential equations". In: *AIChE Journal* 20.2 (1974), pp. 250–255.

[8] F. Camastra and A. Staiano. "Intrinsic dimension estimation: Advances and open problems". In: *Information Sciences* 328 (2016), pp. 26–41.

[9] L. Cayton. *Algorithms for manifold learning*. eScholarship, University of California, 2008.

[10] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. "Fast and accurate deep network learning by exponential linear units (elus)". In: *arXiv preprint arXiv:1511.07289* (2015).

[11] A. Cohen, R. DeVore, G. Petrova, and P. Wojtaszczyk. "Optimal stable nonlinear approximation". In: *Foundations of Computational Mathematics* 22.3 (2022), pp. 607–648.

[12] D. Csercsik, G. Szederkényi, and K. M. Hangos. "Identifiability of a Hodgkin-Huxley type ion channel under voltage step measurement conditions". In: *IFAC Proceedings Volumes* 43.5 (2010). 9th IFAC Symposium on Dynamics and Control of Process Systems, pp. 332–337. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20100705-3-BE-2011.00055. URL: https://www.sciencedirect.com/science/article/pii/S1474667016300568.

[13] N. Cunniffe, F. Hamelin, A. Iggidr, A. Rapaport, and G. Sallet. "Observability, Identifiability and Epidemiology-A primer". In: (2023).

[14] H. A. Dau, V. Ciesielski, and A. Song. "Anomaly detection using replicator neural networks trained on examples of one class". In: *Simulated Evolution and Learning: 10th International Conference, SEAL 2014, Dunedin, New Zealand, December 15-18, 2014. Proceedings 10*. Springer. 2014, pp. 311–322.

[15] J. E. Dennis Jr and J. J. Moré. "Quasi-Newton methods, motivation and theory". In: *SIAM review* 19.1 (1977), pp. 46–89.

[16] T. T. Dufera, Y. C. Seboka, C. Fresneda Portillo, et al. "Parameter Estimation for Dynamical Systems Using a Deep Neural Network". In: *Applied Computational Intelligence and Soft Computing* 2022 (2022).

[17] K. Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[18] L. Gabrielli, S. Tomassetti, S. Squartini, and C. Zinato. "Introducing deep machine learning for parameter estimation in physical modelling". In: *Proceedings of the 20th international conference on digital audio effects*. 2017.

[19] P. E. Gill and W. Murray. "Quasi-Newton methods for unconstrained optimization". In: *IMA Journal of Applied Mathematics* 9.1 (1972), pp. 91–108.

[20] V. Grimm, A. Heinlein, A. Klawonn, M. Lanser, and J. Weber. "Estimating the time-dependent contact rate of SIR and SEIR models in mathematical epidemiology using physics-informed neural networks". In: *Electronic Transactions on Numerical Analysis* 56 (2022), pp. 1–27.

[21] P.-J. Guns. ""INSPIRE": A European training network in safety pharmacology creating opportunities for 15 PhD students". In: *Journal of Pharmacological and Toxicological Methods* 105 (2020), p. 106838.

[22] B. Hartke. "Global optimization". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1.6 (2011), pp. 879–887.

[23] T. Kärkkäinen and J. Hänninen. "Additive autoencoder for dimension estimation". In: *Neurocomputing* 551 (2023), p. 126520.

[24] X. Kong, K. Yamashita, B. Foggo, and N. Yu. "Dynamic parameter estimation with physics-based neural ordinary differential equations". In: *2022 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2022, pp. 1–5.

[25] H. Liang and H. Wu. "Parameter Estimation for Differential Equation Models Using a Framework of Measurement Error in Regression Models". In: *Journal of the American Statistical Association* 103.484 (2008), pp. 1570–1583.

[26] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature machine intelligence* 3.3 (2021), pp. 218–229.

[27] H. Miao, X. Xia, A. S. Perelson, and H. Wu. "On identifiability of nonlinear ODE models and applications in viral dynamics". In: *SIAM review* 53.1 (2011), pp. 3–39.

[28] E. H. Moore. *On the reciprocal of the general algebraic matrix*. Vol. 26. 9. Bulletin of the American Mathematical Society, 1920, pp. 394–395.

[29] J. Pereira and M. Silveira. "Unsupervised representation learning and anomaly detection in ECG sequences". In: *International Journal of Data Mining and Bioinformatics* 22.4 (2019), pp. 389–407.

[30] G. Petrova and P. Wojtaszczyk. "Lipschitz widths". In: *Constructive Approximation* 57.2 (2023), pp. 759–805.

[31] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. "On the spectral bias of neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5301–5310.

[32] J. R. Raol, G. Girija, and J. Singh. *Modelling and parameter estimation of dynamic systems*. Vol. 65. Iet, 2004.

[33] O. Sallam and M. Fürth. "On the use of Fourier Features-Physics Informed Neural Networks (FF-PINN) for forward and inverse fluid mechanics problems". In: *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 237 (4 2023), pp. 846–866.

[34] B. Van der Pol. "A theory of the amplitude of free and forced triode vibrations, Radio Review. 1 (1920) 701-710, 754-762". In: *Selected Scientific Papers* 1 (1960).

[35] N. Verdière and C. Jauberthie. "Parameter estimation procedure based on input-output integro-differential polynomials. Application to the Hindmarsh-Rose model". In: *2020 European Control Conference (ECC)*. 2020, pp. 220–225. DOI: 10.23919/ECC51009.2020.9143670.

[36] A. F. Villaverde and G. Massonis. "On testing structural identifiability by a simple scaling method: Relying on scaling symmetries can be misleading". In: *PLoS computational biology* 17.10 (2021), e1009032.

[37] S. Wang, H. Wang, and P. Perdikaris. "Learning the solution operator of parametric partial differential equations with physics-informed DeepONets". In: *Science advances* 7.40 (2021), eabi8605.

[38] D. Wolf and R. Moros. "Estimating rate constants of heterogeneous catalytic reactions without supposition of rate determining surface steps—an application of a genetic algorithm". In: *Chemical Engineering Science* 52.7 (1997), pp. 1189–1199.

[39] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis. "Systems biology informed deep learning for inferring parameters and hidden dynamics". In: *PLOS Computational Biology* 16.11 (Nov. 2020), pp. 1–19. URL: https://doi.org/10.1371/journal.pcbi.1007575.

[40] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. "Dive into deep learning". In: *arXiv preprint arXiv:2106.11342* (2021).

[41] A. Žilinskas and A. Zhigljavsky. "Stochastic global optimization: a review on the occasion of 25 years of Informatica". In: *Informatica* 27.2 (2016), pp. 229–256.