UFR de mathématique et d'informatique
Université de Strasbourg

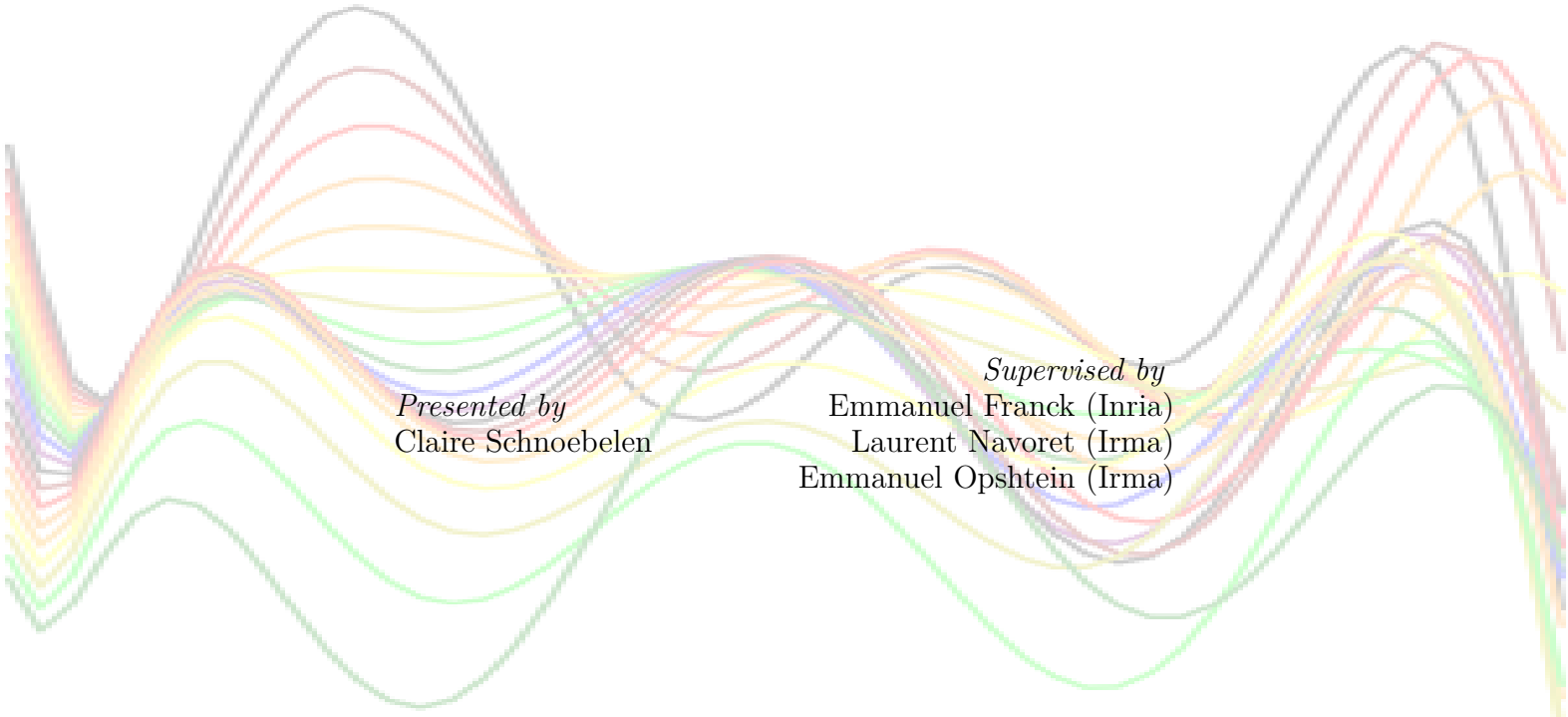
Inria

Non-linear reduced order models for Hamiltonian systems

Internship report

M2 Calcul Scientifique et Mathématiques de l'Information

August 21, 2023



Presented by
Claire Schnoebelen

Supervised by
Emmanuel Franck (Inria)
Laurent Navoret (Irma)
Emmanuel Opshtein (Irma)

Contents

Introduction

This document presents the work I have done during my internship, as part of my second year of Masters. The internship took place at Inria between 13 February and 11 August. I was supervised by Emmanuel Franck, Laurent Navoret and Emmanuel Opshtein, researchers at Inria and Irma.

Inria (for Institut National de Recherche en Informatique et Automatique) is a public establishment created in 1967 within the framework of the *Plan calcul*, a governmental plan intended to develop French knowledge in the field of digital technology and to ensure the digital sovereignty of the country. Today, it has 200 teams spread over 10 research centres, bringing together a total of 3,900 researchers and engineers in mathematics and computer science [13]. The institute works by *équipes-projets*, groups of about 20 people working on the same project and for the most part in collaboration with companies [13]. The Nancy research centre was founded in 1986 and today has 20 teams bringing together over 400 people. It has a branch at the University of Strasbourg, where researchers from the TONUS team (for TOkamaks and NUmerical Simulations) work, including one of my supervisor, Emmanuel Franck.

Irma (for Institut de Recherche en Mathématique Avancée) is a research centre in mathematics under the administrative supervision of the University of Strasbourg and the CNRS (for Centre National de la Recherche Scientifique) [14]. It was established in 1966 as the first research centre associated with the CNRS, a public institution created in 1939 to structure and promote French public research [8]. Irma counts about 130 members distributed in 7 research teams, including the Geometry team, to which Emmanuel Opshtein belongs, and the MoCo (for *Modélisation et Contrôle*) team, to which Laurent Navoret belongs.

This internship is in line with last year's internship, where we studied linear reduction methods for Hamiltonian systems. The objective of this year internship was to extend and improve those methods in the non-linear case. Hamiltonian systems are systems of partial differential equations whose flows have the particularity to preserve the energy of the underlying physical systems. In geometry, these problems are studied in the field of symplectic geometry. Given a partial differential equation, the objective of a reduced order model is to find a family of functions that alone can explain a large part of the behaviour of the equation's solutions. Reduced order modelling is interesting because it allows a faster computation of solutions at any time and for any value of the equation parameters in the interval considered during reduction. As we observed during my last year's internship, linear methods such as Proper Symplectic Decomposition (PSD) give good results for linear Hamiltonian equations but fail for non-linear one.

The main task of my internship was to think of ways to improve the reduction we obtain

with PSD. We have successively explored two different approaches to address this problem. In the first one, we try to correct the application that sends the solutions on the reduced space given by the PSD. Here, we used quadratic corrections and mainly tried to adapt a method proposed in [11] to the Hamiltonian case. This task then involved paper readings, mathematical computations to set the problem and programming (in Python). After fruitless tests for several variations of this idea, we put it aside and started working on the second approach. The second idea is to directly build a Hamiltonian dynamics in the reduced space given by the PSD. Here, we used a control-type method with an explicit computation of the gradient to achieve it. This involved mathematical computations using the adjoint method to find an expression of the gradient we were looking for, reflexion on the resolution of the associated optimisation problem and coding. The tests we have carried out so far have given promising results.

The second part of my internship was devoted to becoming familiar with some geometrical tools that we expect to use in future works to build models for Hamiltonian problems. I fulfilled this task in parallel with the first one. As the first task, it was divided into two parts that I carried out one after another. In a first time, I was asked to learn about generating functions, a kind of applications very useful to build symplectic maps. As the central idea of Hamiltonian reduction is to find a good symplectic reduction, this tool may prove useful in future works on the subject. In the field of learning Hamiltonian dynamics with neural networks, this idea has already been exploited in [6]. Then, the objective was to get familiar with techniques based on what we call h-principle. Those techniques are used in symplectic geometry to prove the existence of symplectic embeddings between two manifolds. In our case, we expect to use them to prove results which would give geometrical guaranties to the reduction methods we are interested to build. To complete this task, I read parts of [16] and [2] to learn about generating functions and [9] to learn about h-principle.

What follows is divided into two parts that correspond to the two main tasks I worked on, starting by the numerical part and finishing by the geometrical one. We only present the hyperreduction *via* control approach and the notions we learned about h-principle since it is the parts that are expected to give the better results short term after the internship. What we did on quadratic corrections and generating functions can be found in appendices.

Part I

Numerical part

1 Context

1.1 Hamiltonian systems

In what follows, we consider a symplectic manifold (M, ω) and we study Hamiltonian systems, i.e. systems of the form

$$\begin{cases} \dot{x} = X_H(x), \\ x(0) = x_0, \end{cases}$$

with $X_H \in \Gamma(M)$ the Hamiltonian vector field defined by

$$\omega(X_H, \cdot) = dH,$$

for $H : M \to \mathbf{R}$ a Hamiltonian function.

If we also consider, in addition to the symplectic structure induced by ω on M, a Riemannian structure induced by a metric $\mathbf{g}$ on this same space, the 2-form ω can be formulated in terms of $\mathbf{g}$, like any bilinear form: for all $x \in M$, there exists A_{ω_x} such that for all $u, v \in T_xM$

$$\omega_x(u, v) = \mathbf{g}(A_{\omega_x} u, v).$$

The matrix A must be skew-symmetric and non-degenerate. Note that $\mathbf{g}$ introduces another structure on M that is not necessary for our purpose. Nevertheless, when we work on $\mathbf{R}^{2n}$, using the Euclidean structure simplifies computations.

We first consider $M = \mathbf{R}^{2n}$ with the standard symplectic form $\omega = d\mathbf{p} \wedge d\mathbf{q}$ and the Euclidean structure induced by the standard scalar product $\langle \cdot, \cdot \rangle$. In this case, for all $u, v \in \mathbf{R}^{2n}$,

$$\omega(u, v) = \langle \mathbf{J}_{2n} u, v \rangle,$$

where

$$\mathbf{J}_{2n} = \begin{pmatrix} 0 & -I_n \\ I_n & 0 \end{pmatrix}.$$

The previous system is then rewritten as

$$\dot{x} = \mathbf{J}_{2n} \nabla H(x). \tag{1.1}$$

If we write $x(t) = (p(t), q(t))$, this is equivalent to

$$\begin{cases} \dot{p} = -\dfrac{\partial H}{\partial q}(p, q), \\ \dot{q} = \dfrac{\partial H}{\partial p}(p, q). \end{cases}$$

1.2 Hamiltonian reduction

1.2.1 Goals

We consider the problem

$$\begin{cases} \dot{x}_g(t) = X_{H_g}(x_g(t)), \ \forall t \in [0,1] \\ x_g(0) = x_0(g) \end{cases} \tag{1.2}$$

where x_g has value in M and $g \in G$ is a parameter for the equation which can affect the Hamiltonian function as well as the initial condition.

Here, we will consider partial differential equations, where the Hamiltonian H_g has the form $H_g(x) = \int_{u \in \Omega} f(u,x)du$, where Ω is a subset of $\mathbf{R}^d$ for a certain $d \in \mathbf{N}^*$. At each time $t \in [0,1]$, the solution $x_g(t)$ is therefore a function of Ω, a point in an infinite dimensional space. To compute numerically $x_g(t)$ for a given g, we discretise Ω into r points, or cells. The numerical solution at t is then a point $x_g^{num}(t)$ in a space of (finite) dimension $2n = dim(M) \times r$. This way, we reduced the initial PDE problem to an ODE one. To ensure numerical stability and obtain an accurate solution, the integer r is usually chosen quite large. For this reason, the resolution of this problem over $[0,1]$ for a lot of different values of g is expensive in terms of time and computing resources.

However, if G is of low dimension, we expect that the space of solutions

$$\Sigma = \{x \in M \quad | \quad \exists g \in G, t \in [0,1] \mid x = x_g(t)\}$$

is also of low dimension. However, we still need to compute them in high dimension, which is the space where our numerical solvers work. The aim of a reduce order model is to compute the solutions in the low dimensional space Σ without mention of the high dimensional space. For that, we need $f : \Sigma \times G \to T\Sigma$, $\hat{x}_0 : G \to \Sigma$ and $D : \Sigma \to \mathbf{R}^{2n}$ such that the solutions $\hat{x}_g : [0,1] \to \Sigma$ of

$$\begin{cases} \dot{\hat{x}}(t) = f(\hat{x}(t), g), \ \forall t \in [0,1] \\ \hat{x}(0) = \hat{x}_0(g) \end{cases} \tag{1.3}$$

verify $D(\hat{x}_g(t)) = x_g^{num}(t)$ for all $t \in [0,1]$.

To achieve the reduction, we start by computing some solutions of the original problem (1.2) in high dimension. From these data, we then look for Σ, D and f. In short, building a reduced order model consists in extracting a low dimensional dynamics from a data set.

Here, we focus on reduced models such that (1.3) is also in Hamiltonian form. We then suppose that Σ is a $2k$-dimensional manifold, that we will denote by Σ^{2k}, for $k \ll n$, and we look for a function f such that

$$f(\cdot, g) = X_{\hat{H}_g} = \mathbf{J}_{2k} \nabla \hat{H}_g$$

for a certain parametrised Hamiltonian function $\hat{H}_g : \Sigma^{2k} \to \mathbf{R}$. The problem (1.3) then becomes:

$$\begin{cases} \dot{\hat{x}} = X_{\tilde{H}_g}(\hat{x}), \ \forall t \in [0,1] \\ \hat{x}(0) = \hat{x}_0(g). \end{cases} \tag{1.4}$$

Remark 1.1. The reason why we look for another Hamiltonian system in low dimension is that previous works in the field of reduction for Hamiltonian systems showed that the induced reduction usually gives better results in terms of stability and accuracy (see [1]). Here is

a completely informal discussion about why this could not be surprising. In fact, it seems reasonable since we then compute trajectories in low dimension using the same kind of rules than in high dimension. In particular, we know that the Hamiltonian is conserved along trajectories and this property is physically important since the Hamiltonian usually represents the energy. If the Hamiltonian in low dimension is well chosen, its conservation along trajectories in low dimension may be a guaranty that the energy actually does not vary in the high dimensional space. By conserving the geometrical structure on the space of solutions, the tools available to describe them are still at our disposal. Then, numerical solvers specially created to be stable while integrating the solutions in high dimension has good chances to work also in the low dimensional space. However, we have to note that we may not catch the actual dimension of the solution manifold, which can even be odd. We therefore look for a symplectic manifold Σ^{2k} of very low dimension, on which the original problem has a Hamiltonian formulation.

For the reduction, we need a proper low dimensional symplectic manifold (Σ^{2k}, η). In the methods we will present here, we always first start by building Σ^{2k} and we always build it as a submanifold of $(\mathbf{R}^{2n}, \omega_{2n})$ where ω_{2n} is the usual symplectic form on $\mathbf{R}^{2n}$. In particular, the symplectic form η will be the restriction to Σ^{2k} of ω_{2n}. This implies that Σ^{2k} is a symplectic submanifold and that the inclusion $i : \Sigma^{2k} \to \mathbf{R}^{2n}$ is symplectic.

From now on, we assume that Σ^{2k} is symplectically parametrised by $\mathbf{R}^{2k}$. We call *decoder* the map $D : \mathbf{R}^{2k} \to \mathbf{R}^{2n}$ which associates to the coordinates of a point $\hat{x}$ in Σ^{2k} the point $i(x)$ in $\mathbf{R}^{2n}$ and *encoder* the map E which associates its coordinates to a point of Σ^{2k} as submanifold embedded in $\mathbf{R}^{2n}$. In practice, we choose the submanifold Σ^{2k} by the mean of D, which is the object that we actually build.

1.2.2 Reduced model

The assumptions we have made imply that D is isosymplectic and then that $D^*\omega_{2n} = \omega_{2k}$, which is equivalent to

$$\omega_{2n}(d_{\hat{x}}D(u), d_{\hat{x}}D(u)) = \omega_{2k}(u, v)$$

for all $\hat{x} \in \mathbf{R}^{2k}$ and all $u, v \in \mathbf{R}^{2k}$. Using the expression of ω_{2n} and ω_{2k} in terms of the scalar products on $\mathbf{R}^{2n}$ and $\mathbf{R}^{2k}$, we immediately find that a necessary and sufficient condition for $d_{\hat{x}}D$ to preserve the Hamiltonian structure is given by

$$ {}^t d_{\hat{x}}D \mathbf{J}_{2n} d_{\hat{x}}D = \mathbf{J}_{2k} \quad \forall \hat{x} \in \mathbf{R}^{2k}. \tag{1.5}$$

Equation (1.1) can be rewritten as

$$\nabla D(\hat{x})\dot{\hat{x}} = \mathbf{J}\nabla H(D(\hat{x})).$$

From the definition of the gradient in $\mathbf{R}^{2n}$, we get $\nabla(H \circ D)(\hat{x}) = {}^t\nabla D(\hat{x})\nabla H(D(\hat{x}))$ for all $\hat{x} \in \mathbf{R}^{2k}$, where $\nabla D(\hat{x})$ represents the Jacobian matrix of D at $\hat{x} \in \mathbf{R}^{2k}$. Then, if we multiply the previous equation by ${}^t\mathbf{J}_{2k}{}^t\nabla D(\hat{x})\mathbf{J}_{2n}$, the condition on $d_{\hat{x}}D$ gives

$$\begin{aligned}
&{}^t\mathbf{J}_{2k}{}^t\nabla D(\hat{x})\mathbf{J}_{2n}\nabla D(\hat{x})\dot{\hat{x}} = {}^t\mathbf{J}_{2k}{}^t\nabla D(\hat{x})\mathbf{J}_{2n}\mathbf{J}_{2n}\nabla H(D(\hat{x})),\\
\iff &{}^t\mathbf{J}_{2k}\mathbf{J}_{2k}\dot{\hat{x}} = {}^t\mathbf{J}_{2k}{}^t\nabla D(\hat{x})(-I_{2n})\nabla H(D(\hat{x})),\\
\iff &\dot{\hat{x}} = \mathbf{J}_{2k}\nabla(H \circ D)(\hat{x}).
\end{aligned}$$

The original problem thus takes a Hamiltonian form in the low dimensional space:

$$\dot{\hat{x}} = X_{H\circ D}(\hat{x}).$$

When we have D and Σ^{2k}, we have not finished the reduction. If we want to compute the solutions in the low dimensional space without coming back at each step to the high dimensional one, we need to find an expression, or at least an approximation, of $H \circ D$. This last step is called the *hyperreduction*.

1.2.3 Linear reduction

First, let us assume that the equation depends linearly on the parameters and on time. We then look for a linear reduction, in other words, for a matrix $A \in \mathcal{M}_{2n,2k}(\mathbf{R})$ which preserves the Hamiltonian structure. In this case, the symplecticity condition (1.5) becomes

$$ {}^tA\mathbf{J}_{2n}A = \mathbf{J}_{2k}. $$

Symplecticity conditions for linear maps

A matrix $A \in \mathcal{M}_{2n,2k}(\mathbf{R})$ is said to be *symplectic* if it satisfies the above condition. If we decompose A into four submatrices A_1, A_2, A_3, A_4 in $\mathcal{M}_{n,k}(\mathbf{R})$ such that

$$ A = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}, $$

we have

$$ \begin{aligned} {}^tA\mathbf{J}_{2n}A &= \begin{pmatrix} {}^tA_1 & {}^tA_3 \\ {}^tA_2 & {}^tA_4 \end{pmatrix} \begin{pmatrix} 0 & -I_n \\ I_n & 0 \end{pmatrix} \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} = \begin{pmatrix} {}^tA_1 & {}^tA_3 \\ {}^tA_2 & {}^tA_4 \end{pmatrix} \begin{pmatrix} -A_3 & -A_4 \\ A_1 & A_2 \end{pmatrix} \\ &= \begin{pmatrix} {}^tA_3A_1 - {}^tA_1A_3 & {}^tA_3A_2 - {}^tA_1A_4 \\ {}^tA_4A_1 - {}^tA_2A_3 & {}^tA_4A_2 - {}^tA_2A_4 \end{pmatrix}. \end{aligned} \tag{1.6} $$

Then, A is symplectic if and only if ${}^tA_3A_1$ and ${}^tA_4A_2$ are symmetric and ${}^tA_4A_1 - {}^tA_2A_3 = I_k$.

Encoder and symplectic inverse

When a linear map links two spaces that do not have the same dimensions, it is hopeless to try to inverse it. However, when it has full rank, it admits a left or right inverse which takes a simple form in some cases. For example, it is the transposed for orthogonal matrices. It happens that in the symplectic case, we also have a simple expression for the left or right inverse.

We define the *symplectic inverse* of a matrix $A \in \mathcal{M}_{2n,2k}$ as

$$ A^+ := {}^t\mathbf{J}_{2k}{}^tA\mathbf{J}_{2n}. $$

It is easy to check that if A is symplectic, then $A^+A = I_{2k}$ and ${}^t(A^+)$ is symplectic: if A is symplectic, then using the fact that $\mathbf{J}_{2k}$ is orthogonal,

$$ A^+A = {}^t\mathbf{J}_{2k}{}^tA\mathbf{J}_{2n}A = {}^t\mathbf{J}_{2k}\mathbf{J}_{2k} = I_{2k}. $$

Similarly,

$$ A^+\mathbf{J}_{2n}{}^tA^+ = ({}^t\mathbf{J}_{2k}{}^tA\mathbf{J}_{2n})\mathbf{J}_{2n}({}^t\mathbf{J}_{2n}A\mathbf{J}_{2k}) = {}^t\mathbf{J}_{2k}({}^tA\mathbf{J}_{2n}A)\mathbf{J}_{2k} = {}^t\mathbf{J}_{2k}\mathbf{J}_{2k}\mathbf{J}_{2k} = \mathbf{J}_{2k}. $$

Since it is a left inverse of A and since it is symplectic if A is symplectic, then it is a reasonable choice for the encoder if A has been chosen to be the decoder.

Proper Symplectic Decomposition

To perform a linear reduction, the idea is to find a symplectic matrix $A \in \mathcal{M}_{2n,2k}(\mathbf{R})$ such that $A\hat{x}$ is as close as possible to x, where $\hat{x}$ is the solution of the reduced problem induced by A and x is the solution of the initial problem. To do this, we first compute the solution of the initial problem for some values of the parameters and time, and we evaluate the difference between these solutions and their images after encoding and decoding, i.e.

$$\|S - AA^+S\|_F. \tag{1.7}$$

We would like to minimise this loss. However, as an examination of the symplecticity conditions shows, the set of symplectic matrices is not bounded. Therefore, this optimisation problem does not admit an explicit solution. Several methods have been proposed to find an optimal A under additional constraints. We present here the Proper Symplectic Decomposition, also known as cotangent lift or complex SVD, which is an adaptation of the Proper Orthogonal Decomposition in the symplectic case.

We restrict the space where we look for the minimum of (1.7). The idea is to look for an operator A of the form

$$\begin{pmatrix} \phi & 0 \\ 0 & \phi \end{pmatrix},$$

with $\phi \in \mathcal{O}(n)$. As shown in [18], this is in fact equivalent to choosing the k columns of ϕ among the $\{q_1, ..., q_m, p_1, ..., p_m\}$ using classical POD.

1.3 Application to a piano vibrating string

In this work, we test the reduction on a set of equations modelling a piano string vibration, proposed in [5].

1.3.1 The model

We consider the following problem:

$$\begin{cases} \partial_{tt}^2 U(x,t) = \partial_x\Big[\nabla V(\partial_x U(x,t))\Big] & \forall (x,t) \in \Omega \times \mathbf{R}_+ \\ U(x,0) = U_0(x) & \forall x \in \Omega, \\ \partial_t U(x,0) = U_1(x) & \forall x \in \Omega, \\ U(x,t) = 0 & \forall (x,t) \in \partial\Omega \times \mathbf{R}_+. \end{cases}$$

In what follows, $U(x,t) = (u(x,t), v(x,t))$ represents the longitudinal and transverse variations of the position of the point x in a piano string on the oscillation plane. The domain Ω represents the string and will be here the interval $[0,1]$. We study the case where we suddenly release the string after appling a sinusoidal deformation to it. This configuration is modeled by setting, in the previous system, $U_1 \equiv 0$ and U_0 of the form $x \mapsto (l_1 \sin(2\pi x), l_2 \sin(2\pi x))$. In our work, we will take (l_1, l_2) in $[0, 0.2]^2$.

Set $q = U = (u, v)$ and $p = (\partial_t u, \partial_t v)$. The previous equation is rewritten

$$\begin{cases} \dfrac{\partial q}{\partial t} = p, \\ \dfrac{\partial p}{\partial t} = \partial_x\Big[\nabla V(\partial_x q)\Big]. \end{cases}$$

It has a Hamiltonian formulation with the energy function

$$H(p,q,t) = \int_\Omega \frac{1}{2}|p|^2 + V(\partial_x q)dx.$$

Indeed, on the one hand

$$H(p+p',q,t) = \int_\Omega \frac{1}{2}|p|^2 + V(\partial_x q)dx + \int_\Omega p\cdot p'dx + \int_\Omega \frac{1}{2}|p'|^2dx = H(p,q,t) + \langle p,p'\rangle + o(|p'|),$$

from which

$$\nabla_p H(p,q,t) = p.$$

On the other hand,

$$\begin{aligned} H(p,q+q',t) &= \int_\Omega \frac{1}{2}|p|^2 + V(\partial_x q)dx + \int_\Omega \nabla V(\partial_x q)\cdot\partial_x q'dx + \int_\Omega o(|\partial_x q'|) \\ &= H(p,q,t) + \int_\Omega \nabla V(\partial_x q)\cdot\partial_x q'dx + o(|q'|). \end{aligned}$$

After an integration by parts, since by hypothesis q' is zero on $\partial\Omega$, we find

$$\int_\Omega \nabla V(\partial_x q)\cdot\partial_x q'dx = -\int_\Omega \partial_x\nabla V(\partial_x q)\cdot q'dx,$$

from which

$$\nabla_q H(p,q,t) = -\partial_x\nabla V(\partial_x q).$$

The Hamiltonian is separated. The term in p represents the kinetic energy and the term in q the potential one. To solve this problem in high dimension, we can use explicit Störmer-Verlet symplectic solver, given in [12].

1.3.2 Application of the PSD

One of the potential energies proposed in [5] induces a linear model. As we have seen in a previous work (see my M1 internship report), the PSD works well in this case. As we see on Figures 1.2 and 1.1, this is not the case for a choice of V which leads to a non-linear model. This work aims to find reduction methods that improve the results we get with the PSD. We therefore not consider the linear model and focus on the non-linear one, that we present now.

We consider

$$V(u,v) = \frac{1-\alpha}{2}u^2 + \frac{1}{2}v^2 + \frac{\alpha}{2}(u^2v + \frac{1}{4}u^4),$$

where $\alpha \in [0,1]$ is a parameter which depends on the characteristics of the string. More precisely, it is $\frac{EA-T_0}{EA}$, where E, A and T_0 are respectively Young's modulus, the section area and the tension at rest of the string. The solution of this Hamiltonian system depends on three parameters: α, which parametrises the Hamiltonian function, and the pair (l_1,l_2), which parametrises the initial condition. In what follows, we will either fix α or (l_1,l_2) and so take $g = (l_1,l_2) \in G = [0,0.2]^2$ or $g = \alpha \in G = [0,1]$.

This potential yields the following system:

$$\begin{cases} \partial^2_{tt}u = \partial_x\Big[(1-\alpha)\partial_x u + \alpha(\partial_x u\partial_x v + \frac{1}{2}(\partial_x u)^3)\Big], \\ \partial^2_{tt}v = \partial_x\Big[\partial_x v + \frac{\alpha}{2}(\partial_x u)^2\Big]. \end{cases}$$

Numerically, we approach first and second derivatives with finite differences:

$$\partial_x u(x^i) \approx \frac{u(x^{i+1}) - u(x^i)}{\Delta x}$$

and

$$\partial_x^2 u(x^i) \approx \frac{u(x^{i+1}) - 2u(x^i) + u(x^{i-1})}{\Delta x^2}.$$

During last year's internship, we tested the reduced order model obtained with the PSD for $G = [0, 1]$ and $g = \alpha$. As we see on Figure 1.2, it gives inaccurate solutions for $k = 3$. The bumps are too sharp on the position computed with PSD and the variations of speed through time is completely wrong. Looking at H^1 errors and energy, we see that the model built with PSD produces an unstable solution.

When we take $k = 10$, these problems are still visible. To obtain a satisfactory solution with the reduced model, one should increase the reduced space dimension and take $k = 20$. This in fact means that the reduction failed because we did not succeed in capturing the low dimensional structure of the problem. We can deduce that the problem we want to solve here is too far from being linear to admit a linear reduction. If we still want to use PSD, we thus have to look further to improve the reduction.

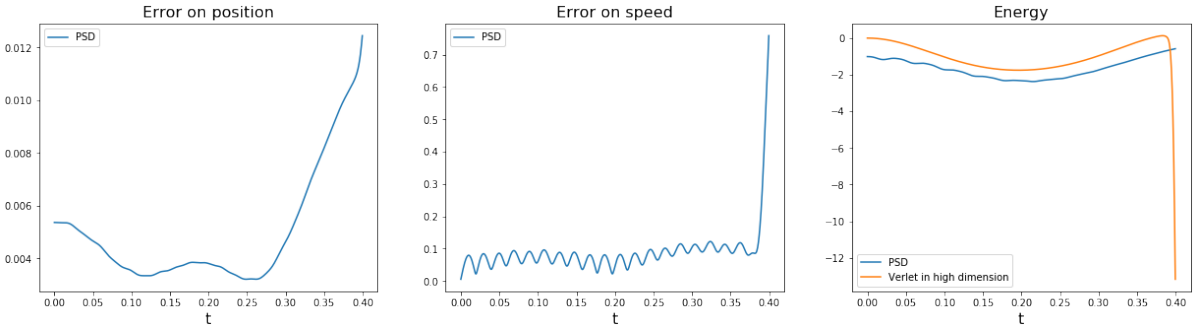


Figure 1.1: Estimation of the errors made on the string position (left) and speed (middle) through time and the variations of energy during time (bottom). The reference solution is the solution we computed in high dimension. The estimation of error was made on the trajectories computed for $g = 0.99$. The reduction was made using PSD with $k = 3$ using 10 trajectories, computed in high dimension with explicit Stormer-Verlet scheme, for values of g uniformly sampled in $I = [0, 1]$, $n = 200$, $dt = 0.0005$ and $m = 800$. Trajectories in low dimension were computed using the reduced model with the same discretisation in time and space. Initial speed in high dimension was 0 for all $x \in [0, 1]$ and initial displacement was given by $x \mapsto (0.1 \sin(2\pi x), 0.05 \sin(2\pi x))$.

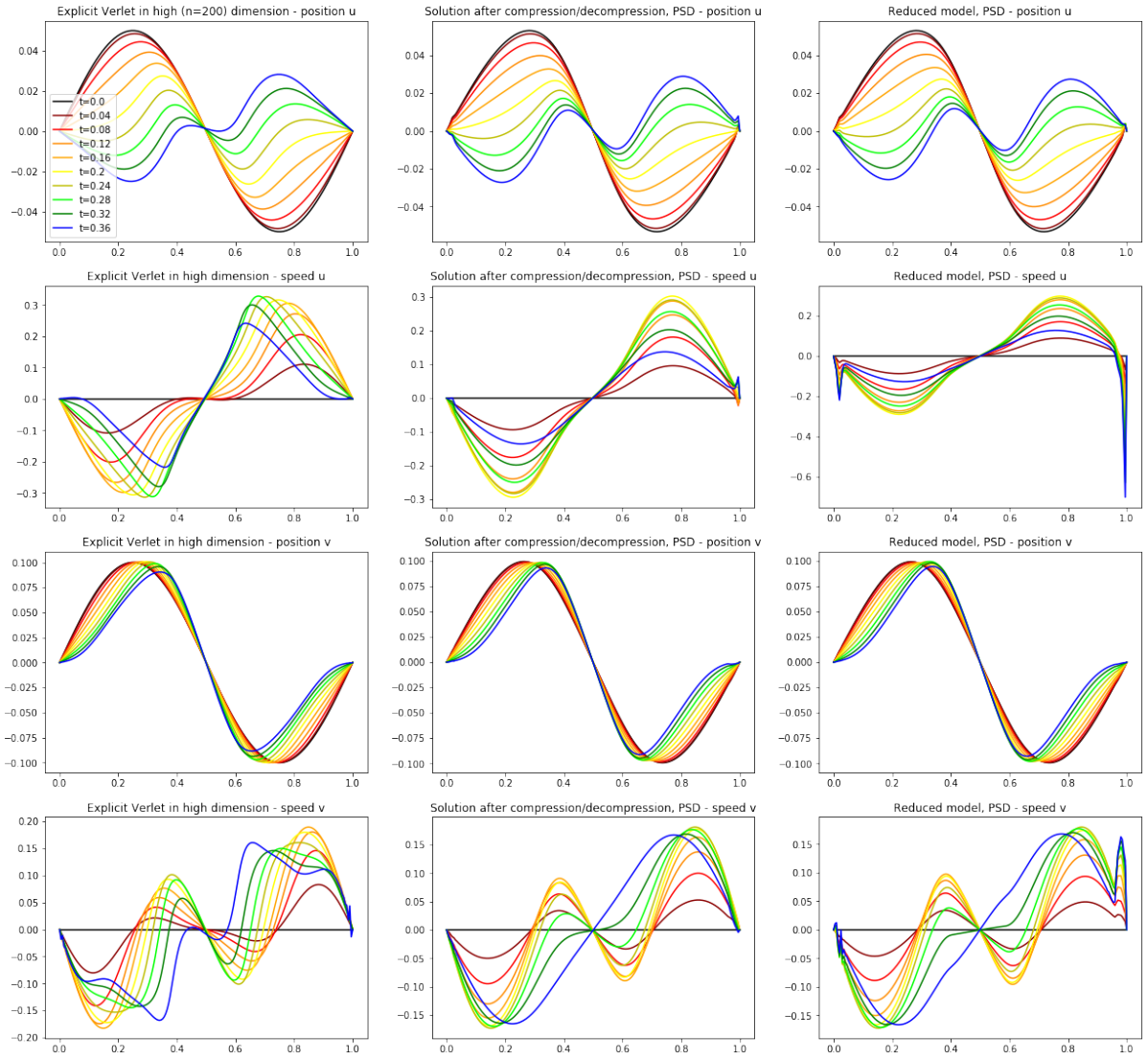


Figure 1.2: Numerical solution of non-linear piano string equation with $g = 0.99$. The first column represents the piano string position and speed for different times computed in high dimention. The second one represents the trajectory we obtain after compression and decompression of the previous ones. The third column represents the decompressed trajectory we obtained with the reduced model. On the first line, we show the horizontal speed of the string in function of the position x on the string. On the second line is the horizontal displacement of the string in function of x. On the third line is the vertical speed in function of x. On the fourth line is the vertical displacement of the string in function of x. The reduction was made using PSD with $k = 3$ using 10 trajectories, computed in high dimension with explicit Stormer-Verlet scheme, for values of g uniformly sampled in $I = [0, 1]$, $n = 200$, $dt = 0.0005$ and $m = 800$. Trajectories in low dimension were computed using the reduced model with the same discretisation in time and space. Initial speed in high dimension was 0 for all $x \in [0, 1]$ and initial displacement was given by $x \mapsto (0.1 \sin(2\pi x), 0.05 \sin(2\pi x))$. The value of the parameter g we have taken to compute the solution is not in the set sampled to build the PSD.

2 Hyperreduction with an optimal control approach

In this chapter, we suppose that we have already performed a Proper Symplectic Decomposition or any other linear reduction method that gives a decoding map $D : \mathbf{R}^{2k} \to \mathbf{R}^{2n}$ from the low dimensional space, where we want to compute the solutions of the studied Hamiltonian system, to the high dimensional space, where we originally compute them. We also suppose that the same method gives us an encoding map $E : \mathbf{R}^{2n} \to \mathbf{R}^{2k}$ to compress solutions from the high dimensional space to the low dimensional one. Recall that the Hamiltonian system is supposed to be parametrised by a certain $g \in G$ and that this parametrisation can affect the Hamiltonian function H and the initial condition x_0. In the following, they will be denoted H_g and $x_0(g)$.

In the original PSD approach, we then compute solutions in low dimension using the Hamiltonian obtained by composition of the Hamiltonian H_g of the original, high-dimensional, problem with the decoder. Of course, computing solutions this way is as costly as computing them in the classical high-dimensional way. This is why we usually add a hyperreduction step to the reduction, which consists in finding a Hamiltonian function in low dimension which interpolates $H_g \circ D$.

Here, we take another approach, which we hope to be more efficient and to give more accurate results. Instead of interpolating $H_g \circ D$, we directly want to find the Hamiltonian in the low dimensional space that gives the more accurate trajectories, that is trajectories which, when decompressed in the high dimensional space, are the closest to the those that are computed in high dimension. This is achieved using an optimal control approach. In the field of learning Hamiltonian dynamics with neural networks, this kind of method have been used in [7, 19, 15] for instance. Here, we use a method of type Sparse Identification of Non-linear Dynamics (SINDy), proposed in [3]. It briefly consists in taking the target function $H_{\theta,g} = \sum_{i=1}^{K} \theta_i f_i$ in the space V_F spanned by a set of given non-linearities $F = \{f_1, ..., f_K\}$. We would like that for all admissible parameter $g \in G$, the decompressed low dimensional trajectory $D\hat{x}_{\theta,g}$ with $\hat{x}_{\theta,g}$ defined by

$$\begin{cases} \dot{\hat{x}}_{\theta,g}(t) = X_{H_{\theta,g}}(\hat{x}_{\theta,g}(t)) \quad \forall t \in [0,1], \\ \hat{x}_{\theta,g}(0) = Ex_0(g), \end{cases} \tag{2.1}$$

is as close as possible to the high dimensional trajectory x_g defined by

$$\begin{cases} \dot{x}_g(t) = X_{H_g}(x_g(t)) \quad \forall t \in [0,1], \\ x(0) = x_0(g), \end{cases}$$

where $x_0 : G \to \mathbf{R}^{2k}$ is given.

Coefficients $\theta = (\theta_1, ..., \theta_K)$ are then chosen such that the decompressed flow of H_θ and the flow of H are as close as possible, while keeping a lot of coefficients exactly equal to zero. We

detail the application of this method to our problem in the following section.

2.1 Problem setting

Let $F = \{f_i : \mathbf{R}^{2k} \to \mathbf{R}\}_i \in [\![1, K]\!]$ with $K \in \mathbf{N}$ a family of non-linear functions of class $\mathcal{C}^2$ and V_F the finite dimensional Hilbert subspace spanned by F. For $\theta \in \mathbf{R}^K$, denote by $\hat{H}_\theta$ the element of V_F such that $\hat{H}_\theta = \sum_{i=1}^K \theta_i f_i$. In the following, we are looking for the value of the parameter θ which minimises the loss

$$\mathcal{L}(\theta) = \int_{g\in G} \int_{t\in[0,1]} \|D\hat{x}_{\theta,g}(t) - x_g(t)\|_{2k}^2 dg dt,$$

where $\hat{x}_{\theta,g} : [0,1] \to \mathbf{R}^{2k}$ denotes the solution of the problem (2.1).

Recall that G is the set of admissible parameter for the high dimensional Hamiltonian H_g and $x_g : [0,1] \to \mathbf{R}^{2n}$ is the trajectory of $x_0(g)$ along the flow of H_g. We also denote by E and D the linear encoder and decoder built at the previous reduction step using the PSD.

Numerically, a classical method to find a value close to an optimal value, is a gradient descent. To implement it, however, one needs the gradient of the loss $\mathcal{L}$ with respect to the variable θ. The following computations aim to find it.

To simplify further computations, we introduce the loss function

$$\mathcal{L}_g(\theta) := \int_{t\in[0,1]} \|D\hat{x}_{\theta,g}(t) - x_g(t)\|_{2k}^2 dt.$$

Remark 2.1. Note that if we have the gradient of $\mathcal{L}_g$ for all $g \in G$, then we simply obtain the gradient of $\mathcal{L}$ by integration over G. Indeed, for all $\theta \in \mathbf{R}^K$, we have $\mathcal{L}(\theta) = \int_{g\in G} \mathcal{L}_g(\theta) dg$. We suppose that we have already proved that $\mathcal{L}_g$ is differentiable and we assume that G is chosen such that we can derive $\mathcal{L}$ under the integral. Then, for all $h \in \mathbf{R}^K$

$$\begin{aligned} d_\theta \mathcal{L}(h) &= \int_{g\in G} d_\theta \mathcal{L}_g(h) dg = \int_{g\in G} \nabla \mathcal{L}_g(\theta) \cdot h \; dg = \int_{g\in G} \sum_{i=1}^K \partial_i \mathcal{L}_g(\theta) h_i \; dg \\ &= \sum_{i=1}^K h_i \int_{g\in G}^K \partial_i \mathcal{L}_g(\theta) dg = \left\langle \int_{g\in G} \nabla \mathcal{L}_g(\theta) dg, h \right\rangle_K. \end{aligned}$$

We deduce that $\nabla \mathcal{L}(\theta) = \int_{g\in G} \nabla \mathcal{L}_g(\theta) dg$. In what follows, we will therefore restrict our study to the case where $G = \{g_0\}$ and skip mentions of g in our notations.

2.2 In-out application

Denote by $\mathcal{F} : \mathbf{R}^K \to H^1([0,1])^{2k}$ the application which sends a value of the parameter θ to the trajectory $\hat{x}_\theta$ generated by $\hat{H}_\theta$ in low dimension. In the following, it will be called the *in-out application*. Let also $f : H^1([0,1])^{2k} \to \mathbf{R}$ be such that $f(\hat{x}) = \|D\hat{x} - x\|_{L^2([0,1])^{2k}}$. We have $\mathcal{L} = f \circ \mathcal{F}$ so, provided that both f and $\mathcal{F}$ are differentiable, the chain rule gives

$$d_\theta \mathcal{L} = d_{\mathcal{F}(\theta)} f \circ d_\theta \mathcal{F}.$$

If the variational method gives immediately the differential of f, it does not work with $\mathcal{F}$, which is implicitly defined. The strategy that we develop here is a classical method of optimal control. Briefly, it consists in finding what we call an *adjoint state*, build to satisfy a well-chosen ordinary differential equation. Inserted into the expression of $d_\theta \mathcal{L}$, it allows to neutralise the problematic terms.

2.2.1 Regularity of $\mathcal{F}$

In the first step, we find a differential equation for $d_\theta \mathcal{F}$. Before that, we have to prove that $\mathcal{F}$ is at least differentiable. We use here the implicit function theorem to achieve this.

Proposition 2.2. *The in-out application*

$$\mathcal{F} : \begin{cases} \mathbf{R}^K \to H^1([0,1])^{2k} \\ \theta \mapsto \hat{x}_\theta \end{cases}$$

where $\hat{x}_\theta$ is solution of 2.1 for a fixed parameter value $g \in G$, is of class $\mathcal{C}^1$.

Proof. By translating $\mathcal{F}$ by the constant function x_0, we can reduce the problem where $x_0 = 0$.

Consider the application

$$G : \begin{cases} \mathbf{R}^K \times V \to L^2([0,1])^{2k} \\ \quad (\theta, x) \mapsto x' - X_{\hat{H}_\theta}(x) \end{cases}$$

where $V := \{f \in H^1([0,1])^{2k} \quad | \quad f(0) = 0\}$. It is a closed subspace of H^1 so it is a Hilbert space too for the H^1 scalar product.

We will prove that G is of class $\mathcal{C}^1$ but we first need the following result concerning $\theta \mapsto X_{\hat{H}_\theta}$.

Lemma 2.3. *The application $\theta \to X_{\hat{H}_\theta}$ is linear and continuous. More precisely, we have $X_{\hat{H}_h} = {}^t\mathbf{X} h$ with*

$$\mathbf{X} := [X_{f_i}]_{i=1,\dots,K} \in \mathcal{C}^1(\mathbf{R}^{2k}, \mathbf{R}^{2k})^K.$$

Proof. Let us first see that $\theta \to X_{\hat{H}_\theta}$ is linear. Clearly, this is the case of $\theta \to \hat{H}_\theta$. Thanks to the properties of the symplectic form ω, this is also the case of $\hat{H} \to X_{\hat{H}}$: the bilinearity makes that

$$\omega(X_{\hat{H}_1 + \lambda \hat{H}_2}, \cdot) = d_x(\hat{H}_1 + \lambda \hat{H}_2) = d_x \hat{H}_1 + \lambda d_x \hat{H}_2 = \omega(X_{\hat{H}_1}, \cdot) + \lambda \omega(X_{\hat{H}_2}, \cdot) = \omega(X_{\hat{H}_1} + \lambda X_{\hat{H}_2}, \cdot)$$

and the non-degeneracy then ensures that $X_{\hat{H}_1 + \lambda \hat{H}_2} = X_{\hat{H}_1} + \lambda X_{\hat{H}_2}$.

Then, since $X_{\hat{H}_\theta} = \sum_{i=1}^K \theta_i X_{f_i}$, if we introduce $\mathbf{X}$ as in the statement of the lemma, we have $X_{\hat{H}_\theta} = {}^t\mathbf{X}\theta$. This gives

$$\|X_{\hat{H}_h}\|_{L^2} = \int_{x \in \mathbf{R}^{2k}} \|{}^t\mathbf{X}(x) \cdot h\|_{2k} dt \leqslant \|h\|_K \int_{x \in \mathbf{R}^{2k}} |||\mathbf{X}(x)||| dt.$$

As $\mathbf{X}(x)$ lies in $\mathcal{M}_{2k,K}(\mathbf{R})$, which is a finite dimensional space, the operator norm $|||\cdot|||$ and the Froebenius norm are equivalent so

$$\|X_{\hat{H}_h}\|_{L^2} \leqslant cst \ \|h\|_K \int_{x \in \mathbf{R}^{2k}} \|\mathbf{X}(x)\|_F dt = cst \ \|h\|_K \sum_{i=1}^{2k} \sum_{j=1}^{K} \|\mathbf{X}_{ij}\|_{L^2(\mathbf{R}^{2k}, \mathbf{R})}.$$

All the coefficients of $\mathbf{X}$ are continuous functions, so the double sum takes a finite value. □

Let us go back to G. Let θ be a point in $\mathbf{R}^K$ and x a function in $V \subset H^1([0,1])^{2k}$. Let h and y be two small elements of the same spaces. For all $t \in [0,1]$, we have

$$G(\theta+h, x+y)(t) - G(\theta,x)(t) = \left(y'(t) - d_{x(t)}X_{\hat{H}_\theta}(y(t)) - X_{\hat{H}_h}(x(t))\right) \\ - \left(\epsilon_1(y)(t) + d_{x(t)}X_{\hat{H}_h}(y)(t) + \epsilon_2(y)(t)\right),$$

where $\epsilon_1(y)$ and $\epsilon_2(h)$ are respectively in $o(\|y\|_{H^1})$ and $o(\|h\|_K)$.

From the regularity assumption on the elements of $V_F = \mathrm{Span}(F)$ and from Lemma 2.3, we know that the terms in the first parentheses are linear and continuous in (h,y). In the second one, ϵ_1 and ϵ_2 are the remaining terms in the first order Taylor's expansion of $X_{\hat{H}_\theta}$ and $X_{\hat{H}_h}$ so they are in $o(\|y\|_{H^1})$ and *a fortiori* in $o(\|(h,y)\|_{\mathbf{R}^K\times H^1})$. The last term, $d_{x(\cdot)}X_{\hat{H}_h}(y)$, is bilinear and continuous in h and y so is also in $o(\|(h,y)\|_{\mathbf{R}^K\times H^1})$. This proves that G is differentiable and that its differential is given by

$$d_{\theta,x}G(h,y) : t \mapsto y'(t) - d_{x(t)}X_{\hat{H}_\theta}(y(t)) - X_{\hat{H}_h}(x(t)).$$

We now have to see that $dG : \mathbf{R}^K \times V \to \mathcal{L}_c\left(V, L^2([0,1])^{2k}\right)$ is continuous. The term $(y,h) \mapsto y'$ does not depend on x nor θ so it is continuous in this variables. Let us study the term $(y,h) \mapsto d_{x(\cdot)}X_{\hat{H}_\theta}(y(\cdot))$. Let $\bar{y}$ and $\bar{h}$ be two small elements of $H^1([0,1])$ and $\mathbf{R}$. We have

$$|||d_{x+\bar{y}}X_{\hat{H}_{\theta+\bar{h}}} - d_xX_{\hat{H}_\theta}||| \leqslant |||d_{x+\bar{y}}X_{\hat{H}_\theta} - d_xX_{\hat{H}_\theta}||| + |||d_{x+\bar{y}}X_{\hat{H}_{\bar{h}}}|||, \tag{2.2}$$

where $|||\cdot|||$ denotes the operator norm for linear continuous functions between $H^1([0,1])^{2k}$ and $L^2([0,1])^{2k}$.

For the second term in the right-hand side of the last expression, we have

$$|||d_{x+\bar{y}}X_{\hat{H}_{\bar{h}}}||| \leqslant \sum_{i=1}^{K} \bar{h}_i \; |||d_{x+\bar{y}}X_{f_i}|||$$

with $|||d_{x+\bar{y}}X_{f_i}||| \to |||d_xX_{f_i}|||$. In fact, $t \mapsto x(t)$ is continuous on the compact set $[0,1]$ so $x([0,1])$ is a compact set too. Since $x \mapsto d_xX_{f_i}$ is continuous, it is uniformly continuous on $x([0,1])$. This means that

$$\forall \epsilon > 0,\ \exists \delta_{\epsilon,x} \quad | \quad \forall \nu \in \mathbf{R}^{2k},\ \|\nu\|_{2k} < \delta_{\epsilon,x} \implies \sup_{z\in\mathbf{R}^{2k}} \left(\frac{\|d_{x(t)+\nu(t)}X_{f_i}(z) - d_{x(t)}X_{f_i}(z)\|_{2k}}{\|z\|_{2k}}\right) \leqslant \epsilon.$$

In particular, if $y(t)$ satisfies the condition on ν and $z = \bar{y}(t)$

$$\|d_{x(t)+\bar{y}(t)}X_{f_i}(y(t)) - d_{x(t)}X_{f_i}(y(t))\|_{2k} \leqslant \epsilon\|y(t)\|_{2k}$$

for all $y \in V$ and all $t \in [0,1]$. Passing to the L^2 norm, we get

$$\|d_{x+\bar{y}}X_{f_i}(y) - d_xX_{f_i}(y)\|_{L^2} \leqslant \epsilon\|y\|_{L^2} \leqslant \epsilon\|y\|_{H^1}$$

for all $y \in V$, which gives

$$\sup_{y\in H^1([0,1])^{2k}} \left(\frac{\|d_{x+\bar{y}}X_{f_i}(y) - d_xX_{f_i}(y)\|_{L^2}}{\|y\|_{H^1}}\right) \leqslant \epsilon$$

provided that $\|\bar{y}\|_\infty \leqslant \delta_{\epsilon,x}$. As $H^1([0,1])^{2k}$ is continuously injected in $\mathcal{C}^0([0,1])^{2k}$, this condition is satisfied when $\|\bar{y}\|_{H^1}$ is small enough. Since ϵ can be chosen arbitrarily small, this proves that

$\|\!|d_{x+\bar{y}}X_{f_i} - d_xX_{f_i}|\!\|$ tends to 0 when $\|\bar{y}\|_{H^1} \to 0$. Then, $\|\!|d_{x+\bar{y}}X_{\hat{H}_{\bar{h}}}|\!\|$ also tends to zero as $(\bar{h},\bar{y}) \to 0$ and the first term of (2.2) is continuous at (x,θ). Replacing X_{f_i} by $X_{\hat{H}_\theta}$, the same argument shows that the first term in the right-hand side of (2.2) is also continuous at (x,θ).

Almost the same argument also shows that the term $(y,h) \mapsto X_{\hat{H}_h} \circ x$ is continuous at (x,θ). In fact, we have

$$\begin{aligned}
\|\!|X_{H_\cdot}(x+\bar{y}) - X_{H_\cdot}(x)|\!\| &= \sup_{h\in\mathbf{R}^K}\left(\frac{\|X_{H_h}(x+\bar{y}) - X_{H_h}(x)\|_{L^2}}{\|h\|_{\mathbf{R}^K}}\right)\\
&= \sup_{h\in\mathbf{R}^K}\left(\frac{\|\sum_{i=1}^K h_i\big(X_{f_i}(x+\bar{y}) - X_{f_i}(x)\big)\|_{L^2}}{\|h\|_{\mathbf{R}^K}}\right)\\
&\leqslant \max_{i\in[\![1,K]\!]}\|X_{f_i}(x+\bar{y}) - X_{f_i}(x)\|_{L^2}.
\end{aligned}$$

For all $i \in [\![1,K]\!]$, X_{f_i} is uniformly continuous on the compact set $x([0,1])$ so

$$\forall\epsilon>0, \exists\delta_{\epsilon,x}>0 \quad | \quad \forall\nu\in\mathbf{R}^{2k}, \|\nu\|_{2k}<\delta_{\epsilon,x} \implies \Big(\forall t\in[0,1]\|X_{f_i}(x(t)+\nu) - X_{f_i}(x(t))\|_{2k} < \epsilon\Big).$$

If $\|\bar{y}(t)\|_{2k} < \delta_{\epsilon,x}$ for all $t \in [0,1]$, then passing the previous inequality with $\nu = \bar{y}(t)$ to the L^2 norm, we obtain $\|X_{f_i}(x+\bar{y}) - X_{f_i}(x)\|_{L^2} < \epsilon$. Then, the continuous inclusion of $H^1([0,1])^{2k}$ in $\mathcal{C}^0([0,1])^{2k}$ makes that $\|\bar{y}\|_{H^1} \to 0$ implies $\|\bar{y}\|_\infty \to 0$ and $\|X_{f_i}(x+\bar{y}) - X_{f_i}(x)\|_{L^2} \to 0$.

This achieves to prove the continuity of dG at (x,θ) and since it is true for all element (x,θ) of $V \times \mathbf{R}^K$, dG is continuous on $V \times \mathbf{R}^K$ and G is of class $\mathcal{C}^1$.

Now, the differential of G in the direction x at $\hat{x}_\theta$, which is given by

$$y \mapsto y' - d_xX_{\hat{H}_\theta}(y)$$

is bijective. In fact, for all $m \in L^2([0,1])^{2k}$, the system

$$\begin{cases} y' = d_xX_{\hat{H}_\theta}(y) + m,\\ y(0) = 0 \end{cases}$$

has an unique solution by Cauchy-Lipschiz's theorem. Then, applying the implicit function theorem, there exists an open neighbourhood $\Omega = \Omega_1 \times \Omega_2$ of any $(\theta, \hat{x}_\theta)$ in $\mathbf{R}^K \times V$ and a function $\phi : \mathbf{R}^K \to V$ of class $\mathcal{C}^1$ such that

$$((h,y)\in\Omega \wedge G(h,y) = 0) \quad \iff \quad (\theta\in\Omega_1 \wedge y = \phi(h)).$$

But if $G(h,y) = 0$, it means that $\mathcal{F}(h) = y$ so $\mathcal{F} = \phi$ on Ω_1. Hence, $\mathcal{F}$ is $\mathcal{C}^1$ around θ and since θ has been arbitrarily chosen, it also means that $\mathcal{F}$ is $\mathcal{C}^1$ on $\mathbf{R}^K$. □

2.2.2 Characterisation of $d_\theta\mathcal{F}$

Now that we have seen that $d_\theta\mathcal{F}$ is well-defined and continuous, we characterise it with a differential problem in $[0,1]$.

Proposition 2.4. *The differential of $\mathcal{F}$ at θ in the direction h is solution of the following Cauchy's system*

$$\begin{cases} \dot{z}(t) = d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}(z(t)) + X_{\hat{H}_h}(\mathcal{F}(\theta)(t)) \quad \forall t\in[0,1],\\ z(0) = 0. \end{cases} \tag{2.3}$$

Proof. Consider the equation

$$\mathcal{F}(\theta+h)-\mathcal{F}(\theta)=d_\theta\mathcal{F}(h)+\epsilon_1(h)$$

for some infinitesimal h in $\mathbf{R}^K$ and derive it with respect to time. This leads to

$$X_{\hat{H}_{(\theta+h)}}(\mathcal{F}(\theta+h)(t))-X_{\hat{H}_\theta}(\mathcal{F}(\theta)(t))=\frac{d}{dt}d_\theta\mathcal{F}(h)(t)+\frac{d}{dt}\epsilon_1(h)(t)$$

$$\iff X_{\hat{H}_\theta}(\mathcal{F}(\theta+h)(t))+X_{\hat{H}_h}(\mathcal{F}(\theta+h)(t))-X_{\hat{H}_\theta}(\mathcal{F}(\theta)(t))=\frac{d}{dt}d_\theta\mathcal{F}(h)(t)+\frac{d}{dt}\epsilon_1(h)(t) \quad (2.4)$$

for all t in $[0,1]$.

Now, developing $X_{\hat{H}_\theta}$, $X_{\hat{H}_h}$ and $\mathcal{F}$ at first order, the left hand side becomes

$$\begin{aligned}
&X_{\hat{H}_\theta}\Big(\mathcal{F}(\theta)(t)\Big)+d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}\Big(d_\theta\mathcal{F}(h)(t)\Big)+d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}\Big(\epsilon_1(h)(t)\Big)+\epsilon_2\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big)\\
&+X_{\hat{H}_h}\Big(\mathcal{F}(\theta)(t)\Big)+d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_h}\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big)+\epsilon_3\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big)\\
&-X_{\hat{H}_\theta}\Big(\mathcal{F}(\theta)(t)\Big)
\end{aligned}$$

for all $t\in[0,1]$. Rearranging the terms, we get

$$\begin{aligned}
&\left[d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}\Big(d_\theta\mathcal{F}(h)(t)\Big)+X_{\hat{H}_h}\Big(\mathcal{F}(\theta)(t)\Big)\right]+\left[d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}\Big(\epsilon_1(h)(t)\Big)\right.\\
&\left.+\epsilon_2\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big)+d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_h}\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big)+\epsilon_3\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big)\right].
\end{aligned}$$

The function

$$m:h\in\mathbf{R}^K\mapsto\Big(t\mapsto d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}\Big(d_\theta\mathcal{F}(h)(t)\Big)+X_{\hat{H}_h}\Big(\mathcal{F}(\theta)(t)\Big)\Big)\in L^2([0,1])^{2k}$$

is linear and continuous as sum and compositions of such functions:

$$\begin{aligned}
\|m(h)\|_{L^2}^2&\leqslant\int_0^1\|d_{x_\theta(t)}X_{\hat{H}_\theta}(d_\theta\mathcal{F}(h)(t))\|_{2k}^2dt+\int_0^1\|{}^t\mathbf{X}(\mathcal{F}(\theta)(t))h\|_{2k}^2dt\\
&\leqslant\int_0^1|||d_{x_\theta(t)}X_{\hat{H}_\theta}|||^2\;|||d_\theta\mathcal{F}(t)|||^2\;\|h\|_K^2dt+\int_0^1|||\mathbf{X}(\mathcal{F}(\theta)(t))|||^2\|h\|_K^2dt\\
&=cst\;\|h\|_K^2.
\end{aligned}$$

Lemma 2.5. *The other terms, that is*

$$t\mapsto d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}\Big(\epsilon_1(h)(t)\Big)+\epsilon_{2+3}\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big)+d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_h}\Big(d_\theta\mathcal{F}(h)(t)+\epsilon_1(h)(t)\Big),$$

with $\epsilon_{2+3}=\epsilon_2+\epsilon_3$, are in $o(\|h\|_K)$ for the L^2 norm.

Proof. By continuity of $d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}$ for all $t\in[0,1]$

$$\Big\|d_{\mathcal{F}(\theta)(\cdot)}X_{\hat{H}_\theta}\Big(\epsilon_1(h)(\cdot)\Big)\Big\|_{L^2}^2=\int_0^1\|d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}\Big(\epsilon_1(h)(t)\Big)\|_{2k}^2dt\leqslant\int_0^1|||d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta}|||\;\|\epsilon_1(h)(t)\|_{2k}^2dt.$$

As $\hat{H}_\theta$ is supposed to be of class $\mathcal{C}^2$, $x \mapsto d_x X_{\hat{H}_\theta}$ is continuous. Since $t \mapsto \mathcal{F}(\theta)(t)$ is continuous, we finally have that $t \mapsto |||d_{\mathcal{F}(\theta)(t)} X_{\hat{H}_\theta}|||$ is continuous on $[0,1]$ and therefore bounded. This gives

$$\Big\| d_{\mathcal{F}(\theta)(\cdot)} X_{\hat{H}_\theta}\Big(\epsilon(h)(\cdot)\Big)\Big\|^2_{L^2} \leqslant cste \int_0^1 \|\epsilon_1(h)(t)\|^2_{2k} dt = cste\ \|\epsilon_1(h)\|^2_{L^2}.$$

As ϵ_1 is in $o(\|h\|_K)$ for the L^2 norm, so is $t \mapsto d_{\mathcal{F}(\theta)(t)} X_{\hat{H}_\theta}\Big(\epsilon_1(h)(t)\Big)$.

In the second non-linear term, the fact that $\epsilon_{2+3}(x)$ is in $o(\|x\|_{2k})$ when x tends to zero can be written as

$$\forall \epsilon > 0,\ \exists \delta_\epsilon > 0 \mid \forall x \in \mathbf{R}^{2k},\ \|x\| < \delta_\epsilon \implies \|\epsilon_{2+3}(x)\|_{2k} < \epsilon \|x\|_{2k}. \tag{2.5}$$

When h tends to zero in $\mathbf{R}^K$, $d_\theta \mathcal{F} \cdot h + \epsilon_1(h)$ tends to zero in $H^1([0,1])$, which is continuously injected in $\mathcal{C}^0([0,1])$. Then,

$$\forall \delta > 0,\ \exists \eta_\delta > 0 \mid \forall h \in \mathbf{R}^K,\ \|h\| < \eta_\delta \implies \big(\forall t \in [0,1],\ \|d_\theta \mathcal{F}(t) \cdot h + \epsilon_1(h)(t)\|_{2k} < \delta\big). \tag{2.6}$$

For $\epsilon > 0$, consider the δ_ϵ given in (2.5) and the $\eta_\epsilon := \eta_\delta$ given in (2.6) with $\delta = \delta_\epsilon$. If $\|h\|_K < \eta_\epsilon$, then by (2.6),

$$\|d_\theta \mathcal{F}(t) \cdot h + \epsilon_1(h)(t)\|_{2k} < \delta_\epsilon$$

so by (2.5),

$$\|\epsilon_{2+3}\big(d_\theta \mathcal{F}(t) \cdot h + \epsilon_1(h)(t)\big)\|_{2k} < \epsilon \|d_\theta \mathcal{F}(t) \cdot h + \epsilon_1(h)(t)\|_{2k}$$

for all $t \in [0,1]$. If we take the L^2 norm of the previous inequality, we get

$$\int_0^1 \|\epsilon_{2+3}\big(d_\theta \mathcal{F}(t) \cdot h + \epsilon_1(h)(t)\big)\|^2_{2k} dt < \epsilon^2 \int_0^1 \|d_\theta \mathcal{F}(t) \cdot h + \epsilon_1(h)(t)\|^2_{2k} dt.$$

Since $d_\theta \mathcal{F} \cdot h + \epsilon_1(h)$ is in $\mathcal{O}(\|h\|_K)$, we have

$$\|\epsilon_{2+3}\big(d_\theta \mathcal{F}(t) \cdot h + \epsilon_1(h)(t)\big)\|_{L^2} < \ cste\ \epsilon \|h\|_K.$$

As ϵ is arbitrarily chosen, we have proven that the second non-linear term is also in $o(\|h\|_K)$.

For the remaining non-linear term, we have

$$\begin{aligned}
\|d_{\mathcal{F}(\theta)(\cdot)} X_{\hat{H}_h}\Big(d_\theta \mathcal{F}(h)(\cdot) + \epsilon_1(h)(\cdot)\Big)\|^2_{L^2} &= \int_0^1 \|d_{\mathcal{F}(\theta)(t)} X_{\hat{H}_h}\Big(d_\theta \mathcal{F}(h)(t) + \epsilon_1(h)(t)\Big)\|^2_{2k} dt \\
&\leqslant \int_0^1 |||d_{\mathcal{F}(\theta)(t)} X_{\hat{H}_h}|||^2\ \|d_\theta \mathcal{F}(h)(t) + \epsilon_1(h)(t)\|^2_{2k} dt \\
&\leqslant \int_0^1 cste\ |||\sum_{i=1}^K h_i\ d_{\mathcal{F}(\theta)(t)} X_{f_i}|||^2\ \|h\|^2_K dt \\
&\leqslant \|h\|^4_K \int_0^1 cste\ \max_{1 \leqslant i \leqslant K} |||d_{\mathcal{F}(\theta)(t)} X_{f_i}|||^2 dt
\end{aligned}$$

so this last term is also in $o(\|h\|_K)$. □

To finish the proof of the proposition, it remains to see that the first term in (2.4) is linear and continuous with respect to h while the second term is in $o(\|h\|)$ in L^2. Identifying the linear parts in both sides of (2.4), we finally find that $d_\theta \mathcal{F}(\theta) \cdot h$ is solution of the affine ODE

$$\dot{z}(t) = d_{\mathcal{F}(\theta)(t)} X_{\hat{H}_\theta}(z(t)) + X_{\hat{H}_h}(\mathcal{F}(\theta)(t)) \quad \forall t \in [0,1].$$

Moreover, as $\mathcal{F}(\theta)(0) = Ex_0$ for all θ, $d_\theta \mathcal{F}(\theta) \cdot h$ also satisfies

$$z(0) = 0_{\mathbf{R}^{2k}}.$$

□

2.3 Gradient of $\mathcal{L}$

We now express the differential of $\mathcal{L}$ in terms of $d_\theta\mathcal{F}$. Fix $g \in G$. Let θ be any point of $\mathbf{R}^K$ and h be an infinitesimal displacement in this same space. Consider $\mathcal{F}$ first order Taylor's expansion

$$\mathcal{F}(\theta + h) = \mathcal{F}(\theta) + d_\theta\mathcal{F}(h) + \epsilon_1(h),$$

where $\epsilon_1 : \mathbf{R}^K \to H^1([0,1])^{2k}$ satisfies $\|\epsilon_1(h)\|_{H^1} = o(\|h\|_K)$.

The developement of $\mathcal{L}$ gives

$$\begin{aligned}\mathcal{L}(\theta + h) &= \|D\mathcal{F}(\theta + h) - x\|_{L^2}^2 \\ &= \|D\mathcal{F}(\theta) - x\|_{L^2} + 2\langle Dd_\theta\mathcal{F}(h), D\mathcal{F}(\theta) - x\rangle_{L^2} \\ &\quad + 2\langle D\epsilon_1(h), D\mathcal{F}(\theta) - x\rangle_{L^2} + \|Dd_\theta\mathcal{F}(h) + D\epsilon_1(h)\|_{L^2}^2.\end{aligned}$$

Since D is orthogonal, we have for each pair x, y in $L^2([0,1])^{2k}$ that

$$\langle Dx, Dy\rangle_{L^2} = \int_0^1 Dx \cdot Dy = \int_0^1 x \cdot {}^tDDy = \int_0^1 x \cdot y = \langle x, y\rangle_{L^2}.$$

Thus,

$$\mathcal{L}(\theta + h) = \mathcal{L}(\theta) + 2\langle d_\theta\mathcal{F}(h), \mathcal{F}(\theta) - {}^tDx\rangle_{L^2} + 2\langle \epsilon_1(h), \mathcal{F}(\theta) - {}^tDx\rangle_{L^2} + \|d_\theta\mathcal{F}(h) + \epsilon_1(h)\|_{L^2}^2.$$

The second term of the previous sum is linear and continuous as composition of such applications. By Cauchy's inequality in $L^2([0,1])^{2k}$, the third term is bounded by

$$\|\epsilon_1(h)\|_{L^2}\|\mathcal{F}(\theta) - {}^tDx\|_{L^2}.$$

Since $\|\epsilon_1(h)\|_{H^1}$ and *a fortiori* $\|\epsilon_1(h)\|_{L^2}$ are supposed to be in $o(\|h\|_K)$, so is the third term. Applying the triangular inequality to the last term and invoking the asymptotic behaviours of the linear continuous $d_\theta\mathcal{F}$ and the 1-order negligible ϵ_1, we prove that the last term is also in $o(\|h\|_K)$. Therefore,

$$d_\theta\mathcal{L} = 2\langle d_\theta\mathcal{F}(h), \mathcal{F}(\theta) - {}^tDx\rangle_{L^2} \tag{2.7}$$

Remark 2.6. Unless what usually occurs in optimal control problem, where the parameter θ depends on time, it is here a fixed point of $\mathbf{R}^K$. This allows to rewrite quite immediately the differential of $d_\theta\mathcal{L}$ in the finite dimensional space:

$$\begin{aligned}\langle d_\theta\mathcal{F}, \mathcal{F} - Dx\rangle_{L^2} &= \int_0^1 \sum_{i=1}^{K} \left(d_\theta\mathcal{F}^i(t) \cdot h\right) \times \left(\mathcal{F}^i(\theta)(t) - D^i x(t)\right) dt \\ &= \int_0^1 \sum_{i=1}^{2k} \left(\sum_{j=1}^{K} \partial_j\mathcal{F}^i(\theta)(t) \times h_j\right) \times \left(\mathcal{F}^i(\theta)(t) - D^i x(t)\right) dt \\ &= \sum_{j=1}^{K} h_j \times \left(\int_0^1 \sum_{i=1}^{2k} \partial_j\mathcal{F}^i(\theta)(t) \times \left(\mathcal{F}^i(\theta)(t) - D^i x(t)\right) dt\right) \\ &= h \cdot \int_0^1 \left(\mathcal{F}^i(\theta)(t) - D^i x(t)\right) \cdot \nabla\mathcal{F}(\theta)(t) dt.\end{aligned}$$

Then, we also have $\nabla\mathcal{L}(\theta) = \int_0^1 \left(\mathcal{F}^i(\theta)(t) - D^i x(t)\right) \cdot \nabla\mathcal{F}(\theta)(t) dt$. Now that we have characterised $\nabla\mathcal{F}(\theta)$, we can compute $\nabla\mathcal{L}(\theta)$. However, the computation of $\nabla\mathcal{F}(\theta)$ involves K differential systems in $\mathbf{R}^{2k}$, with K possibly very large.

In fact, it is possible to write $\nabla\mathcal{L}(\theta)$ in another way which only involves one differential system in $\mathbf{R}^{2k}$. To show that, we use a classical method in optimal control, which makes use of a well-chosen adjoint function $a : [0,1] \to \mathbf{R}^{2k}$. More precisely, we set a as the unique solution of the adjoint Cauchy's problem

$$\begin{cases} \dot{a}(t) = \mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))\mathbf{J}a(t) + \Big(\mathcal{F}(\theta)(t) - {}^tDx(t)\Big), \forall t \in [0,1], \\ a(T) = 0, \end{cases} \tag{2.8}$$

where $\mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))$ represents the Hessian matrix of H_θ in $\mathcal{F}(\theta)(t)$.

Theorem 2.7. *For all θ in $\mathbf{R}^{2k}$,*

$$\nabla\mathcal{L}(\theta) = -2\int_0^1 \mathbf{X}\Big(\mathcal{F}(\theta)(t)\Big)a(t)dt, \tag{2.9}$$

with a and $\mathbf{X}$ as previously defined.

Proof. We have

$$\dot{a}(t) - \mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))a(t) = \Big(\mathcal{F}(\theta)(t) - {}^tDx(t)\Big)$$

for all t in $[0,1]$. Inserting this equality in (2.7), we get

$$d_\theta\mathcal{L}(h) = 2\int_0^1 \langle \dot{a}(t) - \mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))a(t), d_\theta\mathcal{F}(h)(t)\rangle_{\mathbf{R}^{2k}}dt.$$

After an integration by part of the first term, we obtain

$$\begin{aligned} d_\theta\mathcal{L}(h) = 2\int_0^1 &-\langle a(t), d_\theta\dot{\mathcal{F}}(h)(t)\rangle_{\mathbf{R}^{2k}} - \langle \mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))a(t), d_\theta\mathcal{F}(h)(t)\rangle_{\mathbf{R}^{2k}}dt \\ &+ a(T)d_\theta\mathcal{F}(h)(T) - a(0)d_\theta\mathcal{F}(h)(0). \end{aligned}$$

Thanks to the initial and final condition on a and $d_\theta\mathcal{F}(h)$, the last two terms vanish. Using the equation satisfied by $d_\theta\mathcal{F}(h)$ (eq. (2.3)), we have

$$\begin{aligned} d_\theta\mathcal{L}(h) = 2\int_0^1 &-\langle a(t), d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta} \cdot d_\theta\mathcal{F}(h)(t)\rangle_{\mathbf{R}^{2k}} - \langle a(t), X_{\hat{H}_h}(\mathcal{F}(\theta)(t))\rangle_{\mathbf{R}^{2k}} \\ &- \langle \mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))a(t), d_\theta\mathcal{F}(h)(t)\rangle_{\mathbf{R}^{2k}}dt. \end{aligned}$$

Now, it suffices to notice that

$$d_{\mathcal{F}(\theta)(t)}X_{\hat{H}_\theta} = d_{\mathcal{F}(\theta)(t)}\mathbf{J}\nabla\hat{H}_\theta = \mathbf{J}d_{\mathcal{F}(\theta)(t)}\nabla\hat{H}_\theta = \mathbf{J}\mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))$$

and that

$${}^t(\mathbf{J}\mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))) = {}^t\mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t)){}^t\mathbf{J} = -\mathcal{H}_{H_\theta}(\mathcal{F}(\theta)(t))\mathbf{J}$$

to get that the first and the third terms cancel out. We finally get

$$\begin{aligned} d_\theta\mathcal{L}(h) &= -2\int_0^1 \langle a(t), X_{\hat{H}_h}\big(\mathcal{F}(\theta)(t)\big)\rangle_{2k}dt \\ &= -2\int_0^1 \langle a(t), {}^t\mathbf{X}\Big(\mathcal{F}(\theta)(t)\Big)h\rangle_{2k}dt \\ &= -2\int_0^1 \langle \mathbf{X}\Big(\mathcal{F}(\theta)(t)\Big)a(t), h\rangle_K dt \\ &= \langle -2\int_0^1 \mathbf{X}\Big(\mathcal{F}(\theta)(t)\Big)a(t)dt, h\rangle_K. \end{aligned}$$

We conclude that

$$\nabla_\theta \mathcal{L} = -2 \int_0^1 \mathbf{X}\Big(\mathcal{F}(\theta)(t)\Big) a(t) dt.$$

□

To compute this gradient, we have to solve an ODE in dimension $2k$, evaluate $2k \times K$ functions throughout this trajectory and perform a matrix multiplication in dimension $(K, 2k) \times (2k, m)$, where m is the number of time steps. This is far less expensive than solving $2k \times K$ ODE.

Remark 2.8. The equation satisfied by the adjoint can be recovered from a formal resolution of the optimisation problem

$$\inf_{x_\theta} J(x)$$

under the constraint that $x_\theta \in H^1([0,1])^{2k}$ is solution of the Hamiltonian problem associated to $\hat{H}_\theta$. with $J(x_\theta) = \mathcal{L}(\theta)$. The Lagrangian operator of this problem is given by

$$L(\theta, \lambda) = J(x_\theta) + \int_0^1 \lambda \cdot \big(x'_\theta - X_{\hat{H}_\theta}(x_\theta)\big) dt.$$

The point $(\theta, \lambda) \in \mathbf{R}^K \times H^1([0,1])^{2k}$ is a critical point if

$$\begin{cases} \partial_\theta L(\theta, \lambda) = 0, \\ \partial_\lambda L(\theta, \lambda) = x'_\theta - X_{\hat{H}_\theta}(x_\theta) = 0. \end{cases}$$

As all the functions at stake are at least L^2, we can switch the integral and the derivative in the first equation, which gives for all $h \in \mathbf{R}^K$ small enough

$$\partial_x J(x_\theta)\big(d_\theta \mathcal{F}(h)\big) + \int_0^1 \lambda \cdot \left(\frac{d}{dt} d_\theta \mathcal{F}(h) - d_{x_\theta} X_{\hat{H}_\theta}\big(d_\theta \mathcal{F}(h)\big)\right) dt = 0.$$

After an integration by parts, the first term in the integral becomes

$$-\int_0^1 \lambda' \cdot d_\theta \mathcal{F}(h) dt + \lambda(1) \cdot d_\theta \mathcal{F}(h)(1) - \lambda(0) \cdot d_\theta \mathcal{F}(h)(0),$$

where the last term vanishes thanks to the initial condition on $d_\theta \mathcal{F}$. We get

$$\int_0^1 \Big[(x_\theta - {}^t Dx) - \lambda' - \lambda \cdot d_{x_\theta} X_{\hat{H}_\theta}\Big] \cdot d_\theta \mathcal{F}(h) dt + \lambda(1) \cdot d_\theta \mathcal{F}(h)(1) = 0.$$

The previous equality is then verified if

$$\begin{cases} \lambda'(t) = d_{x_\theta} X^*_{\hat{H}_\theta}(\lambda(t)) + (x_\theta(t) - {}^t Dx(t)), & \forall t \in [0,1] \\ \lambda(1) = 0, \end{cases}$$

which is exactly the adjoint problem.

2.4 Penalisation

In practice, the number K can be chosen very large. When this is the case, it slows down the evaluation of $X_{\hat{H}_\theta}$ and then reduces the efficiency of the reduction. To remedy this problem, we can add a penalisation term at the loss function and minimise

$$\mathcal{L}(\theta) = \int_{g\in G}\int_{t\in[0,1]} \|D\hat{x}_{\theta,g}(t) - x_g(t)\|^2_{\mathbf{R}^{2k}} dgdt + \alpha\|\theta\|_1,$$

where α is a positive real number. The additional term forces a lot of coefficients of α to be exactly set to zero.

As it is given, this penalisation term is not differentiable. This is why we replace it in practice by $\theta \mapsto \sqrt{\theta^2+\epsilon}$ for an epsilon chosen very small, 10^{-6} for example. Then, we just have to add $\frac{\theta}{\sqrt{\theta^2+\epsilon}}$ at the gradient we have found in the previous section.

2.5 Algorithms

Now, that we have a satisfying expression for $\nabla\mathcal{L}(\theta)$, we can perform a numerical optimisation using Algorithm 1.

Recall that we suppose that we have computed a set of solution $\{x_{g_i}\}_{i=1,...,m}$ for some values $\{g_i\}_i \subset G$ of the parameter g. It can be a parameter of the primal equation or the initial condition. When g parametrises the primal equation, a way to take this dependence into account in the reduction is to take functions f_i which involve this parameter. By doing this, the optimisation will give a family $\{H_{\theta*,g}\}_g$, where θ^* is the optimal value of the parameter θ over all the trajectories x_{g_i}. When g only appears in the initial condition, we are only looking for one Hamiltonian function $H_{\theta*}$ so we have to take the f_i independent of g.

Algorithm 1 Simple gradient descent

Require: $\theta_0 \in \mathbf{R}^K$, α, $\eta > 0$, $\rho > 0$, $G = \{g_1, ..., g_m\}$, $x_G := \{x_{g_1}, ..., x_{g_m}\}$ and $\mathbf{X}$

$\theta \leftarrow \theta_0$
while $\frac{\|\nabla\mathcal{L}(\theta)\|}{\|\nabla\mathcal{L}(\theta_0)\|} > \eta$ **do**
  $\nabla \leftarrow \frac{\alpha\theta}{\sqrt{\theta^2+\epsilon}}$
  for all $g \in G$ **do**
    compute the solution x_θ of (2.1) for g and current θ
    compute a_θ of (2.8) for current θ, g, x_θ and x_g.
    compute $\nabla\mathcal{L}_g(\theta)$ from (2.9) with a_θ and x_θ
    $\nabla \leftarrow \nabla + \nabla\mathcal{L}_g(\theta)$
  end for
  $\theta \leftarrow \theta - \rho\nabla$
end while

Of course, this basic algorithm can be sophisticated by adding momentum or by using Adam descent instead of the basic gradient descent. We will present comparisons for some cases in the following section.

During tests, it happens that taking into account only a small portion of the studied interval $[0, 1]$ at each step can highly improve the descent efficiency. Below are exposed two precise algorithms which make use of this idea.

Algorithm 2 Progressive gradient descent

Require: $\theta_0 \in \mathbf{R}^K$, α, $\eta > 0$, $\rho > 0$, $G = \{g_1, ..., g_N\}$, $x_G := \{x_{g_1}, ..., x_{g_N}\}$, $w \in [\![1, m]\!]$, Δt, $\mathbf{X}$

$\theta \leftarrow \theta_0$

while $\frac{\|\nabla \mathcal{L}(\theta)\|}{\|\nabla \mathcal{L}(\theta_0)\|} > \eta$ **do**

$\nabla \leftarrow \frac{\alpha\theta}{\sqrt{\theta^2+\epsilon}}$

for $i = 0, ..., \lfloor \frac{m}{w} \rfloor$ **do**

$b \leftarrow iw$

$c \leftarrow b + w$

for all $g \in G$ **do**

compute the solution $x_{\theta,g}$ of (2.1) starting at $x_0 = x_g(b\Delta t)$ on the interval $[b\Delta t, c\Delta t]$.

compute $a_{\theta,g}$ of (2.8) from $x_{\theta,g}$ ending at $a(c\Delta t) = 0$ on the interval $[b\Delta t, c\Delta t]$.

compute $\nabla \mathcal{L}_g(\theta)$ from (2.9) with a_θ and x_θ on the interval $[b\Delta t, c\Delta t]$.

$\nabla \leftarrow \nabla + \nabla \mathcal{L}_g(\theta)$

end for

$\theta \leftarrow \theta - \rho\nabla$

end for

end while

In Algorithm 2, we in fact use a different gradient at each step, which is actually the gradient of $\mathcal{L}_g$ when the integration interval is $[b\Delta t, c\Delta t]$ instead of $[0, 1]$. This way, we introduce stochastic-like effect in the descent. A variation of this method can be to shift the considered window of one time step Δt instead of w time steps. Another one would consist in taking randomly the starting points b at each step. One can also imagine taking at each step several intervals instead of one. We will discuss all these variations in the following section.

2.6 Tests

To test our method, we first start with a case where we know the optimal value of the parameter. We consider a family F of functions from $\mathbf{R}^2$ to $\mathbf{R}$ and take $H = H_{\theta*}$ in the space spanned by F. We then look if one of the three algorithms presented in the previous section succeeds in finding θ^* as the optimal parameter. We suppose that $D = id_2$.

2.6.1 Hand-made convex problem in dimension 2

Problem setting

We start by a very simple case, where all the quantities at stake can be exactly computed in theory. Let $F_1 = \{(p, q) \mapsto p; (p, q) \mapsto q\}$ and $\theta^* = (\theta_1^*, \theta_2^*)$. The targeted Hamiltonian is then $H_{\theta*} : (p, q) \mapsto \theta_1^* p + \theta_2^* q$. The system associated to the Hamiltonian H_θ is given by

$$\begin{cases} \dot{p}_\theta(t) = -\theta_2^*, \\ \dot{p}_\theta(t) = \theta_1^* \end{cases}$$

for all $t \in [0,1]$ and its solution by $x_\theta = (p_0 - \theta_2 t, q_0 + \theta_1 t)$. Therefore, the loss for one trajectory can be rewritten explicitly, which is

$$\mathcal{L}(\theta) = \int_0^1 \left[(\theta_1 - \theta_1^*)^2 + (\theta_2 - \theta_2^*)^2\right] t^2 dt = \frac{(\theta_1 - \theta_1^*)^2 + (\theta_2 - \theta_2^*)^2}{3}.$$

Its gradient is

$$\nabla\mathcal{L}(\theta) = \frac{2}{3}(\theta_1 - \theta_1^*\theta_2 - \theta_2^*).$$

Finally, since the Jacobian matrix of H_θ is always zero, the adjoint state a_θ satisfies

$$\dot{a}_\theta(t) = x_\theta(t) - x_{\theta^*}(t)$$

for all $t \in [0,1]$ and is therefore given by $a_\theta = (\frac{\theta_2^* - \theta_2}{2}, \frac{\theta_1 - \theta_1^*}{2})(t^2 - 1)$.

Intermediate quantities

Figure 2.1 shows the errors we get for different time steps in the resolution of the primal and dual problems. The given values are the mean of the errors we obtained on a sample of 100 random values of θ. For $\theta = \theta^*$, the loss and its gradient are always equal to 0.

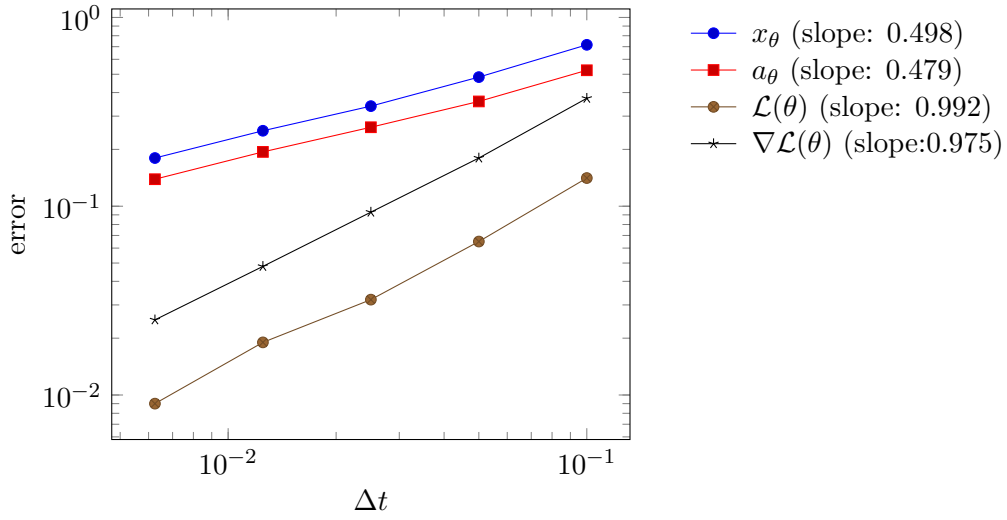


Figure 2.1: Mean error made on the primal and dual states, the loss, and the gradient of the loss at 100 points uniformly sampled in $[-5,5]^2$ for different values of the time step in the numerical resolution of the primal and dual states ($\Delta t = 0.1, 0.05, 0.025, 0.0125$ and 0.00625). We always used the same time step to compute the primal and the dual states on $[0,1]$. We have taken $(p_0, q_0) = (1,0)$, $F = F_1$ and $\theta^* = (\frac{1}{2}, \frac{1}{4})$. The error on the primal and dual states are given in L^2 norm.

Optimisation

For one trajectory, the optimisation problem we want to solve here is strictly convex so it admits an unique minimum which can be easily found using a gradient descent. On Figure 2.2 we show the iterates we obtain with Algorithm 1 on the graph of $\mathcal{L}$. As expected, we see that the minimum is easily reached.

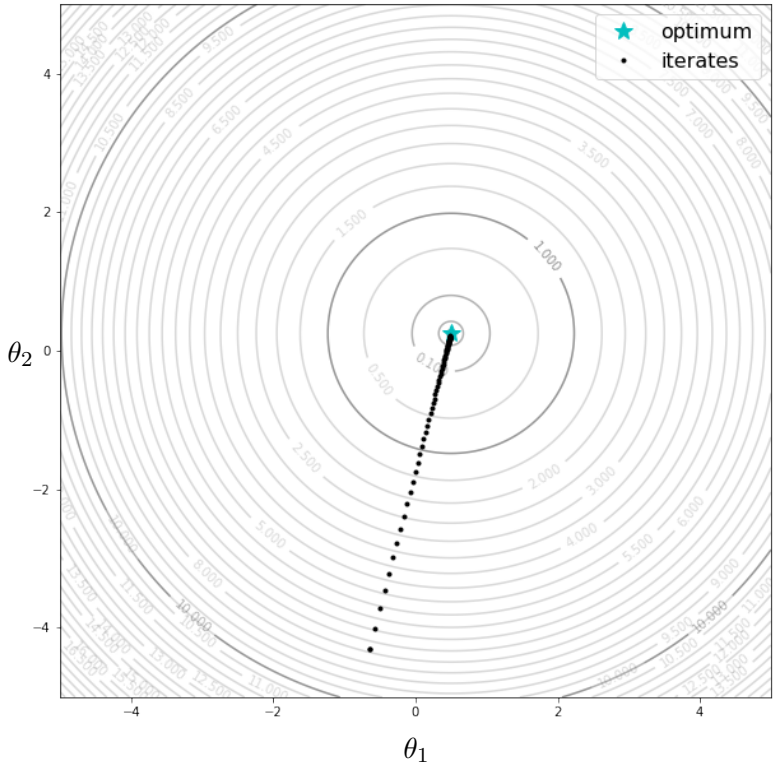


Figure 2.2: Iterates obtained during optimisation process using Algorithm 1 with $\rho = 0.1$, $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 1]$. The targeted Hamiltonian was $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p + \frac{1}{4}q$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$ and taken $F = F_1$ and $\theta^* = (\frac{1}{2}, \frac{1}{4})$. We started the optimisation from a point uniformly sampled in $[-5, 5]^2$ $((-0.646, -4.304))$.

2.6.2 Hand-made non-convex problem in dimension 2

Optimisation with Algorithm 1

We now take $F = \{(p, q) \mapsto p^2, (p, q) \mapsto q^2\}$. In this case, we can compute the exact solutions of the primal problem, but it is more difficult to give an exact expression of the loss and the adjoint state. We computed the loss for numerous values in the square $[0, 1]^2$ and obtain the graph presented Figure 2.3. As we can see, we do not have a convex problem any more and if we do not initialise the descent close to the optimal value, a classical gradient algorithm as Algrithm 1 will hardly find it. As we can see on Figure 2.4, Algorithm 1 stays indeed trapped in local minima.

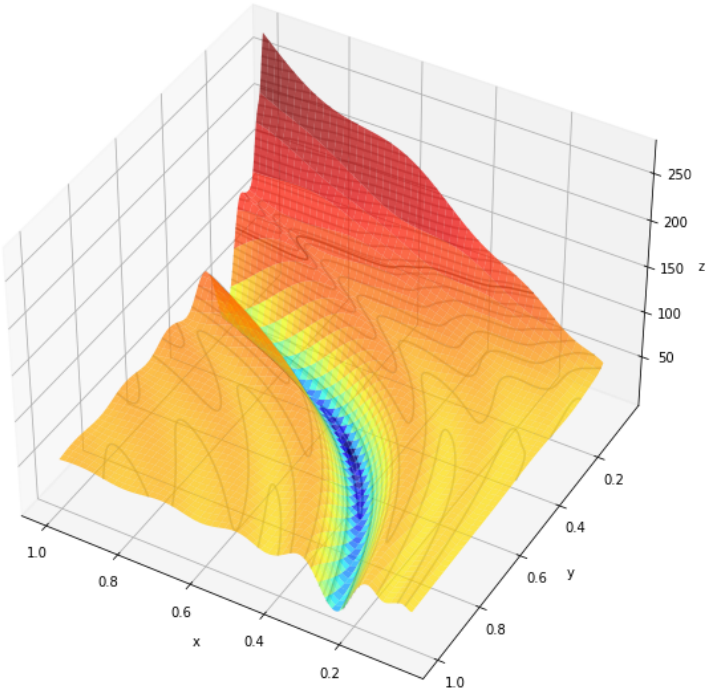


Figure 2.3: Graph of $\mathcal{L}$ computed on one trajectory starting at $(p_0, q_0) = (1, 0)$. We have taken $F = F_2$ and $\theta^* = (\frac{1}{2}, \frac{1}{2})$. The targeted Hamiltonian was $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$.

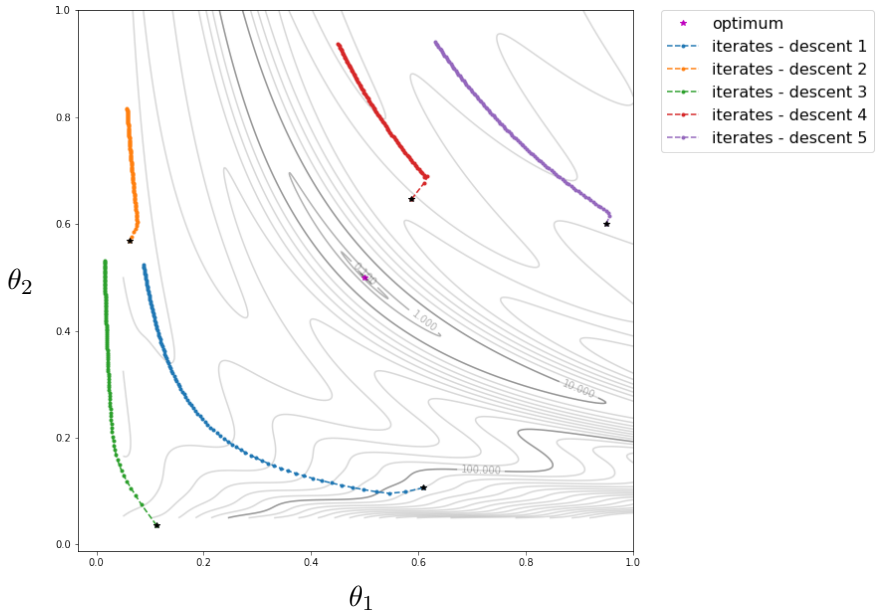


Figure 2.4: Iterates obtained during optimisation process using Algorithm 1 with $\rho = 10^{-4}$ and $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 1]$. The targeted Hamiltonian was $h : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$ and taken $F = F_2$. We started the optimisation from 6 points randomly sampled in $[0, 1]^2$ and used 100 steps.

Optimisation with Algorithm 2

We now use Algorithm 2. We first take the version in which the first point of the window we consider at step $n+1$ of the descent is the end of the window at step n. If the window at step n is $[a^n, b^n]$, it becomes $[b^n, b^n + w]$ at step $n+1$, with w the window size. We test the algorithm for different values of w. As we see on Figure 2.5, we have to change the learning rate ρ depending on the value we have chosen for w: the smaller the width is, the larger the learning rate should be. In fact, the descent direction slightly changes from one step to another, which gives smooth and oscillating trajectories in the parameter space. When the learning rate takes a high value, the iterates seems to jump from an arbitrary point to another in the parameter space. As we can see on Figure 2.6, which shows the decreasing of the loss during the optimisation, the process converges to the optimum all the same. On the contrary, when the learning rate takes small values, the trajectories are more straight and tend slower to the optimum. This link between learning rate and window size is for a great part due to the fact that the sum in the discrete gradient is done on few points when the window size is small, so the gradient is smaller than when we choose a larger window.

In almost all the cases we considered, the algorithm always seems to converge. In some cases, it even converges after few iterations and before reaching the end of the considered time interval. When we look at the trajectories in the parameter space, we see that the iterates jump above the bumps in the loss graph as if we were actually solving a convex optimisation problem. This behaviour is very different from what we observed with the simple descent algorithm.

In fact, when we reduce the window size, we only consider the error made on a subinterval in time. At each step, the problem we consider is slightly different, but it always has the same solution. The optimisation process we present here is therefore similar to the one used in neural networks, when we take a subset of the set of training data. When we choose the window at each step in a way that its first point is the last of the window at the previous step, the process is entirely deterministic.

We can add stochasticity to the optimisation by randomly chosen the starting point of the window at each step, but this does not change the convergence. On Figures 2.7 and 2.8, we present the trajectories of the iterates during the optimisation and the corresponding variations of the loss when we do such choices for the windows. We initialised θ from the same points as in Figure 2.5 and we see that the behaviour of the algorithm does not change from the previous case: it still converges and at the same speed for each couple (ρ, w). Moreover, we also see that the smoothness of the trajectories in the parameter space are globally the same, we just notice that for the stochastic algorithm, they are a little less smooth. This is easily explained by the fact that the problems we consider at a given step can be more different from the problem at the previous step if the corresponding windows are not adjacent.

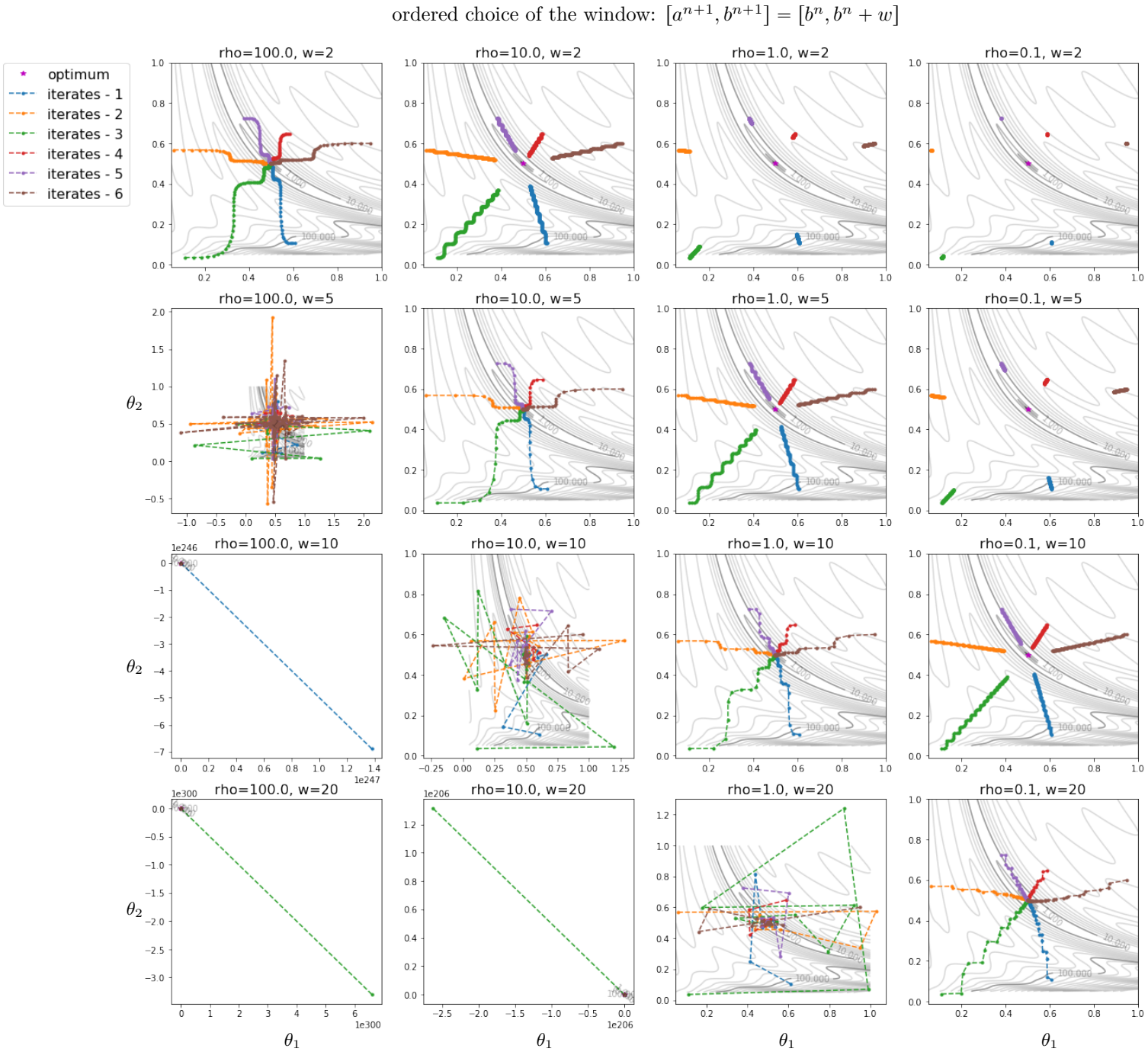


Figure 2.5: Iterates obtained during optimisation process using Algorithm 2 with $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 25]$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$. We have taken $F = F_2$ and $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We started the optimisation from 6 points uniformly sampled in $[0, 1]^2$. The learning rate ρ changes on each column and takes the values 100 (left), 10, 1 and 0.1 (right). The window size changes on each line, taking the values $2\Delta t$ (top), $5\Delta t, 10\Delta t$ and $50\Delta t$ (bottom). The last point of the window at a given step becomes the first point of the window at the following step.

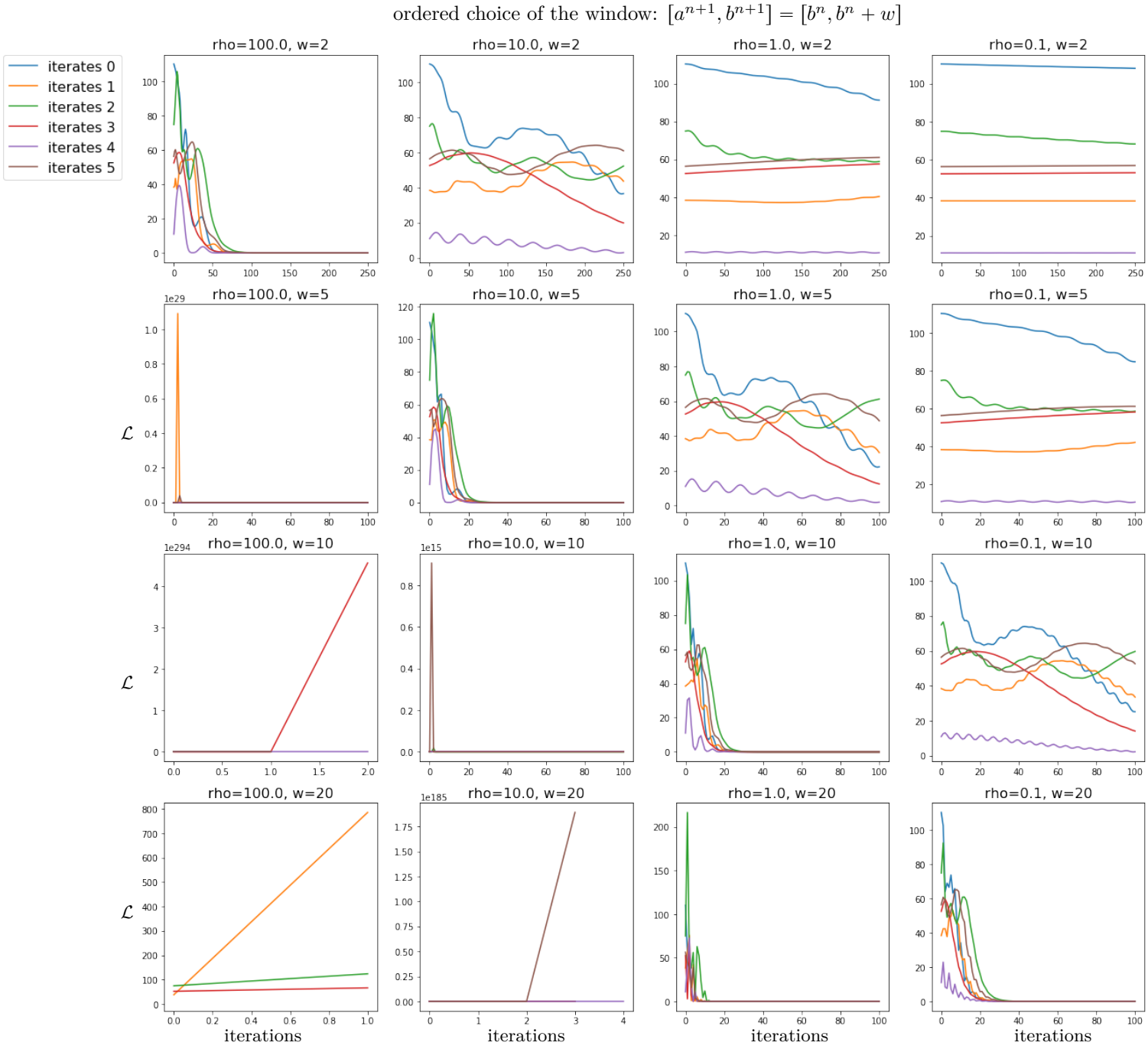


Figure 2.6: Loss on the trajectories taken by the iterates during optimisation process using Algorithm 2 with $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 25]$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$. We have taken $F = F_2$ and $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We started the optimisation from 6 points uniformly sampled in $[0, 1]^2$. The learning rate ρ changes on each column and takes the values 100 (left), 10, 1 and 0.1 (right). The window size changes on each line, taking the values $2\Delta t$ (top), $5\Delta t, 10\Delta t$ and $50\Delta t$ (bottom). The last point of the window at a given step becomes the first point of the window at the following step. The horizontal axis represents the iterations and the vertical axis the logarithm of the loss.

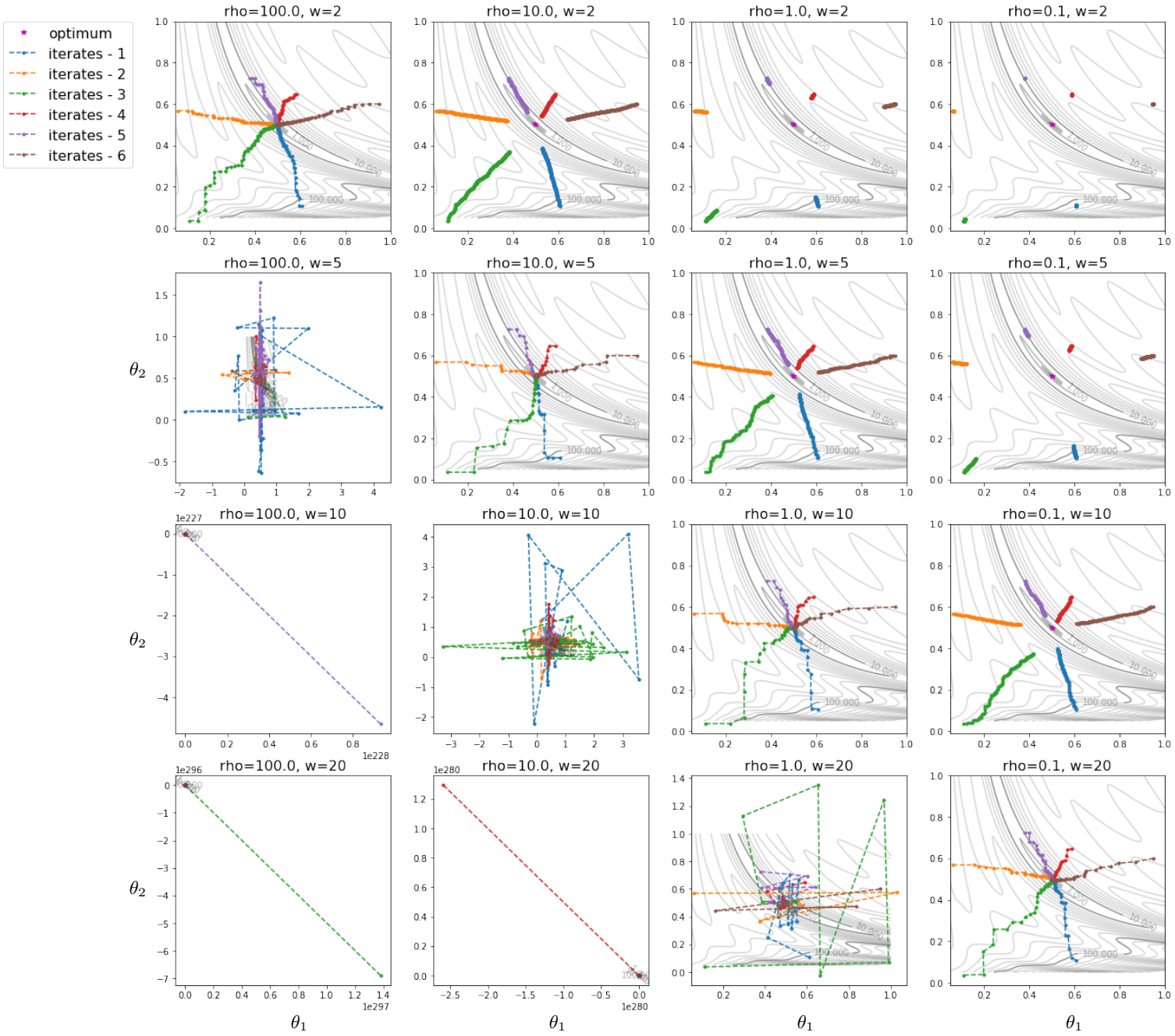


Figure 2.7: Iterates obtained during optimisation process using Algorithm 2 with $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 25]$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$. We have taken $F = F_2$ and $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We started the optimisation from 5 points uniformly sampled in $[0, 1]^2$. The learning rate ρ changes on each column and takes the values 100 (left), $10, 1$ and 0.1 (right). The window size changes on each line, taking the values $2\Delta t$ (top), $5\Delta t, 10\Delta t$ and $50\Delta t$ (bottom). The first point of the window at a given step is randomly chosen.

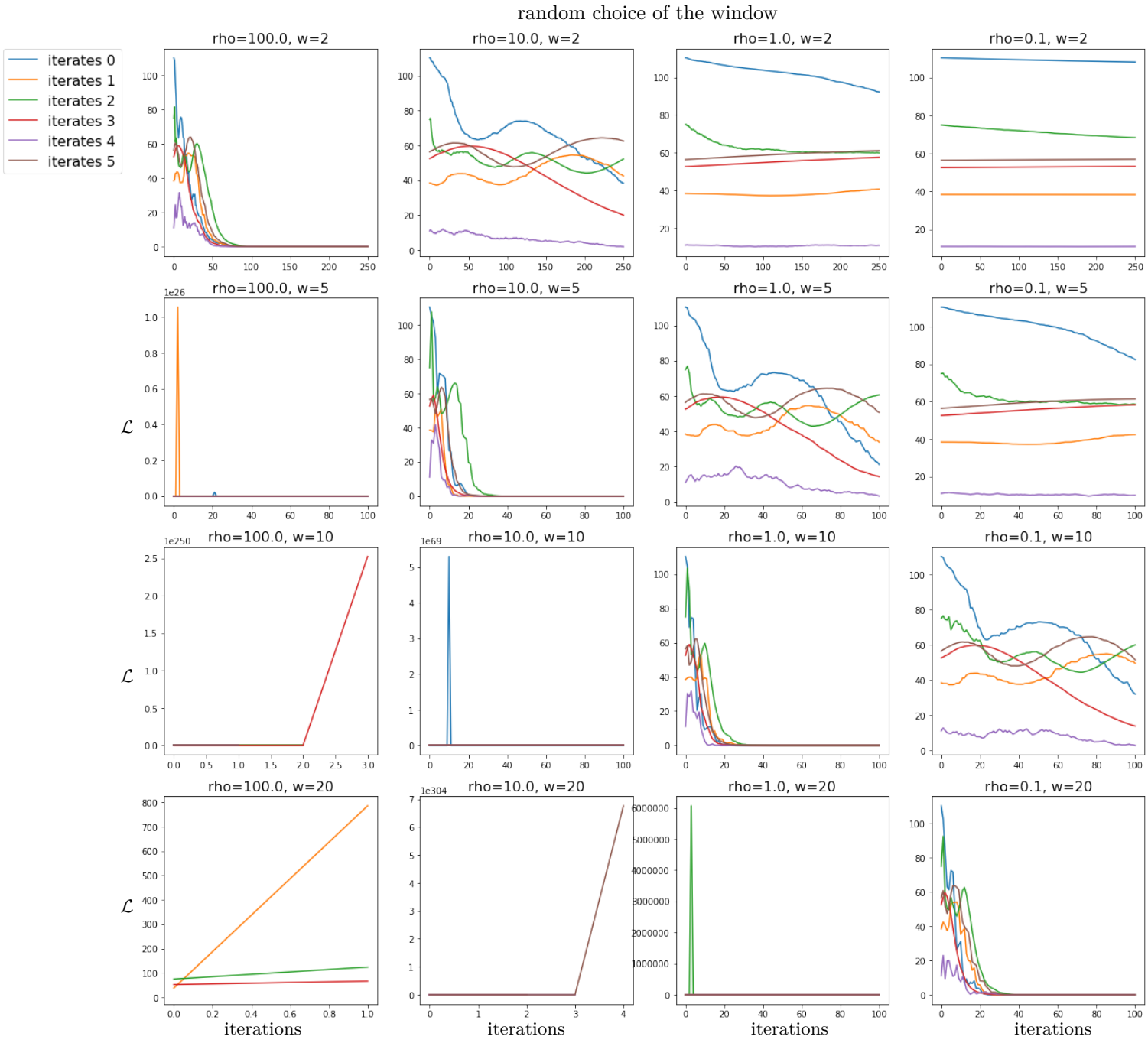


Figure 2.8: Loss on the trajectories taken by the iterates during optimisation process using Algorithm 2 with $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 25]$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$. We have taken $F = F_2$ $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We started the optimisation from 6 points uniformly sampled in $[0, 1]^2$. The learning rate ρ changes on each column and takes the values 100 (left), $10, 1$ and 0.1 (right). The window size changes on each line, taking the values $2\Delta t$ (top), $5\Delta t, 10\Delta t$ and $50\Delta t$ (bottom). The first point of the window at a given step is randomly chosen. The horizontal axis represents the iterations and the vertical axis the logarithm of the loss.

Algorithm 2 starting far from the optimum and over long time

We now test Algorithm 2 on a larger time interval, $[0, 50]$ instead of $[0, 25]$, with an initialisation further from the optimum. We expect that the fact of considering larger time interval won't change the behaviour of the algorithm, since it only uses subintervals. As we see on Figures 2.9 and 2.10, this is indeed not the case. The fact of initializing the optimisation far from the optimum does not prevent the algorithm to converge in most cases. When we take $w = 2\Delta t$ or $5\Delta t$, we find the minimum from the ten starting points we have chosen. When we take $w = 10\Delta t$ or $20\Delta t$, some starting points do not lead to the minimum. In some cases, the algorithm diverges and there is one case (case 1 with $w = 20\Delta t$ in the figures) in which the iterates seem to draw away from the optimum. In all those cases except case 2 with $w = 10\Delta t$, reducing the learning rate does not change these behaviours. We make the same observations for the stochastic version of Algorithm 2.

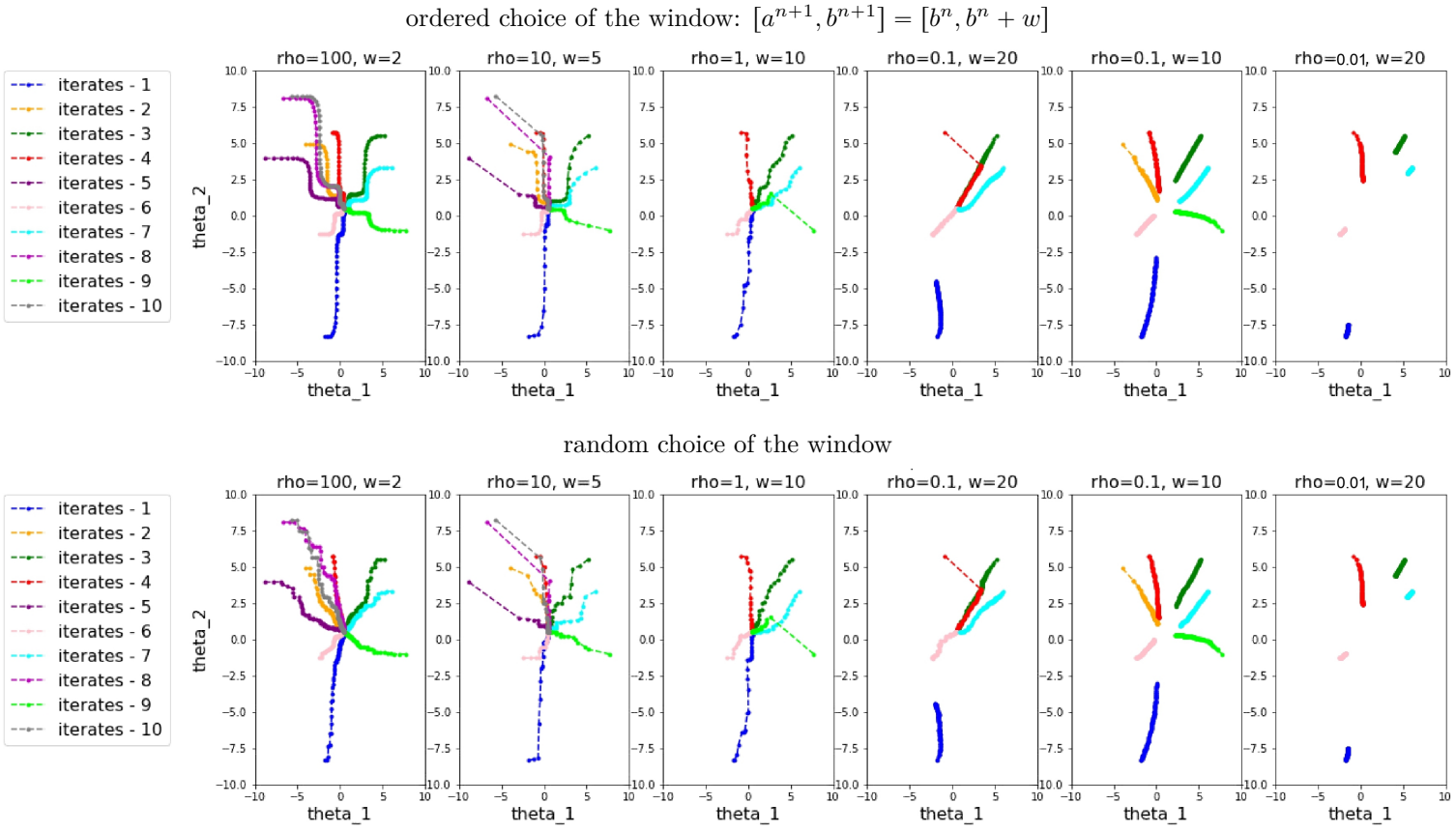


Figure 2.9: Iterates obtained during optimisation process using Algorithm 2 with $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 50]$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$. We have taken $F = F_2$ $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We started the optimisation from 10 points uniformly sampled in $[-10, 10]^2$. The learning rate and the window size change on each column, taking the values $(2\Delta t, 100)$ (left), $(5\Delta t, 10), (10\Delta t, 1), (20\Delta t, 1e-1), (10\Delta t, 1e-1)$ and $(20\Delta t, 1e-2)$ (right). The first point of the window at a given step is the last point of the window at the previous step (first line) or is randomly chosen (second line).

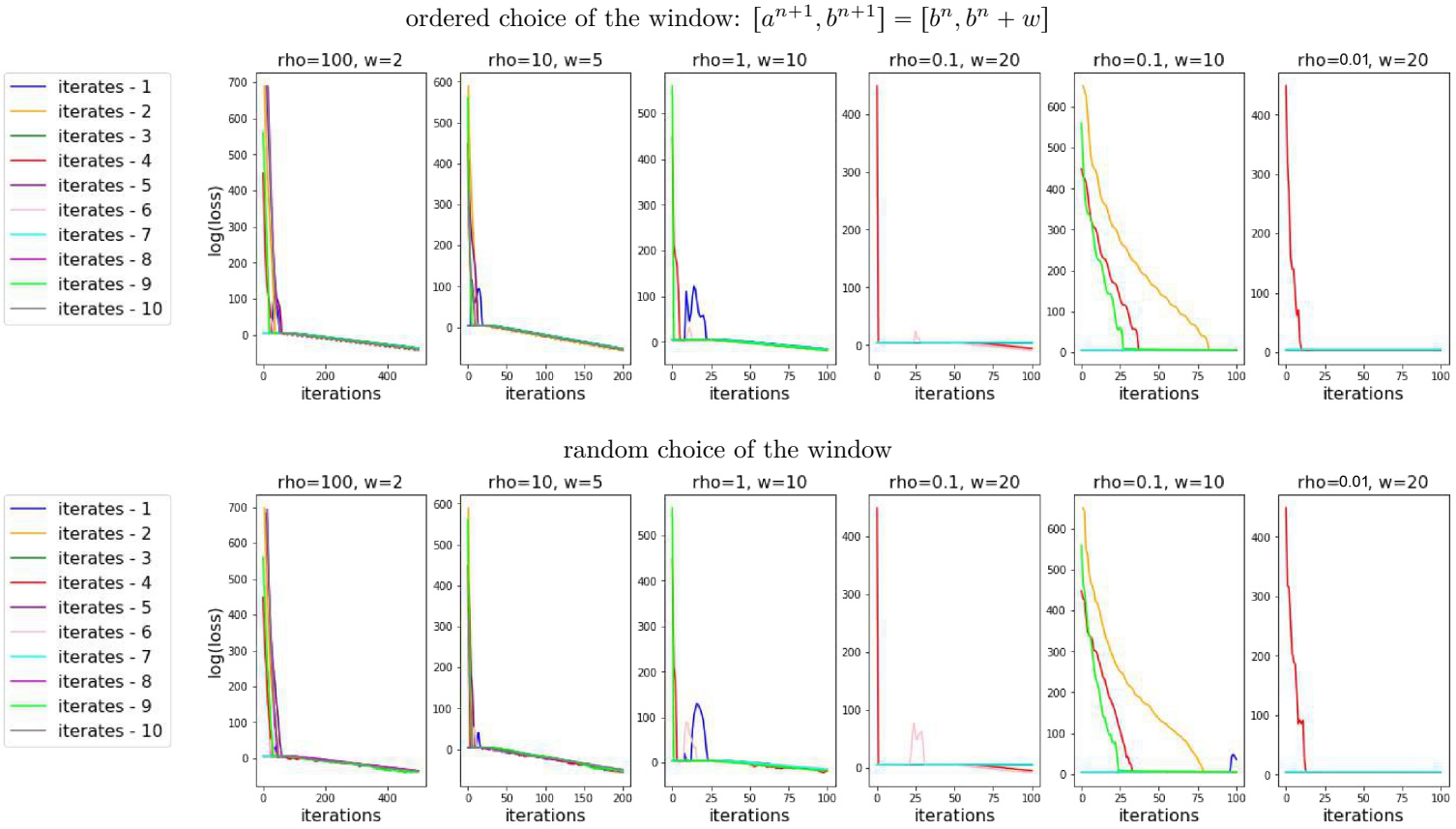


Figure 2.10: Loss on the trajectories taken by the iterates during optimisation process using Algorithm 2 with $\alpha = 0$. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 50]$. We have considered one trajectory starting at $(p_0, q_0) = (1, 0)$. We have taken $F = F_2$ and $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We started the optimisation from 6 points uniformly sampled in $[0, 1]^2$. The learning rate and the window size change on each column, taking the values $(2\Delta t, 100)$ (left), $(5\Delta t, 10), (10\Delta t, 1), (20\Delta t, 1e-1), (10\Delta t, 1e-1)$ and $(20\Delta t, 1e-2)$ (right). The first point of the window at a given step is the last point of the window at the previous step (first line) or is randomly chosen (second line). The horizontal axis represents the iterations and the vertical axis the logarithm of the loss.

Algorithm 2 with several trajectories

We now look what happens if we take G not reduced to a single point. Here, we consider 5 trajectories obtained from the same Hamiltonian with different initial conditions. As the random window choice does not change the behaviour of Algorithm 2 in previous cases, we only did tests for its deterministic version. Results are presented on Figure 2.11. We see that we still converge but a little more quickly. We may explain that by the fact that we did not change the learning rates while the gradient, as a sum over G, is necessarily larger than in the single trajectory case.

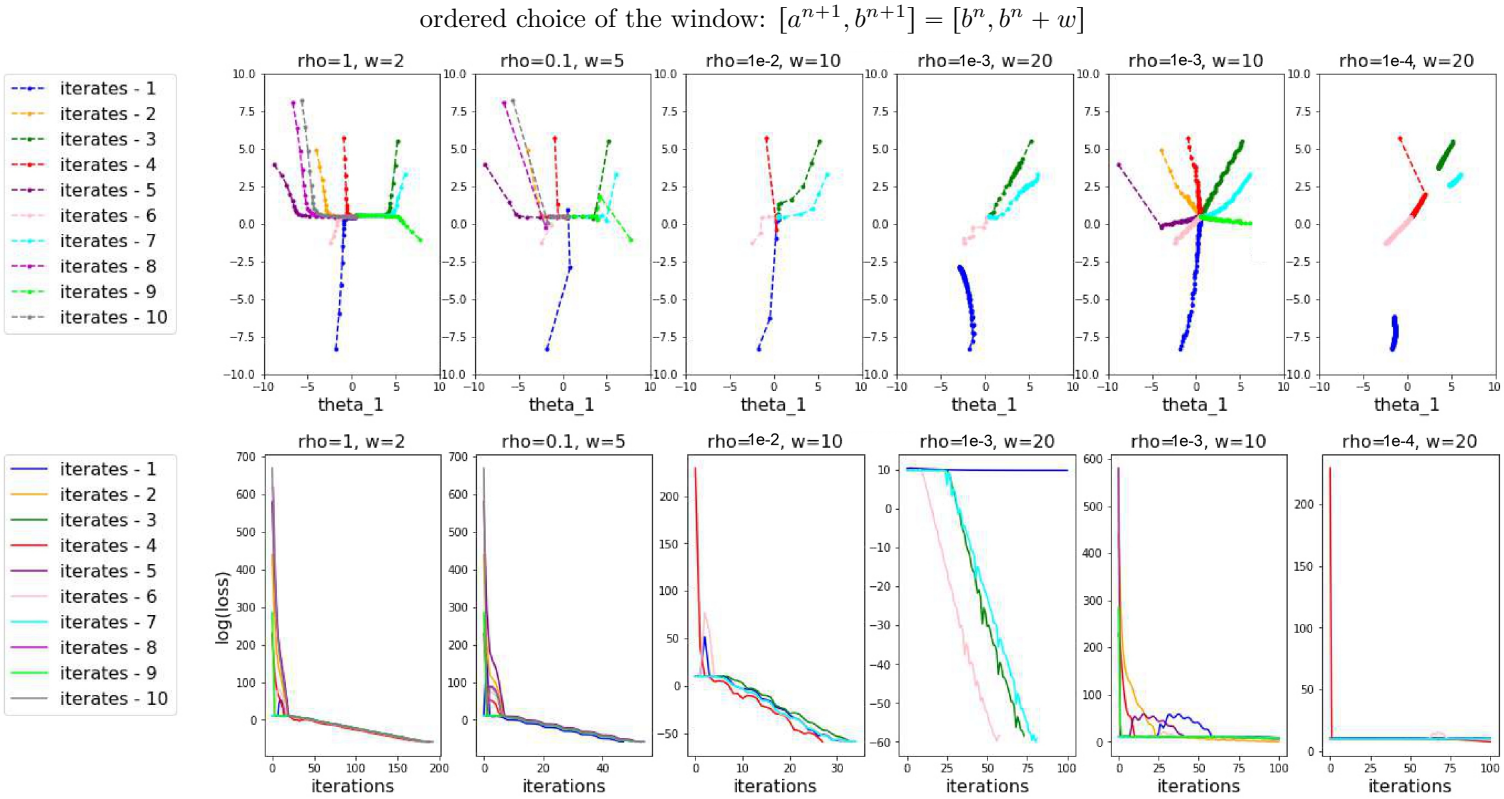


Figure 2.11: First line: iterates obtained during optimisation process using Algorithm 2 with $\alpha = 0$. Second line: corresponding variations of the loss during the optimisation. The horizontal axis represents the iterations and the vertical axis the logarithm of the loss. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 50]$. We have considered 5 trajectories starting at five points uniformly sampled in $[0, 10]$ $((2.61654536, 6.97328654)$, $(8.35401404, 7.58261582)$, $(2.41885671, 6.60159081)$, $(4.64960361, 5.81467692)$ and $(2.95476693, 9.69287747))$. We have taken $F = F_2$ and $H : (p, q) \in \mathbf{R}^2 \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. We started the optimisation from 10 points uniformly sampled in $[-10, 10]^2$. The learning rate and the window size change on each column, taking the values $(2\Delta t, 100)$ (left), $(5\Delta t, 10), (10\Delta t, 1), (20\Delta t, 1e-1), (10\Delta t, 1e-1)$ and $(20\Delta t, 1e-2)$ (right). The first point of the window at a given step is the last point of the window at the previous step.

2.6.3 Hand-made non-convex problem in dimension 16

We now test Algorithm 2 when the family F is large and does not contain only the functions needed to build the targeted Hamiltonian. We choose $F = F_3$ with

$$F_3 = Cos_p \cup Cos_q \cup Sin_p \cup Sin_q \cup \{(p, q) \mapsto 1\},$$

where

$$Cos_p = \left\{ (p, q) \mapsto \cos(\frac{2\pi}{n} p) \right\}_{1 \leqslant n \leqslant 5}, \qquad Cos_q = \left\{ (p, q) \mapsto \cos(\frac{2\pi}{n} q) \right\}_{1 \leqslant n \leqslant 5},$$

$$Sin_p = \left\{ (p, q) \mapsto \sin(\frac{2\pi}{n} p) \right\}_{1 \leqslant n \leqslant 5}, \qquad Sin_q = \left\{ (p, q) \mapsto \sin(\frac{2\pi}{n} q) \right\}_{1 \leqslant n \leqslant 5}.$$

To build the targeted Hamiltonian function, we have randomly chosen a θ^* in $[-10, 10]^K$ with a lot of coefficients exactly equal to 0 and have taken $H = H_{\theta*}$. We have exactly

$$\theta^* = (0, 0, 0, -0.9, 0.1, -2.1, 4.3, 0, 0, 0, 0, 3.0, 0, 0, 0, 0)$$

. As previously, the loss is composed of the sum of the L^2 errors on several trajectories (here $|G| = 5$) starting in 5 randomly sampled initial conditions.

On Figure 2.12, we show the decreasing of the logarithm of the loss during four descents ($\rho = 10$, $w = 2\Delta t$) starting from different θ_0. There are still some oscillations, but $\log(\mathcal{L})$ globally decreases constantly. After 3500 iterations, we obtain a θ which leads to the solutions presented on Figure 2.13 for the initial conditions used to compute the loss (on the figure is presented what we obtain after one of the four descents considered on Figure 2.12 but the three others give similar results). We see that they match exactly to the targeted solutions. This is also the case when we choose initial conditions different from the one sampled to compute the loss, as we see on Figure 2.14 (idem). This was achieved without penalisation (with $\alpha = 0$).

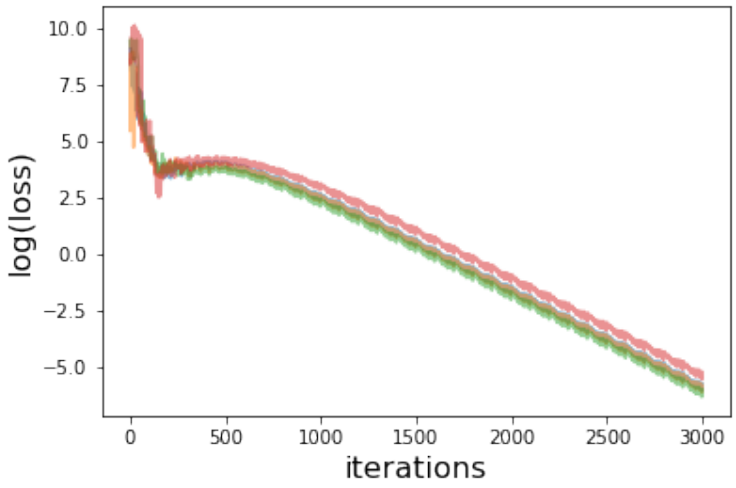


Figure 2.12: Variations of the loss during optimisation. The horizontal axis represents the iterations and the vertical axis the logarithm of the loss. Each colour represents the variations of the loss during one of the four descents we consider, each of them starting from a different θ_0. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 10]$. We have considered 5 trajectories starting at points uniformly sampled in $[0, 10]^2$ ((2.61654536, 6.97328654), (8.35401404, 7.58261582), (2.41885671, 6.60159081), (5.64960361, 4.81467692), (2.95476693, 9.69287747)). We have taken $F = F_3$ and $H = H_{\theta*}$ with $\theta^* = (0, 0, 0, -0.9, 0.1, -2.1, 4.3, 0, 0, 0, 0, 3.0, 0, 0, 0, 0)$. The learning rate was equal to 10 and the window size to $2\Delta t$. The first point of the window at a given step is the last point of the window at the previous step. We have taken $\alpha = 0$.

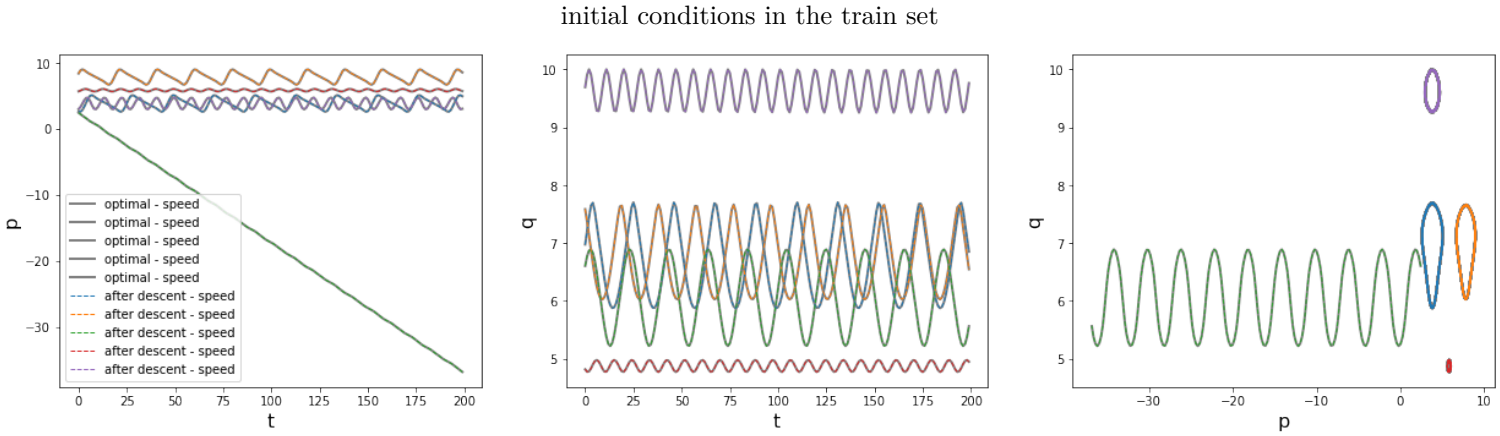


Figure 2.13: Solutions induced by the Hamiltonian obtained after 3500 iterations, where the loss reached its minimum. On the left is represented the speed in function of time, in the centre the position in function of time and on the right the trajectories in the phase space. targeted trajectories are represented in grey and the trajectories induced by H_θ are represented by coloured hashed lines. We obtained θ after the optimisation described on Figure 2.12. Initial conditions are the one used to compute the loss during the descent. Black stars represent the 5 initial points we have chosen. We have taken $\alpha = 0$.

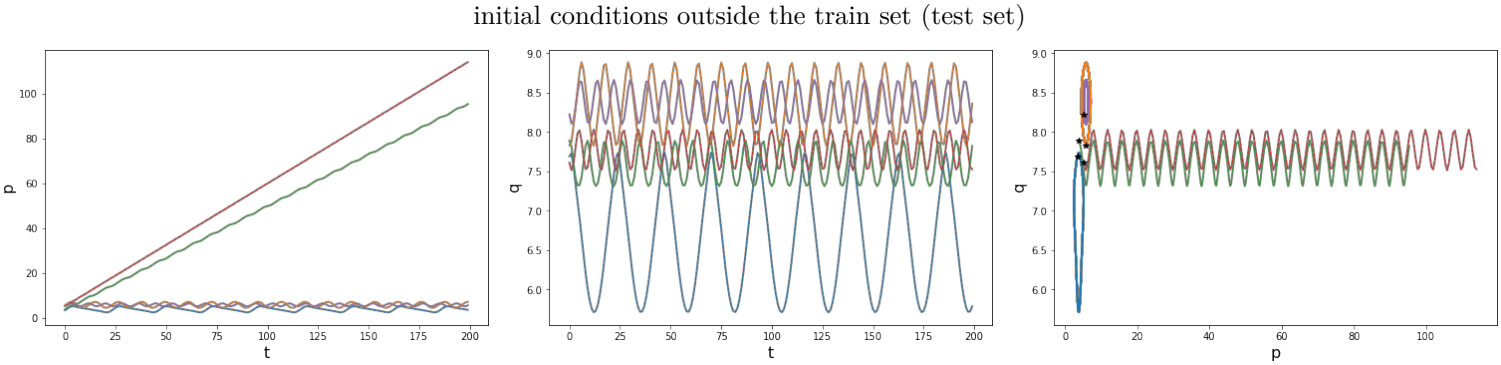


Figure 2.14: Solutions induced by the Hamiltonian obtained after 3500 iterations, where the loss reached its minimum. On the left is represented the speed in function of time, in the centre the position in function of time and on the right the trajectories in the phase space. targeted trajectories are represented in grey and the trajectories induced by H_θ are represented by coloured hashed lines. We obtained θ after the optimisation described on Figure 2.12. Initial conditions are different from the one used to compute the loss during the descent. Black stars represent the initial points we have chosen. We have taken $\alpha = 0$.

When we look at the iterates during optimisation, presented Figure 2.15, we see that the trajectory in the parameter space has the same behaviour as in the previous case: iterates jump from one point to another quite randomly at the beginning and the trajectory becomes smoother as we come closer to the optimum, to which we move towards in a quasi-straight line after some iterations.

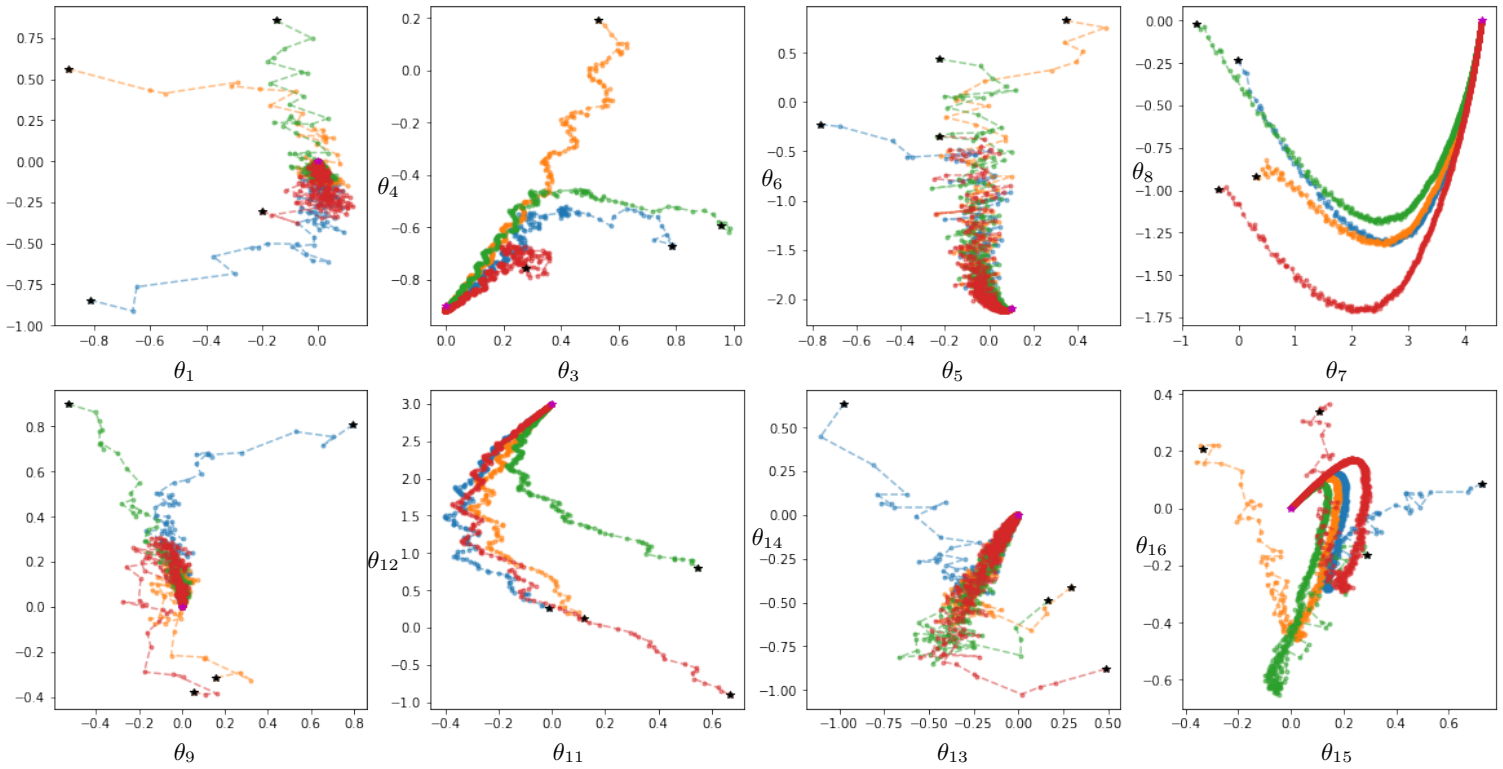


Figure 2.15: Iterations in the parameter space during optimisations presented on Figure 2.12. Each colour represents the iterates obtained during one of the four descents, starting at different θ_0, considered. Each graph shows the projection on two different orthogonal coordinates of $\mathbf{R}^K = \mathbf{R}^{16}$. The optimum is represented by pink stars and the θ_0 by black stars. We have taken $\alpha = 0$.

2.6.4 Hand-made non-convex problem in dimension 11 with penalisation

In the previous case, we did not need to use the penalisation to find the optimal value of θ. Here is a case were we need a penalisation to find the good θ.

We take $H : (p_x, p_y, q_x, q_y) \mapsto p_x p_y + q_x q_y - (1 - q_x)^2 q_y p_y$ and $F = F_4$ with F_4 the family composed of all monomial functions of degre 2 in $\mathbf{R}^4$ and $(x, y, z, t) \mapsto z^2 yt$. The targeted Hamiltonian is then in Span F and the optimal theta is $(0, 1, 0, 0, 0, 0, -1, 0, 1, 0, 1)$.

When we take $\alpha = 0$, the algorithm converges to a value of θ which is not the optimum but which gives accurate results. As we see on Figure 2.16, trajectories induced by the Hamiltonian obtained after optimisation are very close to true ones. This holds for the set that was used to compute the loss and for another set made of five other trajectories. In both cases, the mean error in L^2 norm between the true trajectories ${}^t Dx_g$ and the one induced by $H_{\theta,h}\hat{x}_{\theta,g}$ is of order

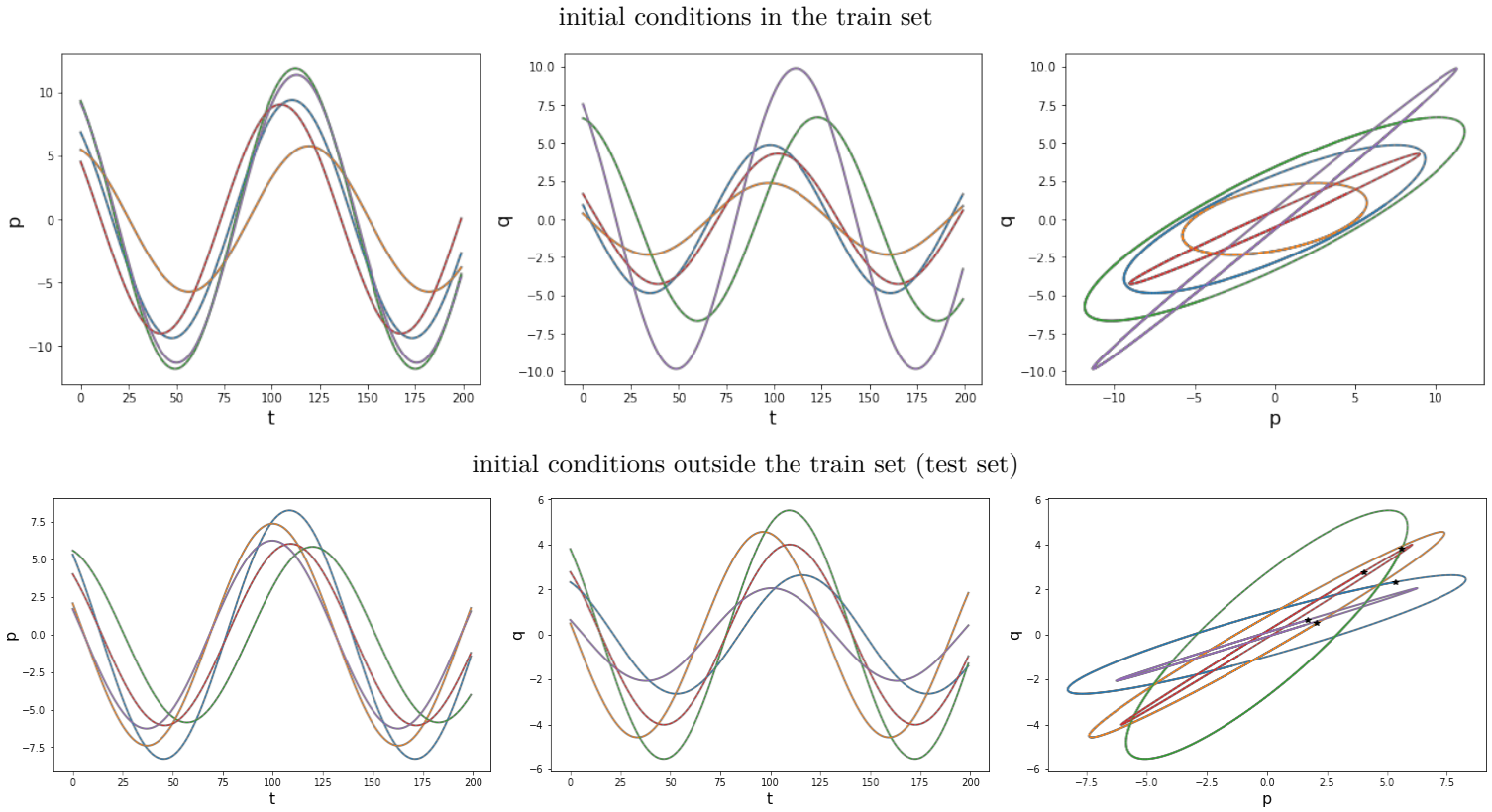


Figure 2.16: Solutions induced by the Hamiltonian obtained after 1000 optimisation steps. On the left is the speed according to the x direction in function of time, in the centre the position in x in function of time and on the right the trajectories in the phase space in x (we obtain similar results if we consider the phase space in y). targeted trajectories are represented in grey and the trajectories induced by H_θ by coloured hashed lines. We obtained θ after the optimisation with $\alpha = 0$, $w = 2\Delta t$ and $\rho = 1$. On the top, initial conditions are the one that were used to compute the loss during the descent. On the bottom, initial conditions were uniformly chosen in the range of the previous ones. Black stars represent these initial points. targeted trajectories are solutions of the Hamiltonian system with $H : (p_x, p_y, q_x, q_y) \mapsto p_x p_y + q_x q_y - (1 - q_x)^2 q_y p_y$ for 5 randomly chosen initial conditions. We have taken $F = F_4$.

In a tentative to reach the optimum, we have considered values of α, the coefficient in front of the penalisation term, that were not zero. We have successively taken $\alpha = 0.001$, $\alpha = 0.01$ and $\alpha = 0.1$ while keeping w and ρ unchanged. Table 2.1 shows how the mean error on targeted trajectories evolves with α. We see that it drastically increases when α grows. However, we also see on Figure 2.17 that the algorithm has converged to a θ which is closer to the targeted one when α is not zero. The better result is obtained with $\alpha = 0.001$: we did not reached the optimal value of θ but all but two of the coefficients of the value we obtained after 1000 iterations are equal to the good ones.

α	0	0.1	0.01	0.001
error	0.001	2.772	26.576	46.597

Table 2.1: Mean L^2 difference between the targeted trajectories and the ones induced by the Hamiltonian obtained after optimisation with different value of the penalisation coefficient α. targeted trajectories are solutions of the Hamiltonian system with $H : (p_x, p_y, q_x, q_y) \mapsto p_x p_y + q_x q_y - (1 - q_x)^2 q_y p_y$ for 5 randomly chosen initial conditions. We have taken $F = F_4$.

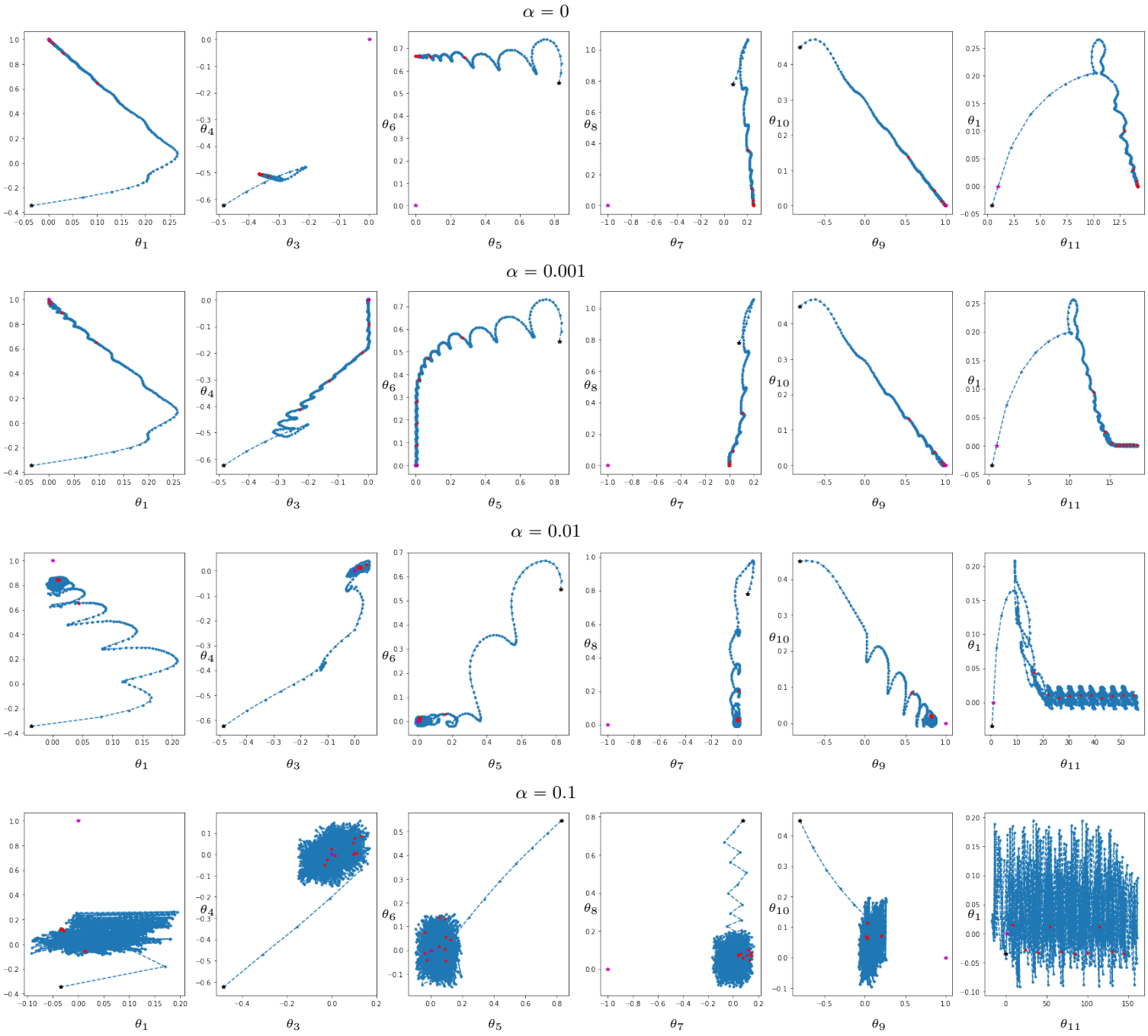


Figure 2.17: Iterations in the parameter space during optimisation process presented in 2.1 with $\alpha = 0$ (top), 0.001 (second line), 0.01 (third line), 0.1 (bottom). On each line, each graph represents the projection of the iterates on two different orthogonal directions of $\mathbf{R}^K = \mathbf{R}^{11}$. Since K is odd, the ordinates of the last graph of each line represents the same direction as the abscisses in the first graph. The optimum is represented by pink stars and θ_0 by black stars. Each iteration of number a multiple of 100 is represented by a red dot.

2.6.5 Projection on a trigonometric basis of size 49

We now want to see if Algorithm 2 still converges if the target Hamiltonian is not in the space spanned by the family F. To evaluate the performances of Algorithm 2 in this case, we consider the Hamiltonian $H : (p, q) \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$ and the family

$$F_5 = Cos_p \cup Cos_q \cup \{(p, q) \mapsto 1\},$$

where

$$Cos_p = \{(p, q) \mapsto \cos(\frac{2\pi}{n}p)\}_{1 \leqslant n \leqslant 24},$$
$$Cos_q = \{(p, q) \mapsto \cos(\frac{2\pi}{n}q)\}_{1 \leqslant n \leqslant 24}.$$

The whole family is of size 49. We have chosen the period of the functions such that the period of the trajectories induced by Hamiltonians in Span$F$ are approximately of the same size as the trajectories induced by H. The idea is to obtain a Fourier-like decomposition of the Hamiltonian.

To compute the loss, we have used 10 trajectories whose starting points were uniformly sampled in $[-1, 1]^2$. We have taken $w = 2\Delta t$ and $\rho = 10$. Figure 2.18 shows the decreasing of the loss during optimisation process. We obtain a Hamiltonian close to H near the 10 trajectories we used to compute the loss: on Figure 2.19 we present the trajectories induced by H_θ after optimisation, and we see that they are very close to the targeted ones. At its minimum during optimisation, the loss was underneath 0.085.

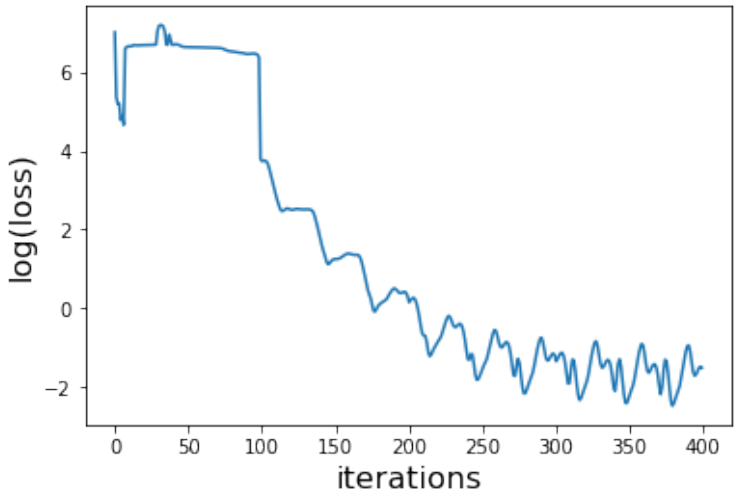


Figure 2.18: Variations of the loss during optimisation. The horizontal axis represents the iterations and the vertical axis the logarithm of the loss. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 10]$. We have considered 10 trajectories starting at points uniformly sampled in $[-1, 1]$ $((0.64400967, 0.44432577)$, $(0.08211142, 0.93158362)$, $(0.93544462, 0.80733196)$, $(0.53982471, -0.64419584)$, $(0.92563482, -0.3046332)$, $(-0.80676766, 0.29964909)$, $(-0.34093966, -0.26624007)$, $(0.32299278, 0.55533709)$, $(0.64681662, 0.0365833)$, $(0.23438805, 0.62507881))$. We have taken $F = F_5$ and $H : (p, q) \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. The learning rate was equal to 10 and the window size to $2\Delta t$. The first point of the window at a given step is the last point of the window at the previous step.

initial conditions in the train set

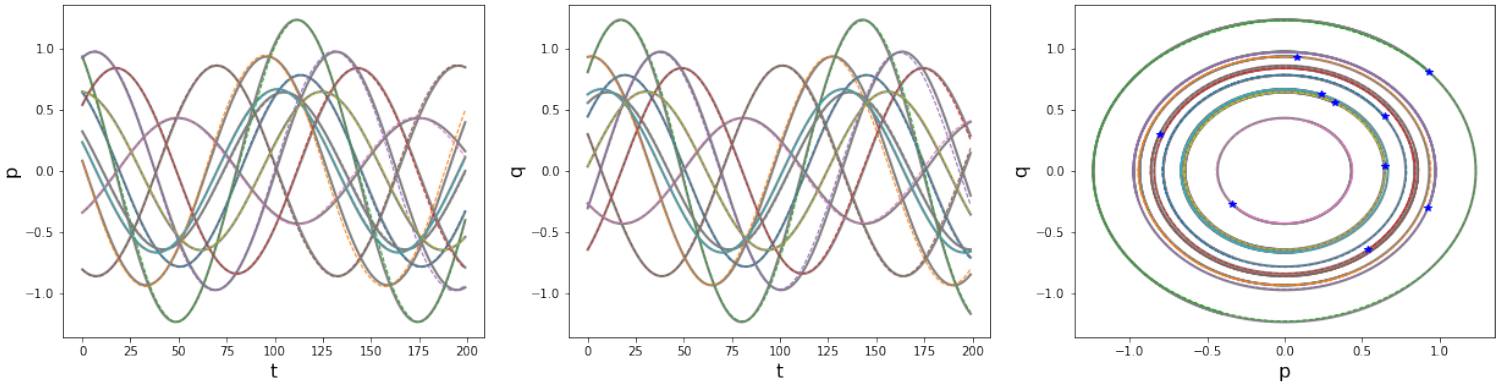


Figure 2.19: Solutions induced by the Hamiltonian at iteration 379, where the loss reached its minimum. On the left is represented the speed in function of time, in the centre the position in function of time and on the right the trajectories in the phase space. targeted trajectories are represented in grey and the trajectories induced by H_θ are represented by coloured hashed lines. We obtained θ after the optimisation described on Figure 2.18. Initial conditions are the one that were used to compute the loss during the descent. Black stars represent the 10 initial points we have chosen.

It happens however that the initial point we sampled to obtain the results presented on Figures 2.18 and 2.19 was particularly good. In fact, as shown on Figures 2.20 and 2.21, all initialisation points θ_0 do not lead to a Hamiltonian H_θ as good as previously. When we continue the descent, we see that loss often starts oscillating without decreasing any more after having reached a certain value. This behaviour is presented on Figure 2.22. In those cases, we usually see that the iterates θ_k does not change a lot. These oscillations are present even if we reduce the learning rate, until it is so small that we do not change the iterates. This seems to mean that the descent is blocked at a certain value. Yet, we did not try to modify the window size during the algorithm. Further tests are needed to see if we can revive the descent this way.

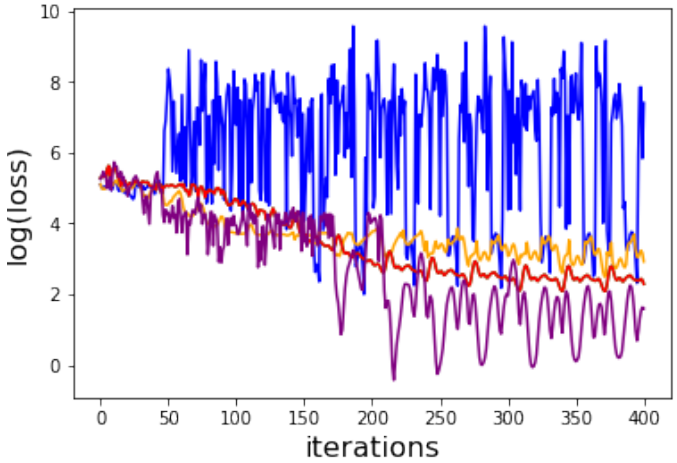


Figure 2.20: Variations of the loss during optimisation. The horizontal axis represents the iterations and the vertical axis the logarithm of the loss. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 10]$. We have considered 10 trajectories starting at points uniformly sampled in $[-1, 1]$ (the same points as in Figure 2.18). We have taken $F = F_5$ and $H : (p, q) \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. The learning rate was equal to 10 and the window size to $2\Delta t$. The first point of the window at a given step is the last point of the window at the previous step.

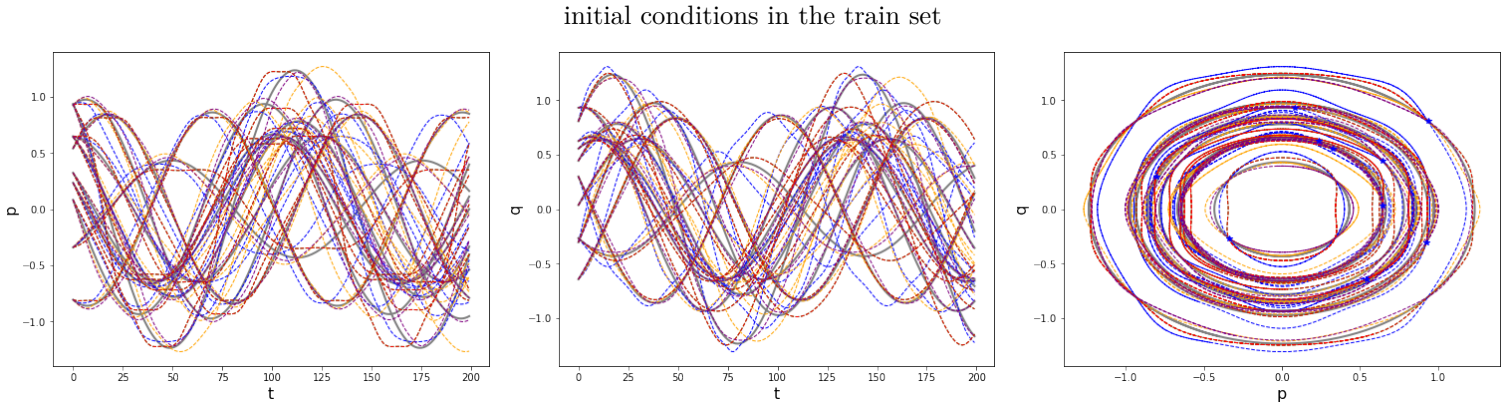


Figure 2.21: Solutions induced by the Hamiltonian at the argmin iteration of the descents starting at 4 different θ_0. On the left is represented the speed in function of time, in the centre the position in function of time and on the right the trajectories in the phase space. targeted trajectories are represented in grey and the trajectories induced by H_θ are represented by colored hashed lines. We obtained θ after the optimisation described on Figure 2.20. Initial conditions are the one that were used to compute the loss during the descent.

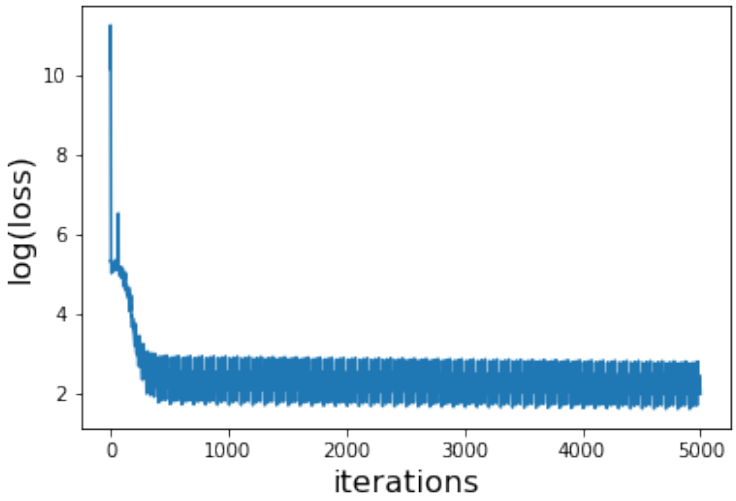


Figure 2.22: Variations of the loss during optimisation. The horizontal axis represents the iterations and the vertical axis the logarithm of the loss. Primal and dual states were computed using a time step of $\Delta t = 0.05$ on $[0, 10]$. We have considered 10 trajectories starting at points uniformly sampled in $[-1, 1]$ (the same points as in Figure 2.18). We have taken $F = F_5$ and $H : (p, q) \mapsto \frac{1}{2}p^2 + \frac{1}{2}q^2$. The learning rate was equal to 10 and the window size to $2\Delta t$. The first point of the window at a given step is the last point of the window at the previous step.

Consider $H_{\theta_{best}}$, the best Hamiltonian function found after the descent presented on Figures 2.20 and 2.21. The loss at θ_{best} reaches 0.65. On Figure 2.23, we show the solutions we obtain with $H_{\theta_{best}}$ when we start from initial conditions that are different from the ones we use to compute the loss and its gradient during descent. We see that the solutions are quite accurate for initial conditions located in the range of the initial conditions we considered during the optimisation but are very bad for the initial condition which is further from them.

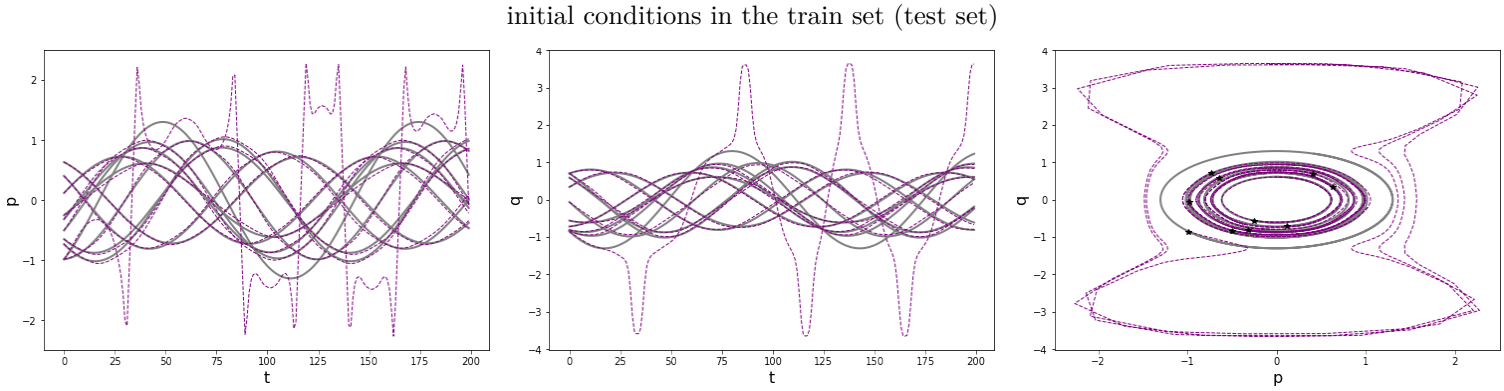


Figure 2.23: Solutions induced by the Hamiltonian obtained after the omptimisation presented on Figure 2.20. On the left is represented the speed in function of time, in the centre the position in function of time and on the right the trajectories in the phase space. targeted trajectories are represented in grey and the trajectories induced by $H_{\theta_{best}}$ are represented by coloured hashed lines. We obtained θ_{best} after the optimisation described on Figure 2.20.

2.6.6 Tests on the reduction problem

For the moment, the tests we made in the context of a reduced order model did not give satisfying results. Recall that in this case, we are looking for the value of θ such that the trajectories induced by H_θ, x_θ, when they are decompressed by D, are as close as possible to the trajectories induced by the original Hamiltonian function, H.

We made the tests on the non-linear piano string model, with D the decoder given by the PSD for $k = 2$. Here, the parameter g lies in $G = [0, 0.2]^2$ and represents the initial deformation of the string in the oscillation plane. To build the PSD, we have uniformly sampled 25 values of g in G and considered the associated trajectories for t in $[0, 0.5]$. Then, we have sampled 16 values of g in G, different from the previous ones, have computed the associated trajectories x_g in high dimension and have taken ${}^tDx_g$ as targets for the optimisation problem. We present them on Figure 2.24.

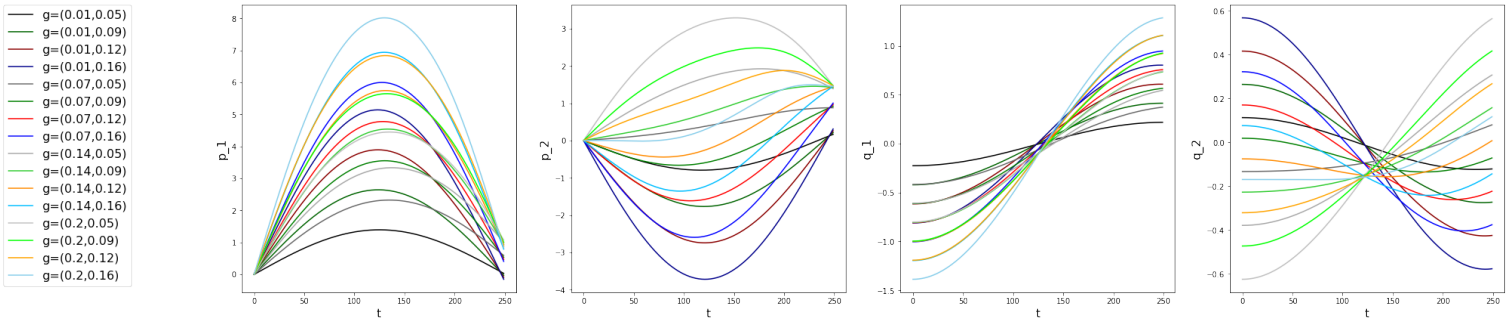


Figure 2.24: targeted trajectories in the reduced order model optimisation problem for t in $[0, 0.5]$. Trajectories have been computed in high dimension from 16 values of the parameter g, uniformly sampled in $G = [0, 0.2]$ and different from the values that were taken to build the PSD, and send to the low dimensional space $\mathbf{R}^4$ with ${}^tD$, where D is the decoder build by the PSD.

We have tried different families F but we have never reached a satisfying Hamiltonian. The

families we have used were composed of sinusoidal functions of $p_1, p_2, q_1, q_2, p_1p_2$ and q_1q_2 of different amplitudes but none of them had led to a good Hamiltonian. We still are looking for a family that would give a better results.

2.6.7 Perspectives and remaining tests

In future work, it would be interesting to continue studying the method presented here. At least two aspects should be explored in greater depth. From the theoretical point of view, it would be interesting to explain how works the optimisation method we proposed (Algorithm 2) and give a proof of its convergence. We expect to achieve this using policy-gradient theory and Hamilton-Jacobi-Bellman equation.

From the numerical point of view, it would be interesting to conduct further tests, in particular in the case of reduced order model, on which we currently do not have satisfactory results. We are particularly interested in what happens when we change the window size during optimisation. This is an additional parameter that has Algorithm 2 compared to traditional gradient descent and we can imagine that an optimal way to use it does not necessarily involve to fix the window size.

Part II

Geometric part

3 Homotopy principle

3.1 Goals

The aim of the geometrical part of this work is to give theoretical justifications to the methods we develop in the numerical part. In particular, we would like to know if there is no geometrical obstacle to learning the manifold Σ^{2k} together with the reduced flow of H. More precisely, we would like to prove the following conjecture:

Conjecture. *Let $n, k \in \mathbf{N}$ such that $k \ll n$. Consider two k-dimensional manifolds Σ^k and $\tilde{\Sigma}^k$ embedded in $\mathbf{R}^{2n}$ endowed with its usual symplectic structure. Denote by $i : \Sigma^k \to \mathbf{R}^{2n}$ and $\tilde{i} : \tilde{\Sigma}^k \to \mathbf{R}^{2n}$ the corresponding inclusions.*

If k is sufficiently small in front of n, then there exists a symplectic homeomorphism

$$h : \mathbf{R}^{2n} \to \mathbf{R}^{2n}$$

such that $h(\Sigma^k) = \tilde{\Sigma}^k$. Moreover, if i and $\tilde{i}$ are $\mathcal{C}^0$-close, then h is $\mathcal{C}^0$-close from the identity.

If this result is true, then for all Hamiltonian function $H : \mathbf{R}^{2n} \to \mathbf{R}$ whose flow preserves Σ^{2k}, the flow of the composition $\tilde{H} = H \circ h^{-1}$ preserves $\tilde{\Sigma}^k$. This is immediate since $\phi^t_{\tilde{H}} = h \circ \phi^t_H \circ h^{-1}$. In the case Σ^{2k} and $\tilde{\Sigma}^k$ are $\mathcal{C}^0$-close, then H and $\tilde{H}$ are also $\mathcal{C}^0$-close and the restriction of their flows on bounded intervals of $\mathbf{R}$ too. In other words, if we make a small error when learning the solution manifold, which is highly probable when we interpolate it with a finite number of points, then the part of the errors on solutions induced by errors we made on Σ^{2k} remains small. If this result is true, then we can hope to learn the dynamics on the interpolated manifold as we try to do.

This result has already been proved in the particular case of isotropic submanifolds in [4] using methods based on Gromov's h-principle. The h-principle, an abbreviation for *homotopy principle*, is a principle or a characteristic of some spaces which, if it holds, guaranty the existence of solutions for differential problems. It involves a new point of view on differential equalities and inequalities, involving for example notions of jets and differential relations. When working with the h-principle, one usually wants to establish it and there are some particular techniques to achieve this.

During this internship, we worked to understand the methods that were used to prove the results in [4]. Eventually, we will use them to extend the proof of 3.1 to the general case. To become familiar with the h-principle, we read the parts of [9] devoted to the h-principle and its proofs using holonomic approximation theorem. In this chapter, we present the main notions that one needs to understand what is the h-principle and some of its applications. We present all these notions in a way that the chapter leads to the proof of the following theorem:

Theorem 3.1. *[9] Let (V, ω_V) and (W, ω_W) be two symplectic manifolds of respective dimensions $n = 2l$ and $q = 2m$. Let $f_0 : V \to W$ be an embedding such that $f_0^*[\omega_W] = [\omega_V]$. Suppose also that $F_0 = df_0$ is homotopic to an isosymplectic homomorphism F_1 via $F_t \subset \mathcal{R}_{imm}$ such that $bsF_t = f_0$ for all $t \in [0, 1]$.*

Then, if V is open and $m < l$, there exists an isotopy $f_t : V \to W$ such that f_1 is isosymplectic and df_1 is homotopic to F_1 in the isosymplectic homomorphisms. Moreover, if K is a core of V, we can choose f_t arbitrarily $\mathcal{C}^0$-close to f_0 near K.

Recall that:

Definition 3.2 (Symplectomorphism, isosymplectomorphism)**.** *Let (V, ω_V) and (W, ω_W) be two symplectic manifolds and $f : V \to W$ a map between them. We say that f is a* symplectomorphism *if $f^*\omega_W$ is a symplectic form on V and an* isosymplectomorphism *if $f^*\omega_W = \omega_V$.*

All the definitions and the results we present here are taken from [9] and will be explored in the following section. The proofs are taken from the same reference, but we had details in most of them.

In what follows, we are sometimes required to consider a metric on some manifolds. Every time we talk about $\mathcal{C}^0$-closeness of applications, we imply the existence of a distance on the space where those functions take their values. We also need it to build normal neighbourhoods in some proofs. A simple way to define a distance on manifolds is to consider a Riemannian metric on this manifold, and we know that this is always possible (see [10] for a proof of this assertion). When needed, we therefore consider a Riemannian structure on the considered manifold. This additional structure is only useful to properly define normal neighbourhoods or $\mathcal{C}^0$-closeness, it does not change anything to the geometry of the problems we consider.

3.2 Definitions

3.2.1 Jets

Jet spaces

For our purpose, we are interested on derivatives of functions. The notion of jets allows us to designate and manipulate the functions along with their derivatives. Let us first see how we define jets in the simple case of functions of $\mathbf{R}^l$.

Definition 3.3 (Space of r-jets on Euclidean spaces [9])**.** *Let $r \in \mathbf{N}^*$. The* space of r-jets *of functions $\mathbf{R}^n \to \mathbf{R}^q$ is the space of all* a priori *possible values of a function $f : \mathbf{R}^n \to \mathbf{R}^q$ and its derivatives of order at most r at a point of $\mathbf{R}^n$, that is*

$$\mathbf{R}^n \times \mathbf{R}^q \times \mathbf{R}^{qd(n,1)} \times ... \times \mathbf{R}^{qd(n,r)},$$

where $d(m, l)$ is the number of partial derivatives of order l for a function $f : \mathbf{R}^m \to \mathbf{R}$. We note this space $J^r(\mathbf{R}^n, \mathbf{R}^q)$.

We call r-jet extension of f at v *and note $J^r_f(v)$ the element of $J^r(\mathbf{R}^n, \mathbf{R}^q)$ induced by a function f at a point $v \in \mathbf{R}^n$, that is the set of v and the values taken at v by f and its derivatives of order at most r.*

The space of all possible values that can take a function and its derivative at a point $v \in \mathbf{R}^n$ is really the product we have mentioned: for any point P in this product space such that $\pi_1(P) = v$, there exists a polynom of degree r in $\mathbf{R}^n$ whose r-order derivatives at v agree with P.

Remark 3.4. We have $d(n,r) = \frac{(n+r-1)!}{(n-1)!r!}$. This can be proved by induction on n. The formula is clearly true for $n = 1$. Suppose now that the formula is true for all $i \leqslant n$. We have

$$d(n,r) = \sum_{i_1=0}^{n} \sum_{i_2=0}^{i_1} ... \sum_{i_r=0}^{i_{r-1}} 1$$

so $d(n+1,r) = d(n,r) + d(n+1,r-1)$. An immediate induction on r gives then

$$d(n+1,r) = \sum_{i=2}^{r} d(n,i) + d(n+1,1).$$

Since $d(n,1) = n$ and $d(n,0) = 1$ for any n, this can be rewritten as $d(n+1,r) = \sum_{i=0}^{r} d(n,i)$. By assumption, this is equivalent to $d(n+1,r) = \sum_{i=0} \left(\frac{n-1+i}{n-1}\right)$. Using the *hockey cross* formula,

$$\sum_{j=0}^{m-k} \left(\frac{j+k}{k}\right) = \left(\frac{m+1}{m-k}\right),$$

which is true for all $m, k \in \mathbf{N}$ such that $k < m$, we get

$$d(n+1,r) = \left(\frac{n+r}{r}\right) = \frac{(n+r)!}{n!r!}.$$

We then have proved that the formula is also true for $n+1$. By the induction principle, we deduce that the formula is true for all $n > 1$.

We define now jets on manifolds. For this purpose, we use the definition we have given on Euclidean spaces. We simply need to adapt it to make it invariant by a change of coordinates.

Definition 3.5 (Space of r-jets in the general case [9])**.** *Let V and W two manifolds of respective dimensions n and q. Let $v \in V$ and $U \subset V$ be an open neighbourhood of v in V on which is defined a coordinate system $\phi : U \to \mathbf{R}^n$. We say that two functions f and g from U to W are* r-tangent *at v if they agree at v and if all the r-order derivatives of $\psi \circ f \circ \phi^{-1}$ and $\psi \circ g \circ \phi^{-1}$ agree at $\phi(v)$ for $\psi : Op_w W \to \mathbf{R}^q$ a coordinate chart defined in the neighbourhood of $w = f(v) = g(v)$.*

Tangency at v gives rise to an equivalence relation: two functions are in the same class if and only if they are r-tangent. The space of r-jets *$J^r(V,W)$ is defined as the space of all r-tangency classes at any point of V. We call* r-jet extension of f at v *and note $J^r_f(v)$ the class of r-tangency of f at v.*

When $V = \mathbf{R}^n$ and $W = \mathbf{R}^q$, the previous definition is equivalent to the first one.

With the chain rule, we verify that this definition is indeed invariant under a change of coordinates. Let $\phi_1, \phi_2 : U \to \mathbf{R}^n$ be two charts on V and $\psi_1, \psi_2 : U \to \mathbf{R}^q$ be two charts on W. We have

$$\begin{aligned} d_{\phi_1(v)}(\psi_1 \circ f \circ \phi_1^{-1}) &= d_{\phi_1 \circ \phi_2^{-1} \circ \phi_2(v)}(\psi_1 \circ \psi_2^{-1} \circ \psi_2 \circ f \circ \phi_2^{-1} \circ \phi_2 \circ \phi_1^{-1}) \\ &= d_{\psi_2 \circ f(v)}(\psi_1 \circ \psi_2^{-1}) \circ d_{\phi_2(v)}(\psi_2 \circ f \circ \phi_2^{-1}) \circ d_{\phi_1(v)}(\phi_2 \circ \phi_1^{-1}) \end{aligned}$$

Since $d_{\psi_2\circ f(v)}(\psi_1\circ\psi_2^{-1})$ and $d_{\phi_1(v)}(\phi_2\circ\phi_1^{-1})$ are invertible,

$$d_{\phi_1(v)}(\psi_1\circ f\circ\phi_1^{-1}) = d_{\phi_1(v)}(\psi_1\circ g\circ\phi_1^{-1}) \quad\Longleftrightarrow\quad d_{\phi_2(v)}(\psi_2\circ f\circ\phi_2^{-1}) = d_{\phi_2(v)}(\psi_2\circ g\circ\phi_2^{-1}).$$

If we replace f and g by partial derivatives of order inferior to r, we obtain the invariance of the notion of r-tangency for $r > 1$.

Note that for $l < m$, the projection $p_l^m : J^m(V,W) \to J^l(V,W)$, which sends a class of m-tangency to a class of l-tangency by forgetting the derivatives of order superior to l, is invariantly defined. In fact, if two functions are m-tangent at a point v, they are also l-tangent at the same point for all $l \leqslant m$.

On the contrary, the inclusions $J^l(V,W) \subset J^m(V,W)$ are not invariant under a change of coordinates: if we want to set an inclusion function $i : J^l(V,W) \to J^m(V,W)$, we can not simply complete the l-jets with zeros. To see that, take for example $n = 1$, $q = 2$ and the function whose expression in polar coordinates is $f : x \mapsto (x,x)_{r,\theta}$. In Cartesian coordinates, we have $f : x \mapsto (x\cos(x), x\sin(x))_{u,v}$. Thus, $f''_r(x) = f''_\theta(x) = 0$ while $f''_u(x) = -2\sin(x) - \cos(x)$ and $f''_v(x) = 2\cos(x) - x\sin(x)$ for all x in $\mathbf{R}$. This illustrates the fact that extending a jet by 0 does not give the same thing, depending on the coordinates we have chosen on W. This is also true for any choice that we could make to extend the l-jets into m-jets. Therefore, those inclusions are not invariant by a change of coordinates on W.

Remark 3.6. For a fibre bundle $\pi : E \to V$, denote Sec$E$ the set of its sections. In the following, we will consider functions $f : V \to W$ as sections of the trivial fibre bundle $p : V \times W \to V$. We will always suppose that they are of class $\mathcal{C}^\infty$.

Note now that we can replace the projection on V $\pi_1 : V \times W \to W$ by any smooth fibre bundle $p : X \to V$ with X a manifold of dimension $n + q$. The notions of r-tangency and r-jets we have given in the case $X = V \times W$ can be extended to X in the following way. Note by W the fibre of the bundle $p : X \to V$. For $v \in V$, consider a local trivialisation $\varphi : p^{-1}(U) \subset X \to U \times W$ in the neighbourhood U of v in V. For any section $f : U \to p^{-1}(U)$, the map $\varphi \circ f$ is a local section of the trivial bundle $\pi : V \times W \to V$ and has a class of r-tangency at v in the sense of Definition 3.5. Let us see that this class is independent of φ. Take another local trivialisation $\psi : p^{-1}(U) \to V \times W$. Since φ and ψ preserves fibre, this is also the case of the map $\varphi \circ \psi^{-1}$ which is therefore equivalent to a change of coordinates on every fibre $W \times \{v'\}$ for $v' \in U$. Now, what might affect the class of r-tangency of f at v is only the change of coordinates induced on $W \times \{v\}$. Then, the invariance of the notion of r-tangence under coordinate changes on W makes that the class of r-tangency of $\varphi \circ f$ does not depend on the local trivialisation φ.

In the general case, the space of r-jets is denoted $X^{(r)}$. Of course, when $X = V \times W$, we have $X^{(r)} = J^r(V,W)$. When the extension to the general case is immediate, the following results will be presented in the case $X = V \times W$. The reason for this is that it is much more intuitive to think of $J^r(V,W)$ rather than $X^{(r)}$, and we will only use the non-trivial case sporadically.

In fact, the only non-trivial bundle that we will consider is the exterior bundle $p : \Lambda^p V \to V$ above a manifold V, with $\Lambda^p V = \bigsqcup_{v\in V} \Lambda^p T_v V$ and $p(a \in \Lambda^p T_v V) = v$. In this context, the fibre above $v \in V$ is the space $\Lambda^p T_v V$ of alternate p-forms on $T_v V$ and a smooth section of $\Lambda^p V \to V$ is a differential p-form.

Jet extensions of functions

Consider the fiber bundle $p : X \to V$. The notion of r-jets at points v of V gives rise to an another fibre bundle, $p^r : J^r(V,W) \to V$, where $p^r := p \circ p_0^r$ associates to a r-jet the point of V

in which it is defined. Consider on X a smooth atlas whose charts preserve fibres (in the case $X = V \times W$, we can take the atlas induced by the Cartesian product of V and W). Then, we define a smooth atlas on $X^{(r)}$ by extending each coordinate chart $\varphi : U \subset X \to \mathbf{R}^n \times \mathbf{R}^q$ in a coordinate chart

$$\varphi^r : (p_0^r)^{-1}(U) \subset X^{(r)} \to J^r(\mathbf{R}^n, \mathbf{R}^q) \sim \mathbf{R}^{n+q(1+d(n,1)+\ldots+d(n,r))},$$

where φ^r associates a r-jet, or class of r-tangency, to its image by φ in $J^r(\mathbf{R}^n, \mathbf{R}^q)$. As we have defined a class of r-tangency as the inverse image of classes in $J^r(\mathbf{R}^n, \mathbf{R}^q)$ by a coordinate chart, we build this way a smooth structure on the jets space.

Proposition 3.7. *Let $\{(A_i, \varphi_i)\}_i$ be a smooth atlas on X which preserves fibres of $p : X \to V$. We can extend it to a smooth atlas $\{((p_0^r)^{-1}(A_i), \varphi_i^r)\}_i$ on $X^{(r)}$.*

Proof. Let $\varphi : p^{-1}(U) \to X$ be a fiber preserving coordinate local chart on X. Given an element s of $X^{(r)}$, it exists a local section

$$f_s : Op_{\{p^r(s)\}} \subset V \to X$$

such that the r-tangency class of f_s at $p^r(s)$ is s: $J^r_{f_s}(p^r(s)) = s$. On the other hand, since the chart φ preserves fibers, the formula $p^{-1} \circ \varphi \circ \pi_1$, where $\pi : \mathbf{R}^n \times \mathbf{R}^q \to \mathbf{R}^n$ is the trivial bundle, defines a coordinate map $\varphi_V : V \to \mathbf{R}^n$. Then, we define φ^r as the application

$$s = J^r_{f_s}(p^r(s)) \mapsto J^r_{\varphi \circ f_s \circ \varphi_V^{-1}}(\varphi_V \circ p^r(s)) \in J^r(\mathbf{R}^n, \mathbf{R}^q).$$

This element does not depend on the choice of f_s since its construction only involves the values of f_s and its derivative at $p^r(s)$, which are determined by the class of r-tangency s that we consider. This shows that what we called φ^r really is a map from $(p^r)^{-1}(U) \subset X^{(r)}$ to $(\pi^r)^{-1}(\varphi_V(U))$.

This map admits an inverse: for an element $\mathbf{s}$ of $J^r(\varphi_V(U), \mathbf{R}^q)$ and a local section

$$f_{\mathbf{s}} : \varphi_V(U) \to \mathbf{R}^n \times \mathbf{R}^q$$

such that $J^r_{f_{\mathbf{s}}}(\pi^r(\mathbf{s})) = \mathbf{s}$, the section $\varphi^{-1} \circ f_{\mathbf{s}} \circ \varphi_V$ is defined above U and take its values in X so

$$\Phi(\mathbf{s}) := J^r_{\varphi^{-1} \circ f_{\mathbf{s}} \circ \varphi_V}(\varphi_V^{-1} \circ \pi^r(\mathbf{s}))$$

is an element of X^r above U. If $\mathbf{s}$ can be written as

$$J^r_{\varphi \circ f_s \circ \varphi_V^{-1}}(\varphi_V \circ p^r(s)),$$

we can take $f_{\mathbf{s}} = \varphi \circ f_s \circ \varphi_V^{-1}$, which makes

$$\Phi(\mathbf{s}) = J^r_{f_s}(\varphi_V \circ \pi^r(\mathbf{s})).$$

Clearly,

$$\pi^r(J^r_{\varphi \circ f_s \circ \varphi_V^{-1}}(\varphi_V \circ p^r(s))) = \varphi_V \circ p^r(s)$$

so $\Phi(\mathbf{s}) = J^r_{f_s}(p^r(s)) = s$. We have then that $\Phi = (\varphi^r)^{-1}$, which proves that φ^r is bijective.

We now have to show that for another map ψ^r defined from a local coordinate chart

$$\psi : U' \subset X \to \mathbf{R}^n \times \mathbf{R}^q$$

such that $E = U \bigcap U' \neq \varnothing$, the change of coordinates $\psi^r \circ (\varphi^r)^{-1}$ is smooth. Let $\mathbf{s} \in J^r(\mathbf{R}^n, \mathbf{R}^q)$ be a r-jet above an element of $\varphi_V(E)$ and $f_\mathbf{s}$ as previously defined. Note $s = (\varphi^r)^{-1}(\mathbf{s})$ and take $f_s = \varphi^{-1} \circ f_\mathbf{s} \circ \varphi_V$. We have

$$\psi^r \circ (\varphi^r)^{-1}(\mathbf{s}) = \psi^r(s) = J^r_{\psi \circ \varphi^{-1} \circ f_\mathbf{s} \circ \varphi_V \circ \psi_V^{-1}}(\psi_V \circ p^r(s))$$

where

$$\psi_V \circ p^r(s) = \psi_V \circ p^r(J^r_{\varphi^{-1} \circ f_\mathbf{s} \circ \varphi_V}(\varphi_V^{-1} \circ \pi^r(\mathbf{s}))) = \psi_V \circ \varphi_V^{-1} \circ \pi^r(\mathbf{s}).$$

Now, we know that on $\mathbf{R}^n \times \mathbf{R}^q$, we can choose $f_\mathbf{s}$ as the unique polynomial function $p_\mathbf{s}$ of degree r whose derivatives agree with $\mathbf{s}$ at $\pi^r(\mathbf{s})$. The coordinates of $p_\mathbf{s}$ are smooth functions of $\mathbf{s} \in J^r(\mathbf{R}^n, \mathbf{R}^q)$ and since we assumed that $\psi \circ \varphi^{-1}$ and $\varphi_V \circ \psi_V^{-1}$ are smooth, the values of $\psi \circ \varphi^{-1} \circ f_\mathbf{s} \circ \varphi_V \circ \psi_V^{-1}$ and $\psi_V \circ \varphi_V^{-1} \circ \pi^r(\mathbf{s})$ and of their derivatives at a given point also varies smoothly with $\mathbf{s}$. By composition, we get that $\psi^r \circ (\varphi^r)^{-1}$ is a smooth function. □

This allows us to consider regular sections of the jet bundle and from now on, all the ones we consider are supposed to be $\mathcal{C}^\infty$. We note bs$F$ the image by p_0^r of F, in other words the section of $X \to V$ induced by a section F of the jet bundle, forgetting all the derivatives and keeping only the underlying function. Conversely, any section $f : V \to X$ gives rise to a section $J_f^r : V \to X$, which associates to each point v in V the r-class of tangency of f at v. It is called the *r-jet extension of f*.

All the jet sections are not r-jet extensions. Those which has this property, that is F in Sec$x^{(r)}$ such that there exists $f : V \to X$ with $F = J_f^r$ are called *holonomic sections*. The set of holonomic section of $p : X^{(r)} \to V$ is denoted Hol$X^{(r)}$.

Example 3.8. Let us illustrate all these notions with two examples.

- Take $n = 1$, $q = 2$, $V = \mathbf{R}$ and $W = \mathbf{R}^2$. The space of r-jets is

  $$J^r(\mathbf{R}, \mathbf{R}^2) = \mathbf{R} \times \mathbf{R}^2 \times \mathbf{R}^{r \times 2 \times 1}.$$

  For any element $d = (v, a_1, a_2, b_1, b_2, c_1, c_2)$ of $J^2(\mathbf{R}, \mathbf{R}^2)$, the polynom

  $$P : x \mapsto (a_1 + b_1(x - v) + \frac{1}{2}c_1(x - v)^2, a_2 + b_2(x - v) + \frac{1}{2}c_2(x - v)^2)$$

  around v represents the 2-tangency class of d. For a function $f : \mathbf{R} \to \mathbf{R}^2$, we have $F := J_f^r(v) = (v, f(v), f'(v), ..., f^{(r)}(v))$. The section F is then a holonomic function, such that bs$F = f$. For other functions g, h, the map $v \mapsto (v, f(v), g(v), h(v))$ is also a section of $J^2(\mathbf{R}, \mathbf{R}^2)$ but is *a priori* not holonomic.

- Take any n and q and consider the space of 1-jets $J^1(\mathbf{R}^n, \mathbf{R}^q)$. An element of this space is a triplet $(x, y, A) \in \mathbf{R}^n \times \mathbf{R}^q \times \mathcal{M}_{n,q}(\mathbf{R})$. The point x is a point of $\mathbf{R}^n$ above which we consider the value of a local function, y is the point of $\mathbf{R}^q$ on which is sent x by this function and A represents the Jacobian matrix of this function at x.

3.2.2 Differential relations

A lot of categories of functions that we use to manipulate in geometry are defined using differential equations, that are conditions on the derivatives of order 1 or more. This is for example the case of immersions, submersions, diffeomorphisms or symplectomorphisms. Using the notion of jets, we can give another view on these categories.

Definition 3.9 (Differential relation [9])**.** *A* differential relation of order r *between* V *and* W *is a subset of the* r*-jets space* $J^r(V,W)$*.*

The conditions that define groups of functions in which we are interested here can be described in terms of differential relation. When we use h-principle, we work on the jet spaces and we use geometric notions such as immersions or symplectomorphisms through the differential relation they induce. For example, if we want to prove the existence of an immersion, we first start by considering formal solutions of the differential relation induced by the fact of being an immersion. Then, we look for holonomic sections among those formal solutions.

Example 3.10. We describe here the differential relations induced by groups of functions we are interested in.

- **Immersions:** we say that a function $f : V \mapsto W$ is an immersion if its differential is injective at each point of V. This condition only involves the differential of f so the corresponding differential relation $\mathcal{R}_{imm}(V,W)$ is a subset of $J^1(V,W)$. It is precisely the set of all possible values that can take an immersion and its differential above any point $v \in V$. In other words, it is the set of injective linear maps (monomorphisms) from T_vV to T_wW for any pair $(v,w) \in V \times W$. When $V = \mathbf{R}^n$ and $W = \mathbf{R}^q$, $\mathcal{R}_{imm}(V,W)$ is exactly $\{(x,y,A) \in \mathbf{R}^n \times \mathbf{R}^q \times \mathcal{M}_{n,q}(\mathbf{R}) \quad | \quad rg(A) = n\}$.

- **Submersions:** we say that a function $f : V \mapsto W$ is an submersion if its differential is surjective at each point of V. This condition only involves the differential of f so the corresponding differential relation $\mathcal{R}_{surj}(V,W)$ is a subset of $J^1(V,W)$. It is precisely the set of all possible values that can take an submersion and its differential above any point $v \in V$. In other words, it is the set of surjective linear maps (epimorphisms) from T_vV to T_wW for any pair $(v,w) \in V \times W$. When $V = \mathbf{R}^n$ and $W = \mathbf{R}^q$, $\mathcal{R}_{surj}(V,W)$ is exactly $\{(x,y,A) \in \mathbf{R}^n \times \mathbf{R}^q \times \mathcal{M}_{n,q}(\mathbf{R}) \quad | \quad rg(A) = q\}$.

- **Isoymplectomorphism:** an isosymplectomorphism between two symplectic manifolds (V,ω_V) and (W,ω_W) is a function $f : V \to W$ verifying $f^*\omega_W = \omega_V$. This equation involves the first order derivatives of f so it defines a subset of $J^1(V,W)$. We note the differential relation defined this way $\mathcal{R}_{isosymp}(V,W)$.

  With $V = W = \mathbf{R}^{2n}$ endowed with the canonical symplectic structure, a function f is a symplectomorphism if and only if ${}^t\partial_p f_q \partial_p f_p$ and ${}^t\partial_q f_q \partial_q f_p$ are symmetric and ${}^t\partial_q f_q \partial_p f_p - {}^t\partial_q f_p \partial_p f_q = I_k$. Therefore, $\mathcal{R}_{isosymp}(\mathbf{R}^{2n},\mathbf{R}^{2n})$ is exactly

  $$\left\{(v,w,A) \in \mathbf{R}^{2n} \times \mathbf{R}^{2n} \times \mathbf{R}^{4n^2} \mid {}^tA_3A_1, {}^tA_4A_2 \in S_n(\mathbf{R}) \ \wedge \ {}^tA_4A_1 - {}^tA_2A_3 = I_n\right\},$$

  where $A = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}$ with $A_1, A_2, A_3, A_4 \in \mathcal{M}_{n,n}(\mathbf{R})$.

- **Symplectomorphisms:** a symplectomorphism from V to a symplectic manifold (W,ω_W) is a map f such that $f^*\omega_W$ defines a symplectic form on V. Since ω_W is a symplectic form, it is closed and we have $d(f^*\omega_W) = f^*d\omega_W = f^*0 = 0$. The condition on f is then reduced to the fact that $f^*\omega_W$ is non-degenerate.

  In $V = W = \mathbf{R}^{2n}$ endowed with the canonical symplectic structure is symplectic if and only if ${}^t\nabla f(v)\mathbf{J}_{2n}\nabla f(v)$ is non-degenerate for all $v \in V$ so $\mathcal{R}_{symp}(\mathbf{R}^{2n},\mathbf{R}^{2n})$ is exactly

  $$\left\{(v,w,A) \in \mathbf{R}^{2n} \times \mathbf{R}^{2n} \times \mathbf{R}^{4n^2} \;\middle|\; \det \begin{pmatrix} {}^tA_3A_1 - {}^tA_1A_3 & {}^tA_3A_2 - {}^tA_1A_4 \\ {}^tA_4A_1 - {}^tA_2A_3 & {}^tA_4A_2 - {}^tA_2A_4 \end{pmatrix} \neq 0\right\}.$$

- More generally, any differential equation of order r induce a differential relation of order r.

Together with this definition comes the notion of *open* and *close* relations, which correspond to open and close subsets of the jet space. Immersion and submersions relations are open since they are defined as the complement of the closed set composed of morphisms with at least a minor equal to zero. The relation associated with isosymplectomorphisms is closed since it is defined with an equality. On the contrary, the relation associated to symplectomorphisms is open as it is the complement of a closed subset. As usual, relations defined with equalities or large inequalities are closed while relations defined as complement of singularities or with strict inequalities are open.

Of course, smooth solutions of a differential equation, or inequality, of order r are such that their r-jet extension sends V in the induced differential relation. We now extend the notion of solution to all sections of the jet space.

Definition 3.11 (Formal and genuine solutions [9])**.** *A* formal solution *of a given differential relation $\mathcal{R}$ of order r is a section of the r-jet space which takes its values in $\mathcal{R}$, that is $F : V \to \mathcal{R} \subset J^r(V, W)$. We denote by Sec $\mathcal{R}$ the subset of sections of the r-jet space composed of formal solutions of $\mathcal{R}$.*

A genuine solution *of $\mathcal{R}$ is a section $f : V \to V \times W$ whose r-jet extension is a formal solution. We denote by Hol $\mathcal{R}$ the subset of Sec $\mathcal{R}$ composed of holonomic formal solutions of $\mathcal{R}$.*

Example 3.12. For $V = \mathbf{R}^n$ and $W = \mathbf{R}^q$ with $n = q = 2l$ and endow these two spaces with the canonical symplectic structure. Consider the relation $\mathcal{R}_{isosymp}$ defined in the previous example and take any section $f : V \to V \times W$. The 1-jet section $F : v \mapsto (v, f(v), id) \in \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R}^{n^2}$ is a formal solution of $\mathcal{R}_{isosymp}$. Since it is not holonomic, $f = bsF$ is *a priori* not a genuine solution.

When we study differential equations, we look for genuine solutions. In some cases, it can be useful to first see if there is a formal solution to the considered problem. If this is not the case, it is useless to search for genuine solutions. In the following section, we introduce the homotopy-principle, which, if it holds, ensures the existence of genuine solution from the existence of formal solutions.

3.2.3 Homotopy-principle

Definition 3.13 (Homotopy-principle [9])**.** *A differential relation $\mathcal{R}$ satisfies the homotopy-principle (or h-principle) if all formal solutions of $\mathcal{R}$ are homotopic in $\mathcal{R}$ to a holonomic formal solution. In particular, this ensures the existence of a genuine solution.*

In other words, the h-principle holds for a relation $\mathcal{R}$ if any formal solution of the relation can be deformed in Sec $\mathcal{R}$ to the jet extension of a section $f : V \to V \times W$, which is therefore a genuine solution of $\mathcal{R}$.

There are different variations of this principle. Below is a list of some of them.

- **One parameter h-principle:** a differential relation $\mathcal{R}$ satisfies the one parameter h-principle if all homotopy F_t in Sec $\mathcal{R}$ joining two holonomic sections can be smoothly deformed in a homotopy in Hol $\mathcal{R}$ keeping F_0 and F_1 fixed.

- **Multi-parameter h-principle:** a differential relation $\mathcal{R}$ satisfies the multi-parameter h-principle if all smooth family $F_T \subset \text{Sec } \mathcal{R}$ such that $F_T \in \text{Hol}\mathcal{R}$ for $T \in \partial I^k$ can be smoothly deformed in a family in Hol $\mathcal{R}$ keeping F_T fixed for all $T \in \partial I^k$. Here we have noted $I^k = [0,1]^k$.

- **Local h-principle:** let $A \subset V$ and $Op_V(A)$ an open neighbourhood of A in V. A differential relation $\mathcal{R}$ satisfies the local h-principle around A if all formal solutions of $\mathcal{R}$ defined above $Op_V(A)$ are homotopic in the sections of $\mathcal{R}$ defined above a neighbourhood of A to a holonomic formal solution. In other words, the deformation should not change too much the space of definition of the original section. Note however that the holonomic sections we obtain are only defined in the neighbourhood of A.

- **Relative h-principle:** let $B \subset V$ and $Op_V(B)$ an open neighbourhood of B in V. A differential relation $\mathcal{R}$ satisfies the relative h-principle relative to B if all formal solutions of $\mathcal{R}$ holonomic above $Op_V(B)$ in V are homotopic in the sections of $\mathcal{R}$ fixed on $Op_V(B)$ to a holonomic formal solution. In other words, we start from a formal solution which is already holonomic above an open set and we deforme it to obtain a holonomic solution on the whole V without changing the part which is already holonomic.

- **$\mathcal{C}^0$-dense h-principle:** a differential relation $\mathcal{R}$ satisfies the $\mathcal{C}^0$-dense h-principle if any formal solution F_0 of $\mathcal{R}$ is homotopic in $\mathcal{R}$ to a holonomic formal solution F_1 such that $\text{bs}F_0$ and $\text{bs}F_1$ are $\mathcal{C}^0$-close.

It is also possible to work with combinations of these versions such as the $\mathcal{C}^0$-close one parameter h-principle, where we ask that the deformation of the homotopy is $\mathcal{C}^0$-small, or the relative one parameter h-principle, where we ask that the deformation of the homotopy is fixed on B.

Proving that the h-principle holds for a given relation $\mathcal{R}$ can be sometimes difficult. In the following section, we present some tools that we can use to achieve it.

3.3 Proving the h-principle: tools and examples

3.3.1 Holonomic approximation

Below is the theorem on which are based all the results we will present in the following. The proof of this result can be found in [9].

Theorem 3.14 (Holonomic approximation [9])**.** *Let $A \subset V$ a polyhedron of positive codimension and $F : Op_V(A) \to J^r(V,W)$ a section of the jet space.*

For all $\delta, \epsilon > 0$, there exists a diffeotopy $(h^\tau)_{\tau\in[0,1]} : V \to V$ δ-small in the $\mathcal{C}^0$ sense and a holonomic section $\tilde{F} : Op_V(h^1(A)) \to J^r(V,W)$ such that $d(\tilde{F}(v), F(v)) < \epsilon$ for all $v \in Op_V(h^1(A))$.

This result also holds in its parametric and relative forms:

Theorem 3.15 (Parametric holonomic approximation [9])**.** *Let $A \in V$ a polyhedron of codimension > 0 and $F_z : Op_V(A) \to J^r(V,W)$ a family of sections parametrised by $z \in I^k := [0,1]^k$ with F_z holonomic for $z \in \partial I^k$.*

For all $\delta, \epsilon > 0$, there exists a family of diffeotopies $(h_z^\tau)_{\tau\in[0,1]} : V \to V$ δ-small in the $\mathcal{C}^0$ sense and a family of holonomic sections $\tilde{F}_z : Op_V(h_z^1(A)) \to J^r(V,W)$ such that $d(\tilde{F}_z(v), F_z(v)) < \epsilon$ for all $v \in Op_V(h_z^1(1))$ and all $z \in I^k$ and such that $h_z^\tau = id_V$ and $\tilde{F}_z = F_z$ for $z \in \partial I^k$.

Theorem 3.16 (Relative holonomic approximation [9])**.** *Let $A \in V$ a polyhedron of codimension > 0 and $F : Op_V(A) \to J^r(V,W)$ a section of the jet space holonomic in a neighbourhood $Op_V\partial A$.*

For all $\delta, \epsilon > 0$, there exists a δ-small diffeotopy $(h^\tau)_{\tau\in[0,1]} : V \to V$ fixed on $Op_V\partial A$ and a holonomic section $\tilde{F} : Op_V(h^1(A)) \to J^r(V,W)$ such that $d(\tilde{F}(v), F(v)) < \epsilon$ for all $v \in Op_V(h^1(A))$ and $\tilde{F}(v) = F(v)$ on $Op_V(h^1(\partial A)) = Op_V(\partial A)$.

The relative form of the theorem is particularly useful when one wants to prove the existence of a holonomic section on a whole space divided into pieces that are treated one after another. If it is possible to prove the existence on a piece and to build the section on neighbouring pieces while sticking to what has already being done, then we can prove the existence on the whole space.

3.3.2 Application to differential p-forms

Here is a first interesting result which can be proved using holonomic approximation.

Corollary 3.17 (Approximation of differential forms by closed forms [9])**.** *Let V be an open manifold, A a polyhedron of positive codimension, $a \in H^p(V)$ a cohomology class. Near A, we can approach in the $\mathcal{C}^0$ sense any p-form ω by a closed p-form $\tilde{\omega}$ in a. Moreover, given $\Omega \in a$ and a $(p-1)$-form α, we can choose $\tilde{\omega}$ of the form $d\tilde{\alpha} + \Omega$ for $\tilde{\alpha}$ $\mathcal{C}^0$-close to α.*

Proof. (from [9])

Consider the bundle $\mathbf{p} : \Lambda^{(p-1)}V \to V$ where $\Lambda^{(p-1)}V = \bigsqcup_{v\in V} \Lambda^{(p-1)}T_vV$ and

$$\mathbf{p}\left(\omega \in \Lambda^{(p-1)}T_vV\right) = v.$$

Sections of this bundle are differential $(p-1)$-forms on V.

Let $v \in V$ and $\phi = (x_1, ..., x_n) : U \subset V \to \mathbf{R}^n$ a coordinate chart in a neighbourhood U of v in V. The chart ϕ induces coordinates on T_vV and then on $\Lambda^{(p-1)}T_vV$. An element ω_v of $\Lambda^{(p-1)}T_vV$ can then be represented by an element

$$\left(\omega_v^1, ..., \omega_v^{\binom{n}{p-1}}\right) \in \mathbf{R}^{\binom{n}{p-1}}.$$

In the same coordinates, a smooth local section $\omega : U \to \Lambda^{(p-1)}V$ is then represented by an element

$$x \mapsto \left(x, \omega_1(x), ..., \omega_{\binom{n}{p-1}}(x)\right)$$

in $\{id\} \times (\mathcal{C}^\infty(\phi(U), \mathbf{R}))^{\binom{n}{p-1}}$ and can be written as

$$x \mapsto \sum_{1\leqslant j_1<...<j_{p-1}\leqslant n} \omega_i(x) dx_{j_1}(x) \wedge ... \wedge dx_{j_{p-1}}(x),$$

with $\omega_i : \phi(U) \to \mathbf{R}$ for all $1 \leqslant j_1 < ... < j_{p-1} \leqslant n$.

The chart ϕ also induces a coordinate chart on $\left(\Lambda^{(p-1)}V\right)^{(1)}$, whose elements are associated with elements

$$\left(x, a^1, ..., a^{\binom{n}{p-1}}, a_1^1, a_2^1, ..., a_n^1, ..., a_n^{\binom{n}{p-1}}\right)$$

of $\phi(U) \times \mathbf{R}^{\binom{n}{p-1}} \times \left(\mathbf{R}^{\binom{n}{p-1}}\right)^n$. Here, $\left(a^1, ..., a^{\binom{n}{p-1}}\right)$ represents a possible value of a p-form ω at $v = \phi^{-1}(x)$ and $(a_1^i, ..., a_n^i)$ represents possible values of the derivatives of the i-th coordinate map of ω at v, for all $i \in [\![1, \binom{n}{p-1}]\!]$.

The exterior differential of ω is a p-form whose expression on the coordinates induced by ϕ on $\Lambda^p V$ is

$$\begin{aligned}
&\sum_{1 \leqslant j_1 < ... < j_{p-1} \leqslant n} d\omega_i \wedge dx_{j_1} \wedge ... \wedge dx_{j_{p-1}} \\
&= \sum_{1 \leqslant j_1 < ... < j_{p-1} \leqslant n} \sum_{l=1}^{n} \frac{\partial \omega_i}{\partial x_l} dx_l \wedge dx_{j_1} \wedge ... \wedge dx_{j_{p-1}} \\
&= \sum_{1 \leqslant j_1 < ... < j_p \leqslant n} \left(\sum_{l=1}^{n} (-1)^{l+1} \frac{\partial \omega_{i(l;j_1,...,j_p)}}{\partial x_{j_l}} \right) dx_{j_1} \wedge ... \wedge dx_{j_p},
\end{aligned}$$

where $i(l; j_1, ..., j_p) = (j_1, ..., j_{l-1}, j_{l+1}, ..., j_p)$. This p-form is then represented in $\{id\} \times (\mathcal{C}^\infty(\phi(U), \mathbf{R}))^{\binom{n}{p}}$ by

$$x \mapsto \{x\} \times \left(\sum_{l=1}^{n} (-1)^{l+1} \frac{\partial \omega_{i(l;j_1,...,j_p)}}{\partial x_{j_l}}(x) \right)_{1 \leqslant j_1 < ... < j_p \leqslant n}.$$

Set $D : \phi(U) \times \mathbf{R}^{\binom{n}{p-1}} \times \left(\mathbf{R}^{\binom{n}{p-1}}\right)^n \to \phi(U) \times \mathbf{R}^{\binom{n}{p}}$ as the map

$$F = (x, a^1, ..., a^{\binom{n}{p-1}}, a_1^1, a_2^1, ..., a_n^1, ..., a_n^{\binom{n}{p-1}}) \mapsto \{x\} \times \left(\sum_{l=1}^{n} (-1)^{l+1} a_l^{i(l;j_1,...,j_p)} \right)_{1 \leqslant j_1 < ... < j_p \leqslant n} \tag{3.1}$$

and $\tilde{D}_U : \left(\Lambda^{(p-1)}V\right)^{(1)} \to \Lambda^p V$ its pullback in the space of differential forms of V defined above U:

$$\tilde{D}_U = \phi_p^{-1} \circ D \circ (\phi_{p-1}^1)^{-1},$$

where ϕ_p and ϕ_{p-1}^1 are the coordinates charts induced by ϕ on $\Lambda^p V$ and $\left(\Lambda^{(p-1)}V\right)^{(1)}$ above U. The definition of $\tilde{D}_U$ is invariant under a coordinate change: for $\omega \in \Lambda^{(p-1)}V$, we have $\tilde{D}_U \circ J_\omega^1 = d\omega$. The map $\tilde{D}_U$ is simply the lift of the derivation operator on $\Lambda^{(p-1)}V$ to $(\Lambda^{(p-1)}V)^{-1}$. This comes immediately from the definition of the exterior derivative (I give details about its construction in my M1 internship report).

Since $\tilde{D}_U$ can be built above any chart (ϕ, U) of V and is invariant under coordinate changes, we can extend it to the whole manifold. Let us denote $\tilde{D}$ this extension. Since D is continuous, $\tilde{D}$ is continuous too.

Note that the definition 3.1 of D does not involve the coordinates a_i. This means that for all $F \in \Lambda^{(p-1)}V$, $\tilde{D}(F)$ does not depend on $\mathbf{p}_0^1(F)$, that is on the values taken by the underlying form bs$F$, but only on the derivatives in F. Moreover, the linear map which sends the a_j^i to the coordinates of $D(F)$ in $\mathbf{R}^{\binom{n}{p}}$ is surjective: there is $n \times \binom{n}{p-1}$ unknown for $\binom{n}{p}$ independent equations. This implies that $\tilde{D}$ is also surjective.

Let ω be a p-form. It exists a section F_ω of $(\Lambda^{p-1}V)^{(1)}$ such that $\tilde{D} \circ F_\omega = \omega$. Since $\tilde{D}$ is independent of $\mathbf{p}(F_\omega)$, we can choose F_ω such that $\mathrm{bs}F_\omega = \alpha$ for any $(p-1)$-form α.

Let A be a polyhedron of positive codimension in V. We first complete the proof in the case $a = 0 \in H^p(V)$. Theorem 3.14 applied to F_ω near A gives that for all $\delta, \epsilon > 0$, there exists a diffeotopy $(h^\tau)_{\tau\in[0,1]} : V \to V$ δ-small in the $\mathcal{C}^0$ sense and a holonomic section $\tilde{F}_\omega = J^1_{\tilde{\alpha}} : Op_V(h^1(A)) \to J^r(V,W)$ such that $\tilde{F}_\omega(v)$ and $F_\omega(v)$ are ϵ-close for all $v \in Op_V(h^1(A))$. In particular, $\tilde{\alpha}$ is $\mathcal{C}^0$-close to α and, since $\tilde{D}$ is continuous, $\tilde{\omega} := \tilde{D} \circ \tilde{F}_\omega$ is $\mathcal{C}^0$-close to ω. We also have $d\tilde{\alpha} = \tilde{D}(J^1_{\tilde{\alpha}}) = \tilde{D}(\tilde{F}_\omega) = \tilde{\omega}$ so $\tilde{\omega}$ is exact.

Let now a be any cohomology class and $\Omega \in a$. Apply the previous argument to $\theta = \omega - \Omega$ and take $\tilde{\omega} = \tilde{\theta} + \Omega$. It is $\mathcal{C}^0$-close to ω and can be written as $d\tilde{\alpha} + \Omega$. This shows that we can approach any p-form by a closed form of any cohomology class on $Op_V(h^1(A))$. □

Note that the parametric version of this proposition is also true: we just have to apply Theorem 3.15 instead of Theorem 3.14.

Corollary 3.18 (Parametric approximation of differential forms by closed forms [9])**.** *Let V be an open manifold, A a polyhedron of positive codimension, $a \in H^p(V)$ a cohomology class. Let $\omega_t : Op_V(A) \to \Lambda^p V$ be a homotopy of differential p-forms parametrized by $t \in [0,1]$ with ω_0 and ω_1 in a. Near A, we can approach in the $\mathcal{C}^0$ sense ω_t by a homotopy of closed p-form $\tilde{\omega}_t$ in a such that $\tilde{\omega}_0 = \omega_0$ and $\tilde{\omega}_1 = \omega_1$. Moreover, given $\Omega \in a$ and a homotopy of $(p-1)$-form α_t, we can choose $\tilde{\omega}_t$ of the form $d\tilde{\alpha}_t + \Omega$ for $\tilde{\alpha}_t$ $\mathcal{C}^0$-close to α_t.*

Proof. (from [9])

Take first $a = 0 \in H^p(V)$. Let $\tilde{D} : \left(\Lambda^{(p-1)}V\right)^{(1)} \to \Lambda^p V$ such as in the proof of Corollary 3.17. It exists a family of sections F^t_ω of $(\Lambda^{p-1}V)^{(1)}$ such that $\tilde{D} \circ F^t_\omega = \omega_t$. Since $\tilde{D}$ is independent of $\mathbf{p}(F^t_\omega)$, we can choose F^t_ω such that $\mathrm{bs}F^t_\omega = \alpha_t$.

Theorem 3.15 applied to F^t_ω near A gives that for all $\delta, \epsilon > 0$, there exists a family of diffeotopy $(h^\tau_t)_{\tau,t\in[0,1]} : V \to V$ δ-small in the $\mathcal{C}^0$ sense and a homotopy of holonomic section $\tilde{F}^t_\omega = J^1_{\tilde{\alpha}^t} : Op_V(h^1(A)) \to J^r(V,W)$ such that $\tilde{F}^t_\omega(v)$ and $F^t_\omega(v)$ are ϵ-close for all $v \in Op_V(h^1(A))$ and all $t \in [0,1]$, $h^\tau_0 = h^\tau_1 = id_V$, $\tilde{F}^0_\omega = F^0_\omega$ and $\tilde{F}^1_\omega = F^1_\omega$. In particular, $\tilde{\alpha}^t$ is $\mathcal{C}^0$-close to α^t for all $t \in [0,1]$ and, since $\tilde{D}$ is continuous, $\tilde{\omega}^t := \tilde{D} \circ \tilde{F}^t_\omega$ is $\mathcal{C}^0$-close to ω^t. We also have $d\tilde{\alpha}^t = \tilde{D}(J^1_{\tilde{\alpha}^t}) = \tilde{D}(\tilde{F}^t_\omega) = \tilde{\omega}^t$ so $\tilde{\omega}^t$ is exact. Since $\tilde{F}^0_\omega = F^0_\omega$ and $\tilde{F}^1_\omega = F^1_\omega$, $\tilde{\omega}_0 = \omega_0$ and $\tilde{\omega}_1 = \omega_1$.

Let now a be any cohomology class and $\Omega \in a$. Apply the previous argument to $\theta^t = \omega^t - \Omega$ and take $\tilde{\omega}^t = \tilde{\theta}^t + \Omega$. It is $\mathcal{C}^0$-close to ω^t for all $t \in [0,1]$ and can be written as $d\tilde{\alpha}^t + \Omega$. This shows that we can approach any homotopy of p-form by a homotopy of closed form of any cohomology class on $Op_V(h^1(A))$. □

3.3.3 Open Diff_V-invariant relations

We are now interested in a special category of differential relation, that we call *Diff_V-invariant*.

Denote by $\mathrm{Diff}_V X$ the group of diffeomorphisms of X which preserve the fibres, that is $h_X : X \to X$ such that there exists $h_V : V \to V$ satisfying $p \circ h_X = h_V \circ p$. If such a h_V exists, then it is obviously unique: for h^1_V and h^2_V satisfying the previous equation, we have $h^1_V \circ p = h^2_V \circ p$ in X. Since p is surjective, this means that $h^1_V = h^2_V$. Note that all of this can be extended to any fibre bundle $p : E \to F$.

Definition 3.19 (Natural bundle [9])**.** *Let $p : E \to F$ a differential bundle and $\pi : Diff_F E \to Diff F$ the homomorphism which associate a diffeomorphism $h_E \in Diff_F E$ to the unique diffeomorphism $h_F \in Diff F$ such that $p \circ h_E = h_F \circ p$. If π can be inverted, that is if there exists $j : Diff F \to Diff_F E$ such that $\pi \circ j = id_{Diff F}$, then we say that $p : E \to F$ is a* natural bundle*.*

Note that j is not necessarily unique.

Example 3.20. The bundles we are working with in this chapter are natural:

- the trivial bundle $p : V \times W \to V$ with $j : h \mapsto (h, id_W)$,
- the tangent bundle $p : TV \to V$ with $j : h \mapsto dh$,
- the exterior bundle $p : \Lambda^p V \to V$ with $j : h \mapsto d^p h$, where $h^* : (v, \omega) \in V \times \Lambda^p_v V \mapsto (h(v), \omega_h : (a_1, ..., a_p) \mapsto \omega(d_v h^{-1}(a_1), ..., d_v h^{-1}(a_p)))$.

The fact of being a natural bundle can be extended to the jet bundle. More precisely, for $p : X \to V$ a natural fibre bundle, we have that $p^r : X^{(r)} \to V$ is natural.

Proposition 3.21. *Let $p : X \to V$ be a natural bundle and $j : Diff V \to Diff_V X$ be an application verifying $p \circ j(h) = h \circ p$ for all $h \in Diff V$ as in Definition 3.19.*

For all $h \in Diff V$,

$$h^r : s \in X^{(r)} \to J^r_{j(h)\circ\bar{s}\circ h^{-1}}(h \circ p^r(s)),$$

where $\bar{s}$ is a local section of $p : X \to V$ whose r-jet coincides with s at $p^r(s)$, is a diffeomorphism of $X^{(r)}$.

The map $j^r : h \mapsto h^r$ is a homomorphism.

Proof. As in the proof of Proposition 3.7, the definition is independent of the choice of $\bar{s}$ since it only involves its value and the values of its derivatives of order at most r at v, which are fixed by s. Note also that we are allowed to take the r-jet extension of $j(h) \circ \bar{s} \circ h^{-1}$ because it is a section of $p : X \to X$:

$$p \circ (j(h) \circ \bar{s} \circ h^{-1}) = h \circ p \circ \bar{s} \circ h^{-1} = h \circ h^{-1} = id_V.$$

For all $s \in X^{(r)}$, we have

$$(p^r \circ h^r)(s) = p^r(J^r_{j(h)\circ\bar{s}\circ h^{-1}}(h(p(s)))) = (p \circ j(h) \circ \bar{s} \circ h^{-1})(h(p^r(s))) = (h \circ p)(s).$$

This is true for all $h \in \mathrm{Diff}\, V$ so the map $j^r : \mathrm{Diff}\, V \to \mathrm{Diff}_V X$ defined as $j^r(h) = h^r$ for all $h \in \mathrm{Diff}\, V$ verifies

$$p^r \circ j^r(h) = h \circ p^r.$$

It is a homomorphism since for all $s \in X^{(r)}$

$$(j^r(h) \circ j^r(g))(s) = J^r_{j(h)\circ\overline{j^r(g)(s)}\circ h^{-1}}(h \circ p^r \circ j^r(g)(s)),$$

where the section $\overline{j^r(g)(s)}$ has a r-jet which coincides with $J^r_{j(g)\circ\bar{s}\circ g^{-1}}(g \circ p^r(s))$ at

$$p^r(j^r(g)(s)) = g(p^r(s)).$$

We can take $j(g) \circ \bar{s} \circ g^{-1}$ and we obtain

$$\begin{aligned}(j^r(h) \circ j^r(g))(s) &= J^r_{j(h)\circ j(g)\circ\bar{s}\circ g^{-1}\circ h^{-1}}((h \circ g \circ p^r)(s)) = J^r_{j(h\circ g)\circ\bar{s}\circ(h\circ g)^{-1}}((h \circ g)(p^r(s))) \\ &= j^r(h \circ g)(s).\end{aligned}$$

Therefore, j^r satisfies the definition 3.19 so the bundle $p^r : X^{(r)} \to V$ is natural. □

In the following, h_* will denote the action of the diffeomorphism $h : V \to V$ on the sections of $p : X^{(r)} \to V$. More precisely, $h_*F = j^r(h) \circ F \circ h^{-1}$ for all $F \in \mathrm{Sec}X^{(r)}$. If F is a section above $U \subset V$, then h_*F is a section above $h(U)$. To simplify notations, we denote h^* the action of h^{-1} on $\mathrm{Sec}X^{(r)}$, that is $h^*F := (h^{-1})_*F$ for all $F \in \mathrm{Sec}X^{(r)}$.

Note that h_* preserves holonomy for all $h \in \mathrm{Diff}\, V$. In fact, given a section $f : V \to X$ and a point $v \in V$, we have

$$h_* J^r_f(v) = J^r_{j(h)\circ\overline{J^r_f(h^{-1}(v))}\circ h^{-1}}((h \circ p^r \circ J^r_f \circ h^{-1})(v)),$$

where $\overline{J^r_f(h^{-1}(v))}$ represents a local section in the neighbourhood of $\tilde{v} := p^r \circ J^r_f \circ h^{-1}(v) = h^{-1}(v)$ whose r-jet extension at $\tilde{v}$ coincides with $J^r_f(h^{-1}(v))$. We can choose f, which gives

$$h_* J^r_f(v) = J^r_{j(h)\circ f\circ h^{-1}}(v).$$

Thus, $h_* J^r_f$ is the r-jet extension of $j(h) \circ f \circ h^{-1}$ so is holonomic.

Definition 3.22 (Diff_V-invariant differential relation [9])**.** *A differential relation is said to be* Diff_V-invariant *if it is invariant under the action of h_* for all $h \in Diff_V$, that is $h_*F \in Sec\mathcal{R}$ for all $F \in Sec\mathcal{R}$.*

In other terms, a differential relation is Diff_V-invariant if it is invariant under reparametrisation of V. A lot of the relations that we will use in the following have this property.

Example 3.23.

- The relations $\mathcal{R}_{imm}(V, W)$ and $\mathcal{R}_{subm}(V, W)$ are Diff_V-invariant. More generally, any relation in $J^1(V, W)$ which imposes a condition on the rank of the differential is Diff_V-invariant. In fact, for a diffeomorphism $h : V \to V$,

  $$h_*(x, y, A) = J^1_{(h\times id)(x,\bar{y},A)}(h(x)) = (h(x), y, A \circ d_x h).$$

  Since h is a diffeomorphism, its differential at any point is invertible, so its composition with the homomorphism A has the same rank as A.

- The same formula shows that all differential relation which would impose conditions on the image of the differential is also Diff_V-invariant. This will be useful when we will talk about Grassmanian bundles.

The notion of Diff_V-invariance is particularly interesting thanks to the following theorem.

Theorem 3.24 (Local h-principle for open Diff_v-invariant relations [9])**.** *Let V, W be two manifolds and $\mathcal{R} \subset J^1(V, W)$ an open $Diff_V$-invariant differential relation. All the local forms of the h-principle holds for $\mathcal{R}$ near any polyhedron of positive codimension.*

Proof. (from [9]) Let us first suppose that F_t lies in a single coordinate chart.

We first prove the theorem in the 1-parameter case. Let F_0 and F_1 two holonomic solutions of $\mathcal{R}$ and F_t a homotopy between them in $\mathcal{R}$.

Let $A \subset V$ be a polyhedron of positive codimension. For all $\delta, \epsilon > 0$, Theorem 3.15 ensures the existence of a family of δ-small diffeotopies $h^\tau_t : V \to V$ and a family of holonomic sections $\tilde{F}_t : Op_V(h^1_t(A)) \to J^r(V, W)$ such that $\tilde{F}_t$ and F_t are $\mathcal{C}^0$-close for all $t \in [0, 1]$, $\tilde{F}_0 = F_0$, $\tilde{F}_1 = F_1$ on $Op_V(h^1_t(A))$ and $h^\tau_0 = h^\tau_1 = h^0_t = id_V$ for all $\tau, t \in [0, 1]$.

Since F_t is a homotopy in the open set $\mathcal{R}$, we can choose ϵ such that $\tilde{F}_t$, which is ϵ-close to F_t, is also in $\mathcal{R}$ for all $t \in [0,1]$ on $U := Op_V(h_t^1(A))$ and the linear homotopy $\hat{F}_t^\nu = \nu\tilde{F}_t + (1-\nu)F_t$ as well. Here, the addition and the scalar multiplication are the one induced by a coordinate chart in the neighbourhood of F_t. Let $\tilde{U} := (h^1)^{-1}(U)$ and $(G_t^\eta)_{\eta\in[0,1]}$ be the family of homotopies such that

$$G_t^\eta = \begin{cases} (h_t^{2\eta})^*\hat{F}_t^0 = (h_t^{2\eta})^*\left(F_t|_{h_t^{2\eta}(\tilde{U})}\right), & \eta \in [0,1/2] \\ (h_t^1)^*\hat{F}_t^{2\eta-1}, & \eta \in [1/2,1]. \end{cases}$$

It goes from $(h_t^0)^*\left(F_t|_{h_t^0(\tilde{U})}\right) = F_t|_{\tilde{U}}$ to $(h_t^1)^*\hat{F}_t^1 = (h_t^1)^*\tilde{F}_t$, which is holonomic for all $t \in [0,1]$ for the action of $\mathrm{Diff}V$ on $\mathrm{Sec}X^{(r)}$ preserves holonomy. The fact that $h_0^\tau = h_1^\tau = id_V$ for all τ and that $\hat{F}_t^\nu$ remains constant at $t = 0,1$ for all $\nu \in [0,1]$ implies that G_t^η is respectively fixed at F_0 and F_1 when $t = 0$ and $t = 1$ for all $\eta \in [0,1]$. Since $\mathcal{R}$ is Diff_V-invariant, G_t^η is in $\mathrm{Sec}\mathcal{R}$ for all $t, \eta \in [0,1]$.

Moreover, since h_t^τ can be chosen arbitrarily small for all $\tau \in [0,1]$ and $\tilde{F}_t$ arbitrarily close to F_t, we have that $\mathrm{bs}G_t^\eta$ is arbitrarily $\mathcal{C}^0$-close to $\mathrm{bs}F_t$. Note that this $\mathcal{C}^0$-closeness does not hold for the jets since the composition with h_t^τ can drastically change the derivatives.

Therefore, we have established the local $\mathcal{C}^0$-dense 1-parameter h-principle. Note that the above argument also works when t is multivalued, one just has to change the notations. Applying Theorem 3.14 instead of Theorem 3.15 and skipping mentions of t, we also get the result for the simple local h-principle. In the same way, all what we have done is still true in the relative case, after the application of Theorem 3.16 instead of Theorem 3.15: if the diffeotopies h_t^τ fixes $Op(\partial A)$ and the homotopy $\tilde{F}_t$ coincides with F_t on $Op(\partial A)$, then G_t^1 also coincides with F_t on $Op(\partial A)$.

The theorem is then true for all homotopy F_t lying in a single coordinate chart. Now, if F_t does not lie in a single coordinate chart, then we can decompose the compact set $[0,1]$ into a finite number k of subintervals $[t_i, t_{i+1}]$ such that for all $i \in [\![0,k]\!]$, F_t lies in a single coordinate chart (ϕ_i, Ω_i) for all $t \in [t_i, t_{i+1}]$. Then, for all $i \in [\![1,k-1]\!]$, apply the non-parametric version of the theorem to $F_i \in \mathrm{Sec}\mathcal{R}$, and obtain a homotopy $(\check{F}_t^i)_{[0,1]}$ in $\mathcal{R}$ between F_i and a holonomic section $\check{F}^i$ in $\mathcal{R}$ above a neighbourhood U_i of A in V. Set the homotopies $\check{F}_t^0$ and $\check{F}_t^k$ respectively constant to F_0 and F_1 on $[0,1]$. Set also $\check{F}^0 = F_0$ and $\check{F}^k = F_1$. For all $i \in [\![0,k-1]\!]$, the homotopy

$$\hat{F}_t^i := \begin{cases} \check{F}_{1-3t}^i, & t \in [0,1/3] \\ F_{3t(t_{i+1}-t_i)+(2t_i-t_{i+1})}, & t \in [1/3,2/3] \\ \check{F}_{3t-2}^{i+1}, & t \in [2/3,1] \end{cases}$$

links $\check{F}_t^i$ and $\check{F}_t^{i+1}$ in $\mathrm{Sec}\mathcal{R}$ above the neighbourhood $U := \bigcap_{i=1}^k U_i$ of A. We can choose $\check{F}_t^i$ such that $\mathrm{bs}\check{F}_t^i$ are as $\mathcal{C}^0$-close to $\mathrm{bs}F_i$ as we want, and then lies in the same chart. If we consider coordinate charts induced on the jet spaces by an atlas of $V \times W$ as in Proposition 3.7, Ω_i is of the form $(p_0^r)^{-1}(B_i)$ for B_i an open subset of $V \times W$. Since $F_i \in \Omega_i \cap \Omega_{i+1}$, we can suppose that $\check{F}_t^i$ lies in Ω_i and Ω_{i+1}. Then, $\hat{F}_t^i$ lies in Ω_i, and we can apply the parametric version of the theorem. We obtain a smooth family $(G_t^{i,\alpha})_{\alpha,t\in[0,1]}$ of sections in $\mathcal{R}$ above U such that $G_t^{i,0} = \hat{F}_t^i$, $G_t^{i,1} \in \mathrm{Hol}\mathcal{R}$, $G_0^{i,\alpha} = F_i$ and $G_1^{i,\alpha} = F_{i+1}$ for all $t, \alpha \in [0,1]$ (see Figure 3.1). Then, the family of sections $(\hat{G}_t^\alpha)_{t,\alpha\in[0,1]}$ defined above U as $\hat{G}_t^\alpha = G^{i,\alpha}_{\frac{t-t_i}{t_{i+1}-t_i}}$ on $[t_i, t_{i+1}]$ for all $i \in [\![1,k-1]\!]$ verifies $\hat{G}_t^0 = F_t$, $\hat{G}_t^1 \in \mathrm{Hol}\mathcal{R}$, $\hat{G}_0^\alpha = F_0$ and $\hat{G}_1^\alpha = F_1$ for all $t, \alpha \in [0,1]$. As previously, the extensions to the $\mathcal{C}^0$, relative and multi-parameter cases are immediate. The theorem is then true in the general case. □

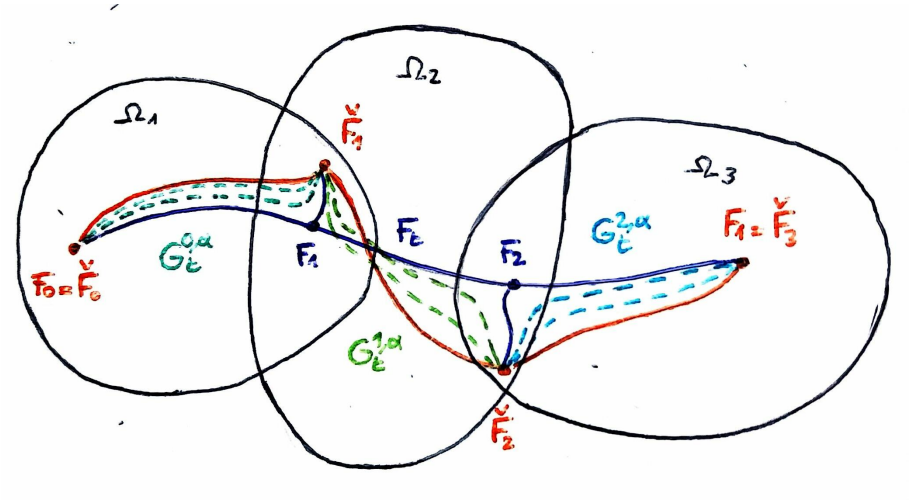


Figure 3.1: Schematisation in Sec$\mathcal{R}$ of the construction of the holonomoic homotopy when F_0 and F_1 does not lie in a same coordinate chart as in proof of Theorem 3.24 with $k = 3$. In orange are represented holonomic sections.

We now extend this result to the global h-principle. For that, we compress the whole space V to the open set on which are defined the objects we are interested in after application of the previous theorem. To make that this compression is possible, we have to assume that V is an open manifold.

Theorem 3.25 (Global h-principle for open Diff_v-invariant relations [9])**.** *Let V be an open manifold and $\mathcal{R} \subset J^1(V,W)$ an open $Diff_V$-invariant differential relation. All the global forms of the h-principle holds for $\mathcal{R}$, except the $\mathcal{C}^0$-dense and the relative one.*

Nevertheless, if V can be retracted into a polyhedron of positive codimension K, then the $\mathcal{C}^0$-closeness is still true in a neighbourhood of K and the relative version of the h-principle holds wih respect to K.

Proof. (from [9])

As V is open, there exists a polyhedron K of positive codimension such that V can be retracted in an arbitrarily small neighbourhood $Op_V(K)$ via a diffeotopy h^τ such that $h^0 = id_V$, $h^1(V) \subset Op_V(K)$ and $h^\tau = id_V$ on K. The polyhedron K is then called a *core* of V. For a proof of this result, see for instance [9].

We start by the 1-parameter version of the h-principle. Let F_0 and F_1 two holonomic section of $\mathcal{R}$ and F_t a homotopy joining them in $\mathcal{R}$. From Theorem 3.24, there exists a family of homotopy $\tilde{F}_t^\tau : Op_V(K) \to \mathcal{R}$ such that $\tilde{F}_t^1$ is holonomic, $\tilde{F}_t^0 = F_t$, $\tilde{F}_0^\tau = F_0$ and $\tilde{F}_1^\tau = F_1$ for all $\tau, t \in [0,1]$ on $Op_V(K)$. Moreover, we can choose it in such a way that $\tilde{F}_t^\tau$ is ϵ-close to F_t for all $\tau, t \in [0,1]$.

Define the family of homotopies G_t^τ such that

$$G_t^\tau = \begin{cases} (h^{6t\tau})^* F_0, & \tau \in [0,1/2], \quad t \in [0,1/3] \\ (h^{2\tau})^* F_{3t-1}, & \tau \in [0,1/2], \quad t \in [1/3,2/3] \\ (h^{6(1-t)\tau})^* F_1, & \tau \in [0,1/2], \quad t \in [2/3,1] \\ (h^{3t})^* F_0, & \tau \in [1/2,1], \quad t \in [0,1/3] \\ (h^1)^* \tilde{F}_{3t-2}^{2\tau-1}, & \tau \in [1/2,1], \quad t \in [1/3,2/3] \\ (h^{3(1-t)})^* F_1, & \tau \in [1/2,1], \quad t \in [2/3,1] \end{cases}$$

(see Figure 3.2). It is defined on V and since $\mathcal{R}$ is Diff_V-invariant, it takes its values in $\mathcal{R}$. Thanks to the fact that $(h^\tau)^*$ preserves holonomy, G_t^1 is holonomic for all $t \in [0,1]$. Finally, we also have that $G_0^\tau = F_0$ and $G_1^\tau = F_1$ for all $\tau \in [0,1]$.

As in Theorem 3.24, changing $t \in [0,1]$ for a multivalued parameter only changes notations. For the simple h-principle, one just has to use the simple version of Theorem 3.24 and forget the index$t$ in the passage from local to global.

For the relative version, if B is a core of V, then the retraction h^τ can be chosen fixed on $Op_V(B)$. Since the use of the relative version of Theorem 3.24 gives homotopies fixed on $Op_V(B)$, the resulting homotopies are fixed on $Op_V(B)$ too.

The decompression of $Op_V(K)$ into V makes that the $\mathcal{C}^0$-closeness does not hold any more in the global case. However, since this decompression is fixed on K, we still have that G_t^1 is $\mathcal{C}^0$-close to F_t in a small neighbourhood of K. □

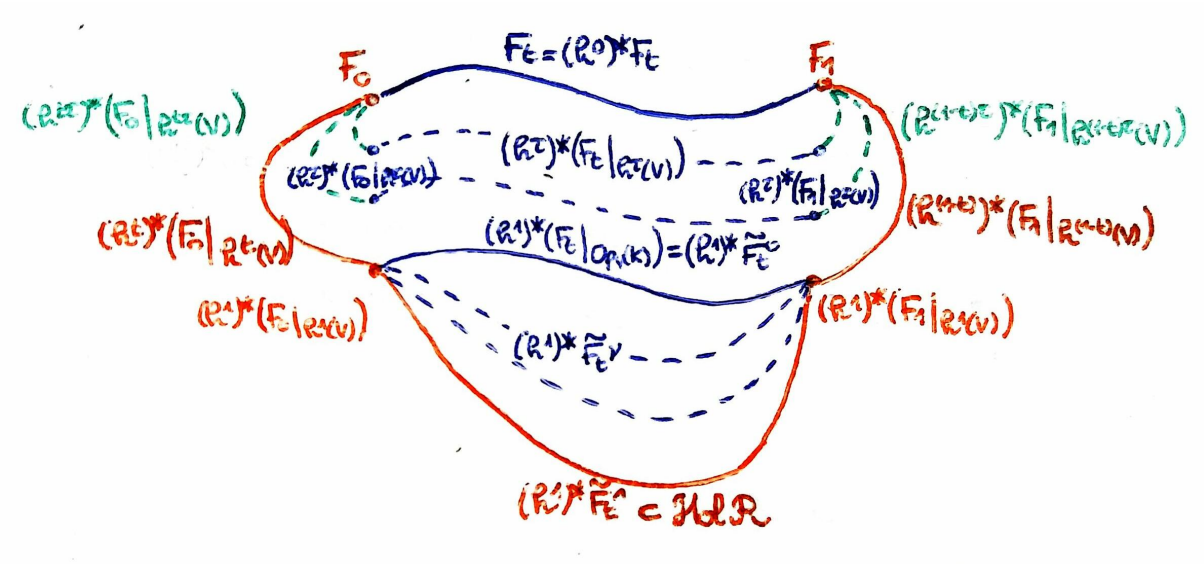


Figure 3.2: Schematisation in Sec$\mathcal{R}$ of the construction of the holonomoic homotopy G_t^τ in proof of Theorem 3.25. In orange and green are represented holonomic sections and the orange one represents $G_t 0$.

Here is a basic example where having proved the h-principle gives the existence of homotopies which are really difficult to visualize.

Example 3.26. Let V be the annulus $\{(x,y) \in \mathbf{R}^2 \mid \epsilon < x^2+y^2 < a\}$ and $W = \mathbf{R}$. The 1-jet space is $J^1(V,W) = V \times \mathbf{R} \times \mathbf{R}^2$. Let $f_0 : (x,y) \mapsto x^2+y^2$ and $f_1 = -f_0$. The two functions

f_0 and f_1 are homotopic in the space of immersions. Looking at the diagram be low, this result seems quite counter-intuitive and it is difficult to imagine how to deform the graph of f_0 to obtain thr graph of f_1. Yet, we can establish that this is possible with the h-principle tools we have presented until now.

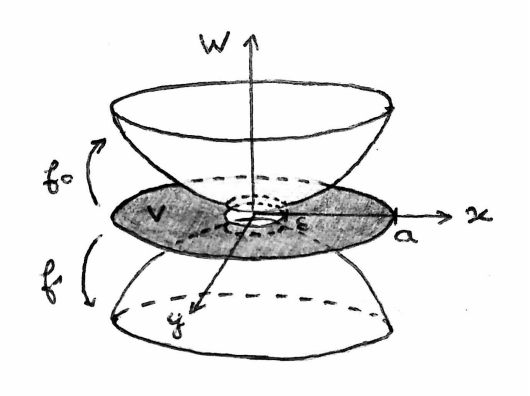


Figure 3.3: Graphs of $f_0 : (x, y) \mapsto x^2 + y^2$ and $f_1 : (x, y) \mapsto -x^2 - y^2$ above the annulus $V = \{(x, y) \in \mathbf{R}^2 \mid \epsilon < x^2 + y^2 < a\}$. Using h-principle tools, we can show that we can deform f_0 to f_1 in the set of immersions.

In fact, since V is open and the relation $\mathcal{R}_{imm}(V, \mathbf{R})$ is open and Diff_V-invariant, Theorem 3.25 implies that it is sufficient to find a formal solution of $\mathcal{R}_{imm}(V, \mathbf{R})$ linking f_0 and f_1. Identifying $\mathbf{R}^2$ with $\mathbf{C}$, we can take

$$F_t : (x, y) \in V \mapsto (x, y, tf_1(x, y) + (1 - t)f_0(x, y), e^{i\pi t}\nabla f_0) \in J^1(V, \mathbf{R}).$$

Using the same proof as for Theorem 3.24 followed by Theorem 3.25 with the invocation of Corollary 3.17 instead of Theorem 3.14, we obtain:

Proposition 3.27 ([9])**.** *Let V be an open manifold, $a \in H^p(V)$ a cohomology class and $\mathcal{R} \subset \Lambda^p V$ an open Diff_V-invariant differential relation.*

Any p-form $\omega : V \to \mathcal{R}$ is homotopic in $\mathcal{R}$ to a closed p-form in a.

Any homotopy of p-forms $\omega_t : V \to \mathcal{R}$ between two closed forms ω_0 and ω_1 in a can be deformed in $\mathcal{R}$ to a homotopy of closed forms in a between ω_0 and ω_1 without changing the ends ω_0 and ω_1.

Proof. (from [9])

The proof is the same as the one of the Theorem 3.24 followed by the one of Theorem 3.25, except that we replace the invocation of the holonomic approximation theorem by Corollary 3.17 and that we also use the fact that the cohomology class of $h^\tau\omega$ is constant for any homotopy h^τ and any p-form ω. We do not need to separate $[0, 1]$ into subintervals to consider homotopies in the same chart since the addition and the scalar multiplication are well-defined on $\Lambda^p V$.

We first prove the theorem in the 1-parameter case. Let ω_0 and ω_1 two p-forms in a with values in $\mathcal{R}$. Let ω_t be a homotopy in $\mathcal{R}$ which links ω_0 and ω_1. Let $A \subset V$ be a polyhedron of positive codimension. For all $\delta, \epsilon > 0$, Corollary 3.18 ensures the existence of a family of δ-small diffeotopies $h_t^\tau : V \to V$ and a family of p-forms $\tilde{\omega}_t : Op_V(h_t^1(A)) \to \Lambda^p V$ such that $\tilde{\omega}_t$ and ω_t are $\mathcal{C}^0$-close for all $t \in [0, 1]$, $\tilde{\omega}_0 = \omega_0$, $\tilde{\omega}_1 = \omega_1$ on $Op_V(h_t^1(A))$ and $h_0^\tau = h_1^\tau = h_t^0 = id_V$ for all $\tau, t \in [0, 1]$.

Since ω_t is a homotopy in the open set $\mathcal{R}$, we can choose ϵ such that $\tilde{\omega}_t$, which is ϵ-close to ω_t, is also in $\mathcal{R}$ for all $t \in [0,1]$ on $U := Op_V(h_t^1(A))$ and the linear homotopy $\hat{\omega}_t^\nu = \nu\tilde{\omega}_t + (1-\nu)\omega_t$ as well. Let $\tilde{U} := (h^1)^{-1}(U)$ and $(G_t^\eta)_{\eta\in[0,1]}$ be the family of homotopies such that

$$G_t^\eta = \begin{cases} (h_t^{2\eta})^*\hat{\omega}_t^0 = (h_t^{2\eta})^*\left(\omega_t\big|_{h_t^{2\eta}(\tilde{U})}\right), & \eta \in [0,1/2] \\ (h_t^1)^*\hat{\omega}_t^{2\eta-1}, & \eta \in [1/2,1]. \end{cases}$$

It goes from $(h_t^0)^*\left(\omega_t\big|_{h^0(\tilde{U})}\right) = \omega_t\big|_{\tilde{U}}$ to $(h_t^1)^*\hat{\omega}_t^1 = (h_t^1)^*\tilde{\omega}_t$. The fact that $h_0^\tau = h_1^\tau = id_V$ for all τ and that $\hat{\omega}_t^\nu$ remains constant at $t = 0,1$ for all $\nu \in [0,1]$ implies that G_t^η is respectively fixed at ω_0 and ω_1 when $t = 0$ and $t = 1$ for all $\eta \in [0,1]$. Since $\mathcal{R}$ is Diff_V-invariant, G_t^η is in $\mathrm{Sec}\mathcal{R}$ for all $t, \eta \in [0,1]$. Since the cohomology class does not change under the action of an homotopy, G_t^1 is in a for all $t \in [0,1]$. Moreover, since h_t^τ can be chosen arbitrarily small and $\tilde{\omega}_t$ arbitrarily close to ω_t, we have that G_t^1 is arbitrarily $\mathcal{C}^0$-close to ω_t.

To build the homotopy in V, take for A a core of V (which exists since V is supposed to be open). Let $g^\gamma : V \to Op_V(A)$ a diffeotopy which retracts V in an arbitrarily small neighbourhood $Op_V(A)$ such that $g^0 = id_V$, $g^1(V) \subset Op_V(A)$ and $g^\gamma = id_V$ on A. Define the family of homotopies H_t^γ such that

$$G_t^\tau = \begin{cases} (g^{6t\tau})^*\omega_0, & \tau \in [0,1/2], & t \in [0,1/3] \\ (g^{2\tau})^*\omega_{3t-1}, & \tau \in [0,1/2], & t \in [1/3,2/3] \\ (g^{6(1-t)\tau})^*\omega_1, & \tau \in [0,1/2], & t \in [2/3,1] \\ (g^{3t})^*\omega_0, & \tau \in [1/2,1], & t \in [0,1/3] \\ (g^1)^*G_{3t-2}^{2\tau-1}, & \tau \in [1/2,1], & t \in [1/3,2/3] \\ (g^{3(1-t)})^*\omega_1, & \tau \in [1/2,1], & t \in [2/3,1] \end{cases}$$

It is defined on V and since $\mathcal{R}$ is Diff_V-invariant, it takes its values in $\mathcal{R}$. We have $H_t^0 = \omega_t$, $H_0^\gamma = \omega_0$ and $H_1^\gamma = \omega_1$ for all $t, \gamma \in [0,1]$. For the same reasons as before, the cohomology class of H_t^γ is a for all $t, \gamma \in [0,1]$ and in particular for $\gamma = 1$.

Applying Corollary 3.17 instead of Corollary 3.18 and skipping mentions of t, we also get the result for the non-parametric case. □

We present below an application of the h-principle for open Diff_V-invariant relations.

3.3.4 Application of the h-principle to the Grassmanian bundle

Let W be a q-dimensional manifold and V a n-dimensional submanifold embedded in W. For $l \leqslant q$, we note Gr_lW the *Grassmanian of l-planes* of W, which is defined as the set of all the vectorial spaces of dimension l tangent at W. Denote by $\pi : Gr_lW \to W$ the Grassmanian bundle of the manifold W. The projection π associates an element of Gr_lW to the point $w \in W$ at which this element is tangent to W.

At each point $v \in V$, the differential of the embedding $dp : TV \to TW$ sends the tangent spaces of v to a vectorial subspace of dimension n tangent to W. We denote $Gdf : V \to Gr_nW$ the map which associates $v \in V$ to $df(T_vV) \subset T_{p(v)}W$. More generally, if $F : TV \to TW$ is a monomorphism, we can define in the same way the map $GF : V \to Gr_nW$. This shows that Gr_lW is a natural fibre bundle.

For $A \subset Gr_nW$, a homomorphism $F : TV \to TW$ is said to be *A-directed* if $GF(V) \subset A$.

Theorem 3.28 (A-directed embeddings of open manifolds [9])**.** *Let A be an open subset of Gr_nW, V an open manifold, $f_0 : V \to W$ an immersion such that $F_0 := df_0$ is homotopic in the space of monomorphisms from TV to TW to a certain A-directed F_1 with $bsF_1 = f_0$.*

Then, it exists $f_1 : V \to W$ isotopic f_0 such that $Gdf_1(V) \in A$ and df_1 is homotopic to F_1 in the space of A-directed monomorphisms. Moreover, the isotopy can be chosen as $\mathcal{C}^0$-small as we want on an open neighbourhood of a core K.

Proof. To the open subset A in Gr_nW, we associate an open differential relation,

$$\mathcal{R}_A := \{F \in J^1(V,W) \quad | \quad F \text{ monomorphism and } GF(V) \subset A\}.$$

From a previous example, it is open and Diff_V-invariant. Theorem 3.25 in its simple version gives the existence of a homotopy $\hat{F}_t : V \to \mathcal{R}_A$ between F_1 and a holonomic A-directed monomorphism F_2 that we can choose such that $\mathrm{bs}F_1 = f_0$ and $f_1 := \mathrm{bs}F_2$ are $\mathcal{C}^0$-close on a small neighbourhood of K in V.

Taking the homotopy between F_0 and F_1 followed by $\hat{F}_t$, we obtain a homotopy $\bar{F}_t$ of monomorphisms between two holonomic sections. Then, Theorem 3.25 in its parametric version gives the existence of a holonomic homotopy $\tilde{F}_t : V \to \mathcal{R}_{imm}$ between F_0 and F_2 with $\mathrm{bs}\tilde{F}_t$ $\mathcal{C}^0$-close to $\mathrm{bs}\bar{F}_t$ near K. In particular, $f_0 = \mathrm{bs}F_0$ is isotopic to $f_1 = \mathrm{bs}F_2$, where f_1 is A-directed and the isotopy small around K, and F_1 is homotopic to $df_1 = F_2$ via a homotopy of A-directed monomorphism. □

3.4 Application to the symplectic relation

In this section, we present an application of what we have presented in the symplectic case. More precisely, we have the following theorem, whose proof is a detailed version of the one proposed in [9].

Theorem 3.29 (Isosymplectic embeddings [9])**.** *Let (V, ω_V) and (W, ω_W) be two symplectic manifolds of respective dimensions $n = 2l$ and $q = 2m$. Let $f_0 : V \to W$ be an embedding such that $f_0^*[\omega_W] = [\omega_V]$. Suppose also that $F_0 = df_0$ is homotopic to an isosymplectic homomorphism F_1 via $F_t \subset \mathcal{R}_{imm}$ such that $bsF_t = f_0$ for all $t \in [0,1]$.*

Then, if V is open and $m < l$, there exists an isotopy $f_t : V \to W$ such that f_1 is isosymplectic and df_1 is homotopic to F_1 in the isosymplectic homomorphisms. Moreover, if K is a core of V, we can choose f_t arbitrarily $\mathcal{C}^0$-close to f_0 near K.

Proof. (from [9])

The proof is made in three steps.

Step 1: as the relation associated to the fact of being isosymplectic is not open, we first consider the relation associated to the fact of being symplectic, which is open. Let A_{symp} the associated subset of Gr_nW.

By Theorem 3.28, there exists an isotopy $\tilde{f}_t : V \to W$ such that $\tilde{f}_0 = f_0$, $\tilde{f}_1$ is symplectic and $\tilde{f}_t$ is $\mathcal{C}^0$-close to f_0 on K for all $t \in [0,1]$. Moreover, $d\tilde{f}_1$ and F_1 are homotopic via Φ_t such that $G\Phi_t(V) \subset A_{symp}$.

Since the cohomology class is invariant by homotopy, the assumption that $f_0^*[\omega_W] = [\omega_V]$ implies that $\tilde{f}_1^*[\omega_W] = [\omega_V]$. Then, by Proposition 3.27 applied to $\mathcal{R}_{symp} \subset (\Lambda^p V)^{(1)}$ and

$[\omega_V] \in H^pV$, there exists a homotopy of symplectic forms ω_t between $\tilde{f}_1^*\omega_W$ and ω_V such that $[\omega_t]$ is constant on $[0,1]$. This allows us to write $\omega_t = \omega_0 + d\alpha_t$ for $t \in [0,1]$.

The proof of the theorem is now reduced to the proof of the following proposition:

Proposition 3.30. *Let V a symplectic manifold of dimension $n = 2m$, (W, ω_W) a symplectic manifold of dimension $q = 2l > n$, $h_0 : V \to W$ a symplectic embedding, $\omega_0 = h_0^*\omega_W$ and $\omega_t = \omega_0 + d\alpha_t$ a homotopy of symplectic forms.*

It exists a symplectic isotopy $h_t : V \to W$ as $\mathcal{C}^0$ close to h_0 as we want such that $h_1^*\omega_W = \omega_1$.

If we apply this proposition to $h_0 = \tilde{f}_1$, we can take the homotopy f_t given by $\tilde{f}_{2t}$ on $[0, \frac{1}{2}]$ and h_{2t-1} on $[\frac{1}{2}, 1]$. It is an isotopy as its two parts are isotopies and it verifies that $f_1 = h_1$ is isosymplectic. As $\tilde{f}_t$ is $\mathcal{C}^0$-small on K and h_t $\mathcal{C}^0$-small on the whole V, f_t is $\mathcal{C}^0$-small on K. Finally, since the space of isosymplectic homomorphisms is convex, the linear homotopy $F_t = tdf_1 + (1-t)F_1$ realizes the desired homotopy between F_1 and df_1.

Step 2: we prove the proposition in the case where $w_t = \omega_0 + tdr \wedge ds$ for $r, s : V \to W$ bounded.

From the symplectic neighbourhood theorem, there exists $\epsilon > 0$ such that $h_0 : V \to W$ can be extended to an isosymplectic embedding $\hat{h}_0 : (E, \omega_E) \to (W, \omega_W)$, where

$$E := V \times D_\epsilon^2 \times D_\epsilon^{q-n-2} \qquad \text{and} \qquad \omega_E = \omega_0 \oplus \eta_2 \oplus \eta_{q-n-2},$$

for (D_ϵ^k, η_k) the ball of radius ϵ in $\mathbf{R}^k$ endowed with the restriction to D_ϵ^k of the standard symplectic form of $\mathbf{R}^k$.

Consider $\phi = (r, s) : V \to \mathbf{R}^2$. As r and s are supposed to be bounded, there exists $R > 0$ such that $\phi(V) \subset D_R^2$. Let $\tau_{R,\epsilon} : D_R^2 \to D_\epsilon^2$ be an area preserving map and set $\psi = \tau_{R,\epsilon} \circ \phi$. We then have that

$$(t\psi)^*\eta_2 = (t\phi)^*\tau_{R,\epsilon}^*\eta_2 = (t\phi)^*\eta_2 = t^2 dr \wedge ds.$$

Consider now $\Phi_t : V \to E$ such that $\Phi_t(v) = \big(v, \sqrt{t}\psi(v), 0\big)$. We have

$$\Phi_t^*\omega_E = \Phi_t^*(\omega_0 \oplus \eta_2 \oplus \eta_{q-n-2}) = id^*\omega_0 + \sqrt{t}\psi^*\eta_2 + 0^*\eta_{q-n-2} = \omega_0 + tdr \wedge ds.$$

Now, if we take $h_t : v \mapsto \hat{h}_0(\Phi_t(v))$, we have

$$h_t^*\omega_W = \Phi_y^*\hat{h}_0^*\omega_W = \Phi_t^*\omega_E = \omega_t$$

and $\hat{h}_0(\Phi_0(v)) = \hat{h}_0(v, 0, 0) = h_0(v)$ with $(h_t)_t$ ϵ-small in the $\mathcal{C}^0$ sense.

Step 3: We now reduce the general case, where $\omega_t = \omega_0 + d\alpha_t$, to the case where $\alpha_t = trds$. For that, note that we can consider any homotopy α_t provided that it agrees with the original one at $t = 0, 1$ and that the resulting ω_t remains symplectic.

We first consider $\hat{\alpha}_t$ linear by parts, build this way:

$$\begin{cases} \hat{\alpha}_{t_i} = \alpha_{t_i} \text{ for some points } t_i \in [0,1], \\ \hat{\alpha}_{t_i+\tau} = \alpha_{t_i} + \dfrac{\tau}{t_{i+1} - t_i}(\alpha_{t_{i+1}} - \alpha_{t_i}) \text{ for } \tau \in [0, t_{i+1} - t_i], \end{cases}$$

where the finite set $\{t_i\}_i$ contains 0 and 1 and is chosen such that $\hat{\omega}_t = \omega_0 + d\hat{\alpha}_t$ is symplectic for all $t \in [0,1]$.

We can restrict our analysis to an interval of the form $[t_i, t_{i+1}]$, where ω_t is of the form $\omega_0 + t\alpha$. In fact, applying the proposition on each interval, we obtain a finite number of isotopies h_i that we have to take one after another to obtain an isotopy on the whole $[0, 1]$.

Suppose now that we have a decomposition of this kind: $\alpha = \beta^1 + ... + \beta^L$ with $L \in \mathbf{N}$ and the β^i of the form $r_i ds_i$, where the r_i and s_i are bounded. Consider β_t linear by parts such that

$$\begin{cases} \beta_0 = 0, \\ \beta_{s_j} = \beta^1 + ... + \beta^j \text{ for } s_j = \dfrac{j}{L}, j \in [\![1, L]\!], \\ \beta_{s_j+\tau} = \beta_{s_j} + L\tau\beta^{j+1} \text{ for } \tau \in [0, \dfrac{1}{L}] \end{cases}$$

and $\tilde{\alpha}_t$ such that

$$\begin{cases} \tilde{\alpha}_{t_i} = t_i\alpha \text{ for } t_i = \dfrac{i}{N}, j \in [\![0, N]\!], \\ \tilde{\alpha}_{t_i+\tau} = t_i\alpha + \dfrac{1}{N}\beta_{N\tau} \text{ for } \tau \in [0, \dfrac{1}{N}], \end{cases}$$

where $N \in \mathbf{N}$ can be chosen arbitrarily large. On each subinterval $\left[\frac{i}{N+\frac{j}{NL}}, \frac{i}{N+\frac{j}{NL}}\right]$, $\tilde{\omega} = \omega_0 + d\tilde{\alpha}$ is of the form $\left(\omega_{t_i} + \frac{1}{N}\beta_{s_j}\right) + \tau\beta^{j+1}$ so is linear by parts. With the same argument as previously, we can restrict the problem to the case where ω_t is of the form

$$\omega_0 + td(rds) = \omega_0 + tdr \wedge ds$$

with r and s bounded, which is the case where the proposition is already proved.

It now remains to show how we obtain the decomposition of the 1-form α. We suppose that V is compact: if it not the case, we consider a compact extension. Let $(\rho_i)_{i\in I}$ be a partition of the unity subordinate to an atlas $(U_i, \phi_i)_{i\in I}$ of V, that is such that $\sum_{i\in I} \rho_i = 1$ and $\text{supp}\rho_i$ is compact and included in U_i for every i. For every $i \in I$, we also consider χ_i such that $\chi_i = 1$ on $\text{supp}\rho_i$ and $\text{supp}\,\chi_i \subset U_i$.

Choose a coordinate system $\{x_1, ..., x_n\}$ on $\phi_i(U_i) \subset \mathbf{R}^n$ and write $(\phi_i)_*\alpha\big|_{U_i} = \sum_{j=1}^n a_j^i dx_j^i$. Then,

$$\alpha_{U_i} = \phi_i^*(a_1^i dx_1^i + ... + a_n^i dx_n^i) = (a_1^i \circ \phi_i)\phi_i^* dx_1^i + ... = (a_1^i \circ \phi_i)d(x_1^i \circ \phi_i) + ...$$

Now, see that

$$\alpha = \left(\sum_{i\in I} \rho_i\right)\alpha = \sum_{i\in I} \rho_i\alpha\big|_{U_i} = \sum_{i\in I} \rho_i \sum_{j=1}^n (a_j^i \circ \phi_i)d(x_j^i \circ \phi_i) = \sum_{i,j} \rho_i(a_j^i \circ \phi_i)d\big(\chi_i(x_j^i \circ \phi_i)\big).$$

If we set $r^{ni+j} = \rho_i(a_j^i \circ \phi_i)$ and $s^{ni+j} = \chi_i(x_j^i \circ \phi_i)$, which are compactly supported, we obtain $\alpha = \sum_l r^l ds^l$, which is the desired form. This achieves at the same time the proofs of the proposition and the theorem. □

Remark 3.31. In the proof, we have used the fact that the cohomology class remains constant when we pull back a form by a homotopy. Let us show this. Let ω a closed p-form on W and $f_t : V \to W$ a homotopy. Saying that the cohomology class of $f_t^*\omega$ is constant in $H^k(V)$ is equivalent to say that $f_t^*\omega$ takes fixed values on a base of $H_k(V)$. Let $\sigma : \Delta_k \to V$ be a cycle. We have that

$$\int_{\Delta_k} \sigma^*(f_t^*\omega) = \int_{\Delta_k} (f_t \circ \sigma)^*\omega.$$

Let $\phi : \Delta_k \times [0,1] \to V$ be such that $\phi(x,t) = (f_t \circ \sigma)(x)$. Then, using Stokes theorem, we have

$$\int_{\Delta_k \times [0,1]} d(\phi^* \omega) = \int_{\Delta \times \{1\}} \phi^* \omega - \int_{\Delta \times \{0\}} \phi^* \omega.$$

Since ω is closed, the left hand side of the equation is equal to zero and $f_0^* \omega$ and $f_1^* \omega$ takes the same value on the cycle σ. As this cycle can be arbitrarily chosen, we have proved that the cohomology class of $f_0^* \omega$ and $f_1^* \omega$ is the same.

Conclusion

This internship was in line with last year's internship, where we explored different linear reduction methods for Hamiltonian problems, including the PSD. The objective was to investigate diverse approaches to enhance the reduction provided by the PSD in non-linear cases and formulate theoretical justifications for the symplectic reduction.

Two approaches have been taken: quadratic corrections of the decoder and hyperreduction through optimal control. Tests conducted within hyperreduction via control approach gave promising results. In particular, we presented a variation of the gradient descent which appeared to be very efficient on simple cases. We are still carrying out additional tests to explain it and adapt the method to more complex cases. For the geometrical part of the internship, we continued to read in order to become familiar with some geometrical tools. In particular, we learned about h-principle, which we intend to use to justify the symplectic reduction approach.

I would conclude that the numerical objectives of the internship have been partially achieved. We explored the quadratic correction approach, but we put it aside due to non-satisfying results. The hyperreduction approach produced results, but we are still working on it and testing the methods we presented. On the other hand, we reached the geometrical goals since we are now ready to begin working on the conjecture we want to prove. I could surely have been expected to code more quickly, especially the quadratic corrections, and this is certainly the reason why the numerical part is less developed than originally planned.

During this internship, I honed some skills I acquired during the two years of Masters. From the programming point of view, I employed Python to implement methods I considered, and this has given me the opportunity to practise this language. In the field of numerical analysis, I enriched my knowledge about reduced order models, which we had seen in class in the case of finite elements and which I had already seen during last year's internship. Likewise, I employed the theoretical tools we learned in the optimal control lesson, specifically the adjoint method, which I had to detail in a slightly more difficult case than the ones we had encountered in class. I also worked on my English as almost all of the papers I had to read were written in this language. Furthermore, I developed new skills during this internship: I discovered the h-principle and some very interesting techniques of proof. In the field of soft skills, I often had to take a step back and consider global mechanisms rather than the technical details, without losing sight of the geometric rigour. Because I had a geometrician and two numerical analysts as supervisors, I sometimes had to switch from a point of view to another to understand what they were saying about the same subject. I found these exercises difficult, and I know that I have a lot of room for improvement.

Bibliography

[1] B.M. Afkham and J.S. Hesthaven. Structure preserving model reduction of parametric hamiltonian systems. Preprint at https://arxiv.org/abs/1703.08345, 2017.

[2] V.I. Arnold. *Mathematical Methods of Classical Mechanics*, chapter 7, 8, 9, 10, pages 163–200. Springer, 1989.

[3] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of non linear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(5):3932–3937, 2016.

[4] L. Buhovsky and E. Opshtein. Quantitative h-principle in symplectic geometry. *Journal of Fixed Point Theory and Applications*, 24(2):38, 2022.

[5] J. Chabassier and P. Joly. Energy preserving schemes for nonlinear hamiltonian systems of wave equations: Application to the vibrating piano string. *Computer Methods in Applied Mechanics and Engineering*, 199(45-48):2779–2795, 2010.

[6] R. Chen and M. Tao. Data-driven prediction of general hamiltonian dynamics via learning exactly-symplectic maps. *CoRR*, abs/2103.05632, 2021.

[7] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. *CoRR*, abs/1806.07366, 2018.

[8] Direction de la communication du CNRS. L'histoire du CNRS. url: https://histoire.cnrs.fr, march 2018. Accessed: 2023-07-20.

[9] Y. Eliashberg and N.M. Mishachev. *Introduction to the h-Principle*, volume 48 of *Graduate studies in mathematics*. American Mathematical Society, 2002.

[10] S. Gallot, D. Hulin, and J. Lafontaine. *Riemannian Geometry*, chapter 1,2. 2004.

[11] R. Geelen, S. Wright, and K. Willcox. Operator inference for non-intrusive model reduction with quadratic manifolds. *Computer Methods in Applied Mechanics and Engineering*, 403, 2023.

[12] E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration illustrated by the störmer-verlet method. *Acta Numerica*, pages 399–450, 2003.

[13] INRIA. Inria et son écosystème. url: https://www.inria.fr/fr/innovation-numerique-ecosysteme, november 2020. Accessed: 2022-08-20.

[14] IRMA. L'institut - Présentation. url: https://irma.math.unistra.fr/linstitut/presentation.html. Accessed: 2023-07-20.

[15] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G.E. Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems, 2020.

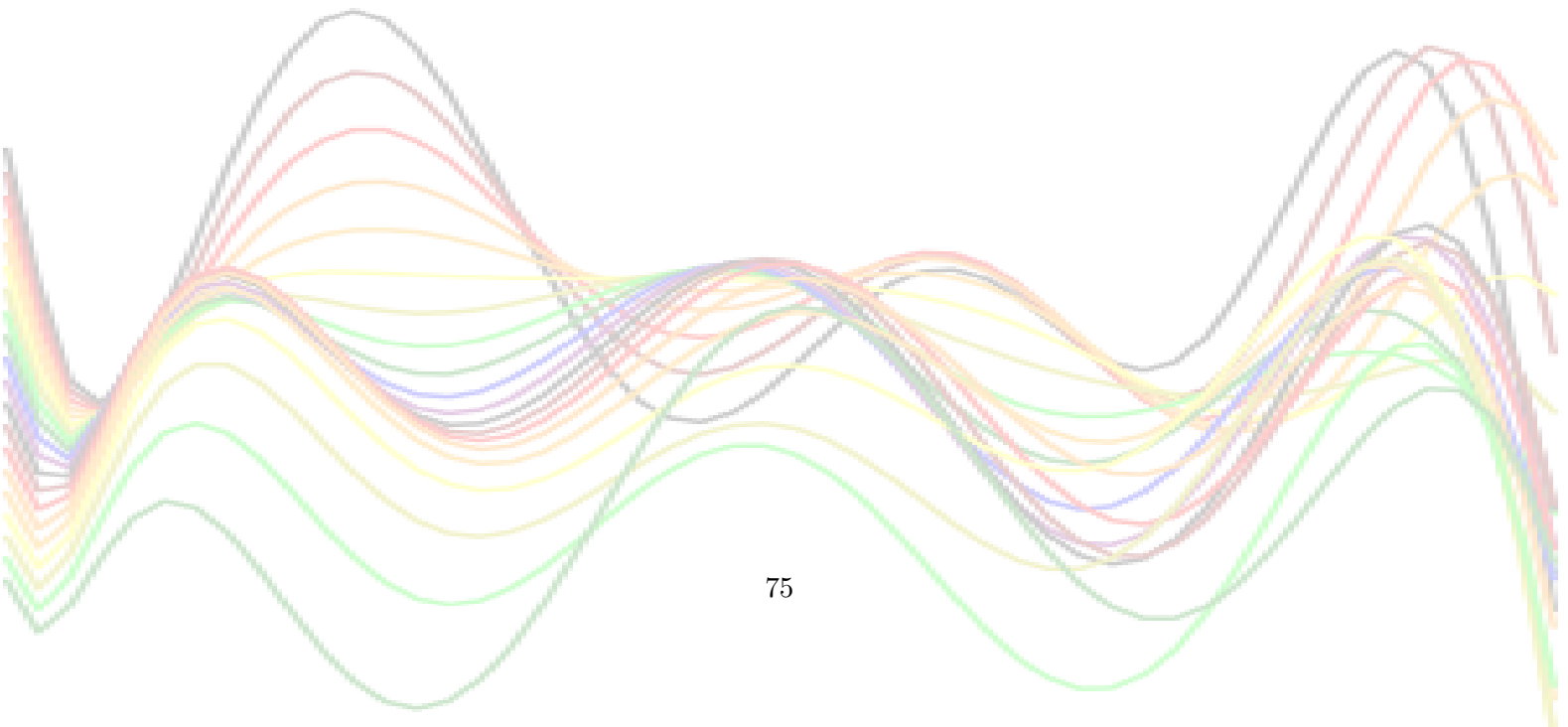
[16] D. McDuff and D. Salamon. *Introduction to symplectic topology*. Oxford University Press, 1998.

[17] J. Moser. On quadratic symplectic mappings. *Mathematische Zeitschrift*, 216:417–430, 1994.

[18] L. Peng and K. Mohseni. Symplectic model reduction of hamiltonian systems. Preprint at https://arxiv.org/abs/1407.6118v2, 2015.

[19] Y.D. Zhong, B. Dey, and A. Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. *CoRR*, abs/1909.12077, 2019.

Appendices



A Quadratic corrections for PSD

In this chapter, we present a different approach to learn the Hamiltonian in low dimension. We worked on it at the beginning of the internship, but it does not give satisfying results. We present our work all the same because it is a lead that we could follow in future works.

As for the previous hyperreduction approach, we supposed that the PSD, or any other linear reduction method, has already given us a decoder $D : \mathbf{R}^{2k} \to \mathbf{R}^{2n}$ and an encoder $E : \mathbf{R}^{2n} \to \mathbf{R}^{2k}$ between the high and the low dimensional spaces. As we have seen for the non-linear piano string model, the induced reduction gives bad results when the equations are too non-linear. This suggests that the submanifold Σ^{2k}, on which lie the solutions of the PDE we are looking at, is too far from being linear

Here, we do not care about hyperreduction, which is viewed as the following step in the reduction process. We actually look if it is possible to improve the passage from the low dimensional space to the high dimensional one by adding a non-linear part to the decoder. More precisely, we want to adapt the method proposed in [11] in the symplectic case and add to the decoder a quadratic term which reduces the compression-decompression error. By doing so, we have to make sure that the new decoder remains symplectic. In the following, we present different variations of this idea and the results we obtained.

A.1 Shears

Before presenting the methods we have explored, we introduce the notion of *shear* that will turn out to be useful to build families of symplectic quadratic maps. We say that a map from $\mathbf{R}^m$ to $\mathbf{R}^r$ is quadratic if all of its r coordinate functions are polynomial of degree inferior or equal to 2.

Definition A.1 (Shear, [17])**.** *A* shear transformation *is a map*

$$\sigma_V : \begin{cases} \mathbf{R}^{2m} \to \mathbf{R}^{2m} \\ (p, q) \mapsto (P, Q) \end{cases}$$

such that $Q_i = q_i$ *and* $P_i = p_i + \partial_{q_i} V(q)$ *with* $V : \mathbf{R}^n \to \mathbf{R}$ *a cubic potential, that is a polynomial function of degree* 3*.*

Proposition A.2. *In the symplectic space* $\mathbf{R}^{2m}$ *endowed with the usual symplectic form, a shear is a symplectic quadratic map.*

Proof. To prove this assertion, it is sufficient to see that σ_V is the transformation induced by the generating function

$$S : (P, q) \mapsto Pq - V(q).$$

In fact, the transformation $(p,q) \mapsto (P,Q)$ induced by S verifies

$$\begin{cases} p_i = \frac{\partial S}{\partial q_i}(P,q) = P_i - \partial_{q_i} V(q), \\ Q_i = \frac{\partial S}{\partial P_i}(P,q) = q_i \end{cases}$$

for all i between 1 and m.

Moreover, as V is supposed to be cubic, σ_V is quadratic. □

Note that the shears form an Abelian group for the composition, and that the correspondence $V \mapsto \sigma_V$ is a homomorphism between the additive group of the cubic functions and the shears.

Let us now see what is the physical meaning of a shear. Consider the Hamiltonian function $H : (p,q) \mapsto \frac{1}{2}|p| + V(q)$. It represents an energy function which is obtained by the sum of a kinetic energy and a potential one. The corresponding system is given by

$$\begin{cases} \dot{p} = -\frac{\partial H}{\partial q} = -V'(q), \\ \dot{q} = \frac{\partial H}{\partial p} = p \end{cases}$$

Denote by (p^t, q^t) the state of the system at a time t. When ϵ comes close to zero, we have

$$\begin{cases} p^{t+\epsilon} \approx p^t - \epsilon V'(q), \\ q^{t+\epsilon} \approx q^t + \epsilon p^t \end{cases}$$

We can see the transformation $(p^t, q^t) \mapsto (p^{t+\epsilon}, q^{t+\epsilon})$, induced by the flow of the equation, as the composition of the transformation $(p^t, q^t) \mapsto (p^t, q^t + \epsilon p^t)$ induced by the Hamiltonian function without potential energy and the shear $\sigma_{\epsilon V}$. Therefore, we can link a shear with the effect of a potential on a system. With that point of view, it seems to be reasonable to try to correct a bad reduction which is particularly wrong on the speed, as it is our case, with a shear.

The following result, proved in [17], gives a normal form, involving shears, for any quadratic symplectic map.

Theorem A.3 (Normal form for symplectic quadratic maps, [17])**.** *Any quadratic symplectic map ϕ on $\mathbf{R}^{2m}$ can be decomposed as the composition of a symplectic linear map l, a shear σ and a symplectic affine function a of $\mathbf{R}^{2m}$, that is*

$$\phi = a \circ \sigma \circ l. \tag{A.1}$$

Moreover, a and l are linked by the formula

$$a = d\phi(0) \cdot l^{-1} + \phi(0).$$

This result implies that any quadratic symplectomorphism is invertible and has another quadratic symplectomorphism for inverse map.

Unfortunately, the proof of this result can not be adapted to the case of symplectic quadratic maps between $\mathbf{R}^{2m}$ and $\mathbf{R}^{2l}$ with $l < m$.

A.2 Quadratic correction with shears in low dimension

In this section, we look for a decoder D_{corr} of the form $D \circ \phi_\lambda$, where $\phi_\lambda : \mathbf{R}^{2k} \to \mathbf{R}^{2k}$ is a quadratic symplectic map.

A.2.1 Expression of the optimisation problem

Expression of the corrected decoder

We first take for the family $\left(\phi_\lambda\right)_\lambda$ the group of the shears. The parameter λ is then the coefficients of V, which we write

$$V(y) = \sum_{1\leqslant i\leqslant k} \lambda_i y_i + \sum_{1\leqslant i\leqslant j\leqslant k} \lambda_{ij} y_i y_j + \sum_{1\leqslant i\leqslant j\leqslant l\leqslant k} \lambda_{ijl} y_i y_j y_l.$$

In the following, R will represent the number of coefficients. We count k terms of order 1 and $\frac{k(k+1)}{2}$ terms of order 2 in the previous expression for V. In the same way, there are

$$\begin{aligned}
\sum_{i=1}^{k}\sum_{j=i}^{k}\sum_{l=j}^{k} 1 &= \sum_{i=1}^{k}\sum_{j=i}^{k}(k-j+1) = \sum_{i=1}^{k}\sum_{m=1}^{k-i+1} m = \sum_{i=1}^{k} \frac{(k-i+1)(k-i+2)}{2} \\
&= \sum_{i=1}^{k}\Big(\frac{(k+1)(k+2)}{2} - i\frac{2k+3}{2} + \frac{i^2}{2}\Big) \\
&= \frac{k(k+1)(k+2)}{2} - \frac{2k+3}{2}\frac{k(k+1)}{2} + \frac{k(k+1)(2k+1)}{12} \\
&= \frac{k(k+1)(k+2)}{6}
\end{aligned}$$

terms of order 3, which gives

$$R = \frac{k(k+1)(k+5)}{6} + k.$$

According to the normal form of symplectic quadratic maps, if we would like to cover the whole space of quadratic symplectomorphism of $\mathbf{R}^{2k}$ in $(\phi_\lambda)_\lambda$, we would have to compose the shears with affines and linear symplectomorphism as in (A.1). This complicates the problem a lot, so we limit ourselves to taking $a = l = id$.

Objective function

We are looking for the $\lambda \in \mathbf{R}^R$ such that

$$\|X - D\phi_\lambda \hat{X}\|^2_{F,2n,N}$$

is minimal, where $\|\cdot\|_{F,2n,N}$ denotes the Froebenius norm in $\mathcal{M}_{2n,N}(\mathbf{R})$, X the matrix of the N sample in $\mathbf{R}^{2n}$ and $\hat{X}$ the matrix of their PSD reduction in $\mathbf{R}^{2k}$. It happens that

$$\begin{aligned}
\|X - D\phi_\lambda \hat{X}\|^2_{F,2n,N} &= \|(P,Q) - D(\hat{P} + \nabla V(\hat{Q}), \hat{Q})\|^2_{F,2n,N} \\
&= \|Q - A_{psd}\hat{Q}\|^2_{F,n,N} + \|P - A_{psd}\hat{P} - A_{psd}\nabla V(\hat{Q}))\|^2_{F,n,N}
\end{aligned}$$

where A_{psd} is the submatrix of D such that $D = \begin{pmatrix} A_{psd} & 0 \\ 0 & A_{psd} \end{pmatrix}$. The previous problem is then equivalent to solving

$$\underset{V\in\mathcal{P}_3(\mathbf{R}^k)}{\operatorname{argmin}} \|P - \hat{P} - \nabla V(\hat{Q}))\|^2_{F,n,N}, \tag{A.2}$$

where $\mathcal{P}_3(\mathbf{R}^k)$ denotes the ring of polynomial functions of degree at most 3 on $\mathbf{R}^k$.

A.2.2 Least-square formulation

Clearly, $\lambda \mapsto \phi_\lambda$ is a linear map. We can therefore rewrite the problem in a way such that it becomes a least-square one. Set

$$\Lambda = {}^t\begin{pmatrix}\lambda_1 & ... & \lambda_k & \lambda_{11} & \lambda_{12} & ... & \lambda_{kk} & \lambda_{111} & ... & \lambda_{kkk}\end{pmatrix} \in \mathbf{R}^R.$$

In what follows, we use the lexicographic order when we work with the elements $\{i,j,l\}$ of $[\![1,k]\!]^3$, starting by the smallest index of the set and finishing by the largest. To simplify further notations, we introduce the function $\mathbf{P} : [\![1,k]\!]^3 \to [\![1,R]\!]$ which associates to the set $\{i,j,l\}$ its position in the ordered list of all the 3-uplets (i_1,i_2,i_3) verifying $i_1 \leqslant i_2 \leqslant i_3$. For $i \leqslant j \leqslant l$, we have that

$$\begin{aligned}
\mathbf{P}(i,i,i) &= \sum_{r=1}^{i-1}\sum_{s=r}^{k}\sum_{t=s}^{k} 1 = \sum_{r=1}^{i-1}\frac{(k-r+1)(k-r+2)}{2} \\
&= \frac{i-1}{2}\left(k(k+3) + i(k+\frac{3}{2}+\frac{i(2i-1)}{6})\right) + i, \\
\mathbf{P}(i,j,j) &= \mathbf{P}(i,i,i) + \sum_{s=i}^{j-1}\sum_{t=s}^{k} 1 = \mathbf{P}(i,i,i) + \frac{(j-i)(2k-i-j+3)}{2}, \\
\mathbf{P}(i,j,l) &= \mathbf{P}(i,j,j) + l - j.
\end{aligned}$$

For $1 \leqslant l \leqslant k$, let $S = 1 + k + \frac{k(k+1)}{2}$. Let also $F_l \in \mathcal{M}_{S,1}(\mathbf{R})$ be the matrix of the map $y \in \mathbf{R}^k \mapsto \partial_l V(y)$, verifying $\partial_l V(y) = F_l \bar{Y}$ with $\bar{Y} = \begin{pmatrix}1 & y_1 & ... & y_k & y_1y_1 & ... & y_ky_k\end{pmatrix}$. For all $1 \leqslant l \leqslant k$, we have that

$$\partial_l V(y) = \lambda_l + \sum_{i=1 i\neq j}^{k} \lambda_{il}x_i + 2\lambda_{ll}x_l + \sum_{1\leqslant i\leqslant j\leqslant k i,j\neq l} \lambda_{ijl}x_ix_j + \sum_{i=1 i\neq l}^{k} 2\lambda_{ill}x_ix_l + 3\lambda_{lll}x_l^2$$

so $F_l = G^l\Lambda$ with G^l in $\mathcal{M}_{S,R}(\mathbf{R})$ such that

$$\begin{cases}
G^l_{1,l} = G^l_{i+1,\mathbf{P}(1,i,l)+k} = G^l_{\mathbf{P}(1,i,j)+k+1,\mathbf{P}(i,j,l)+K+k} = 1, \\
G^l_{l+1,\mathbf{P}(1,l,l)+k} = G^l_{\mathbf{P}(1,i,l)+k+1,\mathbf{P}(i,l,l)+K+k} = 2, \\
G^l_{\mathbf{P}(1,l,l)+k+1,\mathbf{P}(l,l,l)+K+k} = 3
\end{cases} \tag{A.3}$$

for all i,j in $[\![1,k]\!]$ different from l and with $K = \frac{k(k+1)}{2}$. If we set

$$G = \begin{pmatrix} G^1 \\ \hline ... \\ \hline G^k \end{pmatrix} \qquad \text{and} \qquad \bar{\bar{Y}} = \begin{pmatrix} {}^t\bar{Y} & & 0 \\ & \ddots & \\ 0 & & {}^t\bar{Y} \end{pmatrix},$$

we then have that $\nabla V(y) = \bar{\bar{Y}}G\Lambda$. Now, if we want to apply this to the matrix $\hat{Q}$ of the snapshots and if we take

$$\bar{\bar{\hat{Q}}} = \begin{pmatrix} {}^t\bar{\hat{Q}} & & 0 \\ & \ddots & \\ 0 & & {}^t\bar{\hat{Q}} \end{pmatrix},$$

we have to multiply the result by the permutation matrix E with sends the $(ik+j)$-th line of $\bar{\bar{\hat{Q}}}G\Lambda$ on the $(jN+i)$-th one, where N is the number of solutions recorded in X.

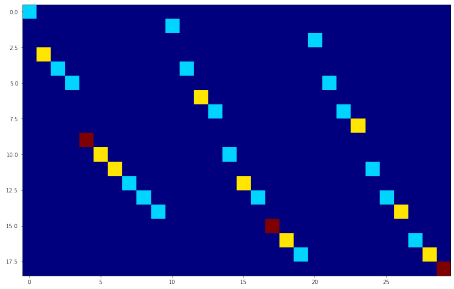


Figure A.1: Visualisation of the matrix $G \in \mathcal{M}_{S,4k^2}(\mathbf{R})$ build in A.3 when $k = 3$. Red squares represent coefficients equal to 3, yellow ones coefficients equal to 2, cyan ones coefficients equal to 1 and blue ones coefficients equal to 0.

This finally gives the least-sqare formulation of the problem (A.2):

$$\underset{V \in \mathcal{P}_3(\mathbf{R}^k)}{\operatorname{argmin}} \|X - E\bar{\bar{\hat{Q}}}G\Lambda\|^2_{F,n,N},$$

whose solution is given by

$$\Lambda = \left({}^t(E\bar{\bar{\hat{Q}}}G)(E\bar{\bar{\hat{Q}}}G)\right)^{-1} {}^t(E\bar{\bar{\hat{Q}}}G)X.$$

A.2.3 Corrected reduced model

Consider a reduce order model with its compression $E : \mathbf{R}^{2n} \to \mathbf{R}^{2k}$ and decompression $D\mathbf{R}^{2k} \to \mathbf{R}^{2n}$ maps. When we compute trajectories, in high or in low dimension, the initial condition $x_0 : \Omega \to M$ is always given in high dimension. Thus, when we do the computations in low dimension, we first have to compress x_0 in the low dimensional space. Now, computations in low dimension are interesting because they are faster than in high dimension but what we are looking for at the end of the process is a solution in high dimension. Then, after having computed $\hat{x}_g$ in low dimension, we decompress it and therefore consider $D\hat{x}_g$. At $t = 0$, we would like to obtain x_0 but we actually have DEx_0.

Unfortunately, we do not have $DE = Id_{2n}$ and DEx_0 is in general different from x_0. However, when D and E are computed using the PSD, E is chosen such that it reaches the minimum of $\|\|DB - I\|\|$ for $B \in \mathcal{M}_{2k,2n}(\mathbf{R})$. In practice, the difference between x_0 and DEx_0 is very slight. Now, when we implement a quadratic correction like the one we have presented, we change the decoder but we keep the encoder. Therefore, the difference between the initial condition and its image after compression-decompression might not be as small as before the correction. To remedy this problem, we change the decoder in function of x_0 and "correct" it by adding the cosntant term $x_0 - DEx_0$.

The new decoder is then $D_{corr} = D \circ \sigma_V + x_{ref}$, where x_{ref} is the corrective additional term described in the previous paragraph. We have

$$D_{corr}(p,q) = (D^p_{corr}(p,q), D^q_{corr}(p,q)) = (A_{psd}p + \nabla V(q), A_{psd}q)$$

and therefore

$$\begin{cases} \nabla_q D^q_{corr}(p,q) = A_{psd}, \\ \nabla_p D^q_{corr}(p,q) = 0, \end{cases} \qquad \begin{cases} \nabla_q D^p_{corr}(p,q) = \nabla^2 V(q), \\ \nabla_p D^p_{corr}(p,q) = A_{psd}. \end{cases}$$

Then, the components of the new reduced Hamiltonian are given by

$$\begin{cases} \nabla_q \hat{H}(p,q) = {}^t\nabla_q D^q_{corr}(p,q) \cdot \nabla_q H(D_{corr}(p,q)) + {}^t\nabla_q D^p_{corr}(p,q) \cdot \nabla_p H(D_{corr}(p,q)) \\ \qquad\qquad = {}^t A_{psd} \cdot \nabla_q H(D_{corr}(p,q)) + {}^t\nabla^2 V(q) \cdot \nabla_p H(D_{corr}(p,q)) \\ \nabla_p \hat{H}(p,q) = {}^t\nabla_p D^q_{corr}(p,q) \cdot \nabla_q H(D_{corr}(p,q)) + {}^t\nabla_p D^p_{corr}(p,q) \cdot \nabla_p H(D_{corr}(p,q)) \\ \qquad\qquad = {}^t A_{psd} \cdot \nabla_p H(D_{corr}(p,q)). \end{cases}$$

A.2.4 Results

We test this correction on the reduced model for the piano string with $g = \alpha$. Solutions and errors computed with the PSD and the corrected PSD are presented on Figures A.3 and A.2. To evaluate the improvements coming only from the quadratic correction, we also add a corrective term $x'_{ref} = x_0 - D_{psd}E_{psd}x_0$ to the reduced model build with the PSD.

Visually, we see no difference between the solution computed with the corrected reduced model and the oslution computed with the original reduced model. Both look very different from the solution computed in high dimension, particularly on the speed and for the longest times. Looking at the error in function of time (Figure A.2), we see that the error on position and on speed rapidly increases after $t = 0.3$ but it is the same for the two models.

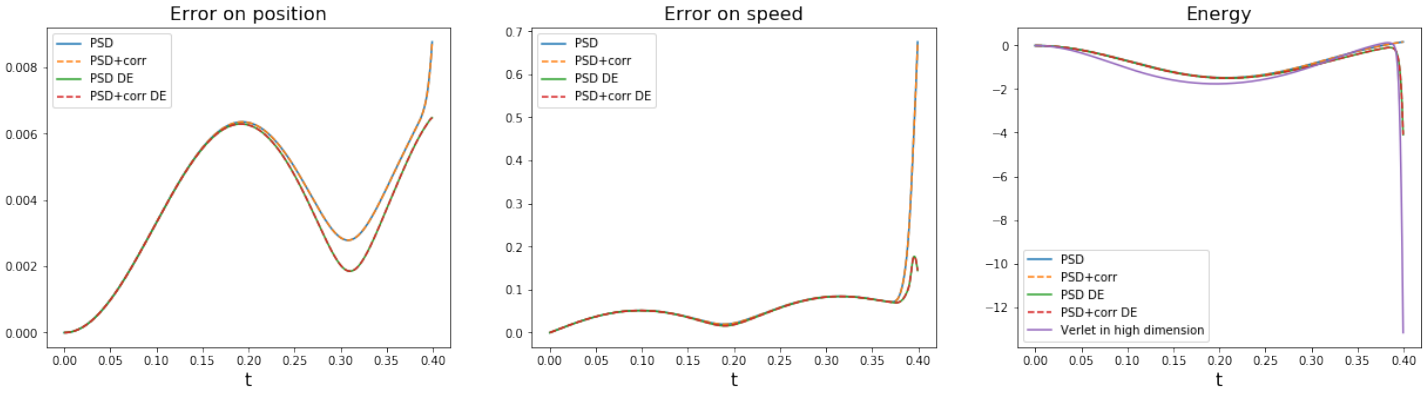


Figure A.2: Estimation of the errors made on the string position (left) and speed (middle) through time and the variations of energy during time (bottom). The reference solution is the solution we computed in high dimension. The estimation of error was made on the trajectory obtained with the value $g = \alpha = 0.99$ for the parameter. Blue and orange curves represents errors and energy obtained through time with the reduced model given by the PSD and the corrected PSD, respectively. Green and red curves represents the errors and the energy through time for the solutions computed in high dimension, compressed in low dimension and decompressed with the decoder coming from the PSD and the corrected PSD, respectively. The reduction made using PSD used 10 trajectories computed in high dimension with explicit Stormer-Verlet scheme for values of g uniformly sampled in $I = [0,1]$, $k = 3$, $n = 200$, $dt = 0.0005$ and $m = 800$. Trajectories in low dimension were computed using the reduced model with the same discretisation in time and space. Initial speed in high dimension was 0 for all $x \in [0,1]$ and initial displacement was given by $x \mapsto (0.1\sin(2\pi x), 0.05\sin(2\pi x))$.

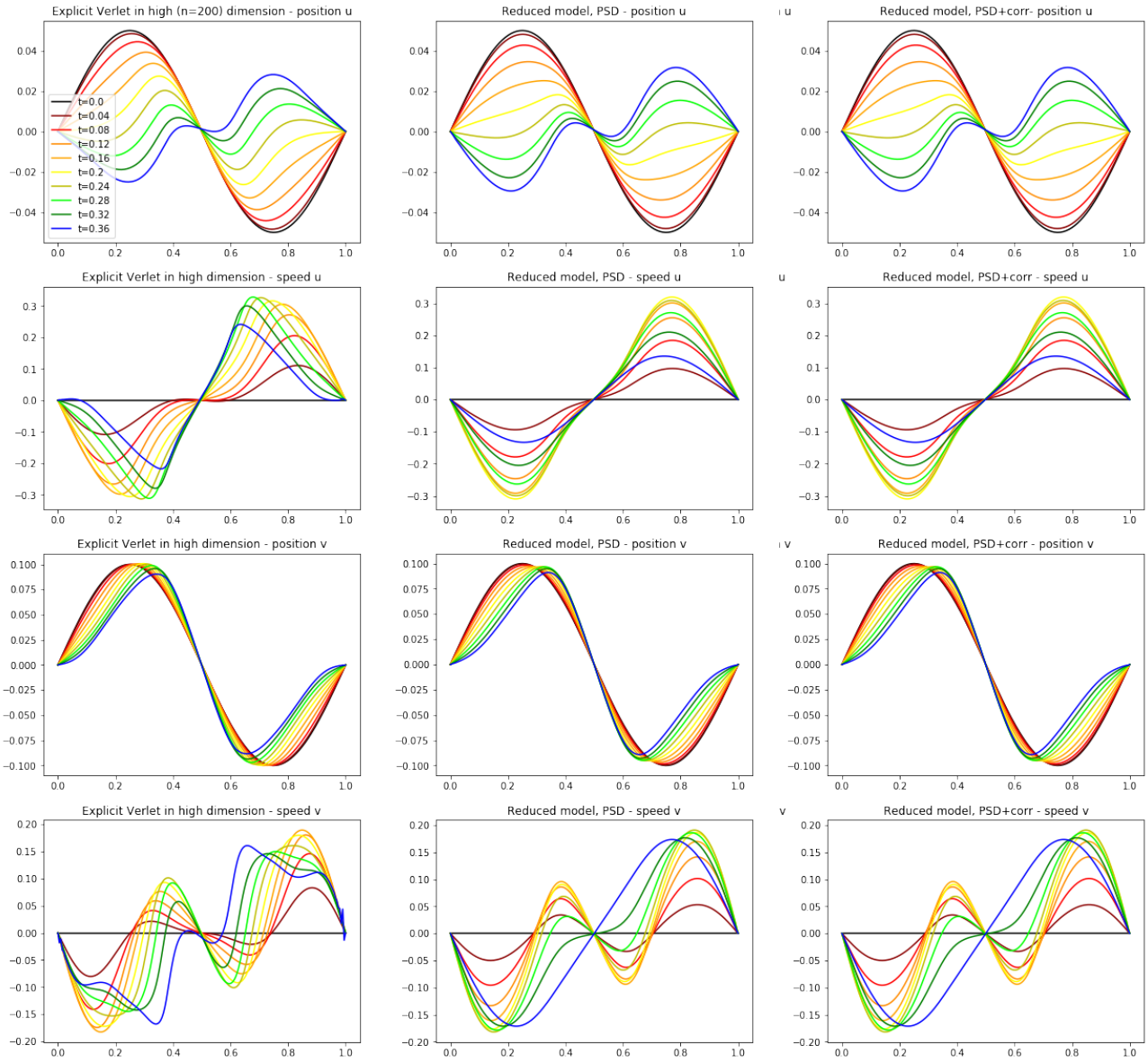


Figure A.3: Numerical solution of non-linear piano string equation with $g = \alpha = 0.99$. The first column represents the piano string position and speed for different times computed in high dimention. The second one represents the trajectory we obtain with the reduced model build with the PSD. The third one represents the trajectory we obtain with the reduced model build with the corrected PSD. On the first line, we show the horizontal speed of the string in function of the position x on the string. On the second line is the horizontal displacement of the string in function of x. On the third line is the vertical speed in function of x. On the fourth line is the vertical displacement of the string in function of x. The reduction was made using PSD with $k = 3$ using 10 trajectories, computed in high dimension with explicit Stormer-Verlet scheme, for values of g uniformly sampled in $I = [0, 1]$, $n = 200$, $dt = 0.0005$ and $m = 800$. Trajectories in low dimension were computed using the reduced model with the same discretisation in time and space. Initial speed in high dimension was 0 for all $x \in [0, 1]$ and initial displacement was given by $x \mapsto (0.1 \sin(2\pi x), 0.05 \sin(2\pi x))$. The value of the parameter g we have taken, 0.81, to compute the solution is not in the set sampled to build the PSD.

This can be explained by the fact that we do not change the image of the decoder in $\mathbf{R}^{2n}$. More precisely, the PSD gives a linear subspace of $\mathbf{R}^{2n}$, but the manifold on which lies the solutions may not be linear at all. The best that we can obtain with the PSD is therefore a linear subspace in which this target manifold is included. In the cases where this manifold is highly non-linear, this subspace can be of high dimension and this is why the linear reduction fails.

Now, when we correct the decoder produced by the PSD, we change the reduced Hamiltonian but we do not change the space in which it takes its values. If the dimension of the linear subspace obtained with the PSD is too high or if it does not include the true manifold, then the quadratic correction has few chance to really improve the resolution in low dimension.

A.3 Additive quadratic correction

Following an idea found in [11], we now look for a decoder of the form $D_{corr} = D_{psd} + \phi_\lambda$.

A.3.1 Tentative 1

Symplecticity condition

We look for ϕ_λ of the form $y \mapsto \bar{A}\tilde{Y}$ where $\tilde{Y} = (y_1 y_1, y_2 y_2, ..., y_{2k} y_{2k})$ and $\bar{A}$ is in $\mathcal{M}_{2n,2k}(\mathbf{R})$. In this case, the Jacobian matrix of the new decoder at point $y \in \mathbf{R}^{2k}$ is given by $D + 2\bar{A}Y$, with $Y = \begin{pmatrix} Y_q & 0 \\ 0 & Y_p \end{pmatrix}$, where Y_q and Y_p are the $n \times k$ diagonal matrices with diagonal coefficients $y_1, ..., y_k$ and $y_{k+1}, ..., y_{2k}$.

Decompose $\bar{A}$ into four submatrices A, B, C and D of $\mathcal{M}_{2n,2k}(\mathbf{R})$:

$$\bar{A} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

The jacobian matrix of the corrected decoder is then

$$\mathcal{J}D_{corr}(y) = \begin{pmatrix} A_{psd} + 2AY_q & 2BY_p \\ 2CY_q & A_{psd} + 2DY_p \end{pmatrix}.$$

According to 1.6, D_{corr} is symplectic if and only if

$$\begin{cases} 2Y_q^t C(A_{psd} + 2AY_q) = 2^t(A_{psd} + 2AY_q)CY_q, \\ 2^t(A_{psd} + 2DY_p)BY_p = 2Y_p^t B(A_{psd} + 2DY_p), \\ {}^t(A_{psd} + 2DY_p)(A_{psd} + 2AY_q) - 4Y_p^t BCY_q = I_k \end{cases}$$

for all Y_p and Y_q diagonal matrices in $\mathcal{M}_{n,k}(\mathbf{R})$. Rearranging the terms and taking into account that ${}^t A_{psd} A_{psd} = I_k$ by construction, we obtain

$$\begin{cases} 2Y_q({}^t AC - {}^t CA)Y_q + {}^t A_{psd} CY_q - Y_q {}^t C A_{psd} = 0, \\ 2Y_p({}^t DB - {}^t BD)Y_p + {}^t A_{psd} BY_p - Y_p {}^t B A_{psd} = 0, \\ 2Y_p({}^t BC - {}^t DA)Y_q - {}^t A_{psd} AY_q - Y_p {}^t D A_{psd} = 0. \end{cases} \tag{A.4}$$

for all Y_p and Y_q.

When we set $(Y_p, Y_q) = (0, I_k)$ and $(Y_p, Y_q) = (I_k, 0)$ in the third equation, we respectively get ${}^t A_{psd} A = 0$ and ${}^t A_{psd} D = 0$. Inserting these results in the third equation with $(Y_q, Y_p) = (I_k, I_k)$,

we have ${}^tBC - {}^tDA = 0$. Conversely, if we have ${}^tA_{psd}A = {}^tA_{psd}D = 0$ and ${}^tBC - {}^tDA = 0$, the third equation is true for all Y_q, Y_p. On the other hand, when we take $Y_q = I_k$ and $-I_k$ in the first equation, we obtain $2({}^tAC - {}^tCA) + {}^tA_{psd}C - {}^tCA_{psd} = 0$ and $-2({}^tAC - {}^tCA) + {}^tA_{psd}C - {}^tCA_{psd} = 0$. Adding the two equation gives ${}^tA_{psd}C - {}^tCA_{psd} = 0$, subtracting them ${}^tAC - {}^tCA = 0$. Now, if we introduce the last expression in the third equation of (A.4), we get ${}^tA_{psd}CY_q - Y_q{}^tCA_{psd} = 0$ for all Y_q. Let us see that the symmetry of ${}^tA_{psd}C$ that we have just shown implies then that ${}^tA_{psd}C = 0$. In fact, if for all indices i, j in $[\![1, n]\!]$,

$$\sum_{l=1}^{n}(A_{psd})_{li}C_{lj} = \left[{}^tA_{psd}C\right]_{ij} = \left[{}^tCA_{psd}\right]_{ij} = \sum_{l=1}^{n}(A_{psd})_{lj}C_{li},$$

then

$$\sum_{l=1}^{n}(A_{psd})_{li}C_{lj}(Y_q)_{jj} = \left[{}^tA_{psd}CY_q\right]_{ij} = \left[Y_q{}^tCA_{psd}\right]_{ij} = \sum_{l=1}^{n}(A_{psd})_{lj}C_{li}(Y_q)_{ii} = \sum_{l=1}^{n}(A_{psd})_{li}C_{lj}(Y_q)_{ii}$$

so

$$((Y_q)_{jj} - (Y_q)_{ii})\left[{}^tA_{psd}C\right]_{ij} = ((Y_q)_{jj} - (Y_q)_{ii})\sum_{l=1}^{n}(A_{psd})_{li}C_{lj} = 0$$

for all indices i, j and for all values of $(Y_q)_{jj}$ and $(Y_q)_{ii}$. This implies that ${}^tA_{psd}C = 0$. Conversely, if we have ${}^tA_{psd}C = 0$ and ${}^tAC - {}^tCA = 0$, then the first equation of (A.4) in true for all value of Y_q. Exactly in the same way, we find that the second equation of (A.4) is equivalent to ${}^tA_{psd}B = 0$ and ${}^tDB - {}^tBD = 0$.

Optimisation problem

Consider the loss function

$$\mathcal{L} : \begin{cases} \mathcal{M}_{n,k}(\mathbf{R})^4 \to \mathbf{R} \\ (A, B, C, D) \mapsto \|\bar{X} - \bar{A}\tilde{\hat{X}}\|^2_{F,2n,N}, \end{cases}$$

where $\bar{X}$ denotes the compression-decompression error made by the PSD on the samples, that is $X - D\hat{X}$.

According to the preceding section, D_{corr} is symplectic if and only if $\bar{A}$ is a zero of the functions

$$\begin{aligned}
g_1 &: (A, B, C, D) \mapsto \|{}^tA_{psd}A\|^2_{F,k,k}, \\
g_2 &: (A, B, C, D) \mapsto \|{}^tA_{psd}B\|^2_{F,k,k}, \\
g_3 &: (A, B, C, D) \mapsto \|{}^tA_{psd}C\|^2_{F,k,k}, \\
g_4 &: (A, B, C, D) \mapsto \|{}^tA_{psd}D\|^2_{F,k,k}, \\
g_5 &: (A, B, C, D) \mapsto \|{}^tAC - {}^tCA\|^2_{F,k,k}, \\
g_6 &: (A, B, C, D) \mapsto \|{}^tDB - {}^tBD\|^2_{F,k,k}, \\
g_7 &: (A, B, C, D) \mapsto \|{}^tBC - {}^tDA\|^2_{F,k,k}.
\end{aligned}$$

Whatever is the dimension of the matrices we are looking at, the Froebenius norm is Euclidean for the scalar product $(\cdot, \cdot) : (A, B) \mapsto Tr({}^tAB)$. On $\mathcal{M}_{n,k}(\mathbf{R})^4$, we use the scalar product induced by the Cartesian product

$$\langle (A, B, C, D); (E, F, G, H) \rangle = (A, E) + (B, F) + (C, G) + (D, H)$$

and the associated norm $|||\cdot|||$.

Note

$$K_i = \{(A,B,C,D) \in \mathcal{M}_{n,k}(\mathbf{R})^4 \mid g_i(A,B,C,D) = 0\}$$

and $K = \bigcap_{i=1}^{7} K_i$. We want to solve

$$\min_{(A,B,C,D)\in K} \mathcal{L}(A,B,C,D).$$

Consider Taylor's expansion of $\mathcal{L}$ at (A,B,C,D) in the direction of (h_1,h_2,h_3,h_4):

$$\begin{aligned}
&\mathcal{L}(A+h_1,B+h_2,C+h_3,D+h_4)\\
&= \|\bar{Q} - (A+h_1)\tilde{\hat{Q}} - (B+h_2)\tilde{\hat{P}}\|^2_{F,n,k} + \|\bar{P} - (C+h_3)\tilde{\hat{Q}} - (D+h_4)\tilde{\hat{P}}\|^2_{F,n,k}\\
&= \|\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}}\|^2_F - 2\left(\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}}, h_1\tilde{\hat{Q}} + h_2\tilde{\hat{P}}\right) + \|h_1\tilde{\hat{Q}} + h_2\tilde{\hat{P}}\|^2_F\\
&\quad + \|\bar{P} - C\tilde{\hat{Q}} - D\tilde{\hat{P}}\|^2_F - 2\left(\bar{P} - C\tilde{\hat{Q}} - D\tilde{\hat{P}}, h_3\tilde{\hat{Q}} + h_4\tilde{\hat{P}}\right) + \|h_3\tilde{\hat{Q}} + h_4\tilde{\hat{P}}\|^2_F.
\end{aligned}$$

Thanks to the properties of the trace, we have

$$\begin{aligned}
\left(\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}}, h_1\tilde{\hat{Q}}\right) &= Tr\left({}^t(\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}})h_1\tilde{\hat{Q}}\right) = Tr\left(\tilde{\hat{Q}}^t(\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}})h_1\right)\\
&= \left((\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}})^t\tilde{\hat{Q}}, h_1\right).
\end{aligned}$$

Equivalent expressions holds for the other terms involving one of the h_i, which gives

$$\begin{aligned}
&\mathcal{L}(A+h_1,B+h_2,C+h_3,D+h_4)\\
&= \mathcal{L}(A,B,C,D) - 2\left((\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}})^t\tilde{\hat{Q}}, h_1\right)) - 2\left((\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}})^t\tilde{\hat{P}}, h_2\right)\\
&\quad - 2\left((\bar{P} - C\tilde{\hat{Q}} - D\tilde{\hat{P}})^t\tilde{\hat{Q}}, h_3\right) - 2\left((\bar{P} - C\tilde{\hat{Q}} - D\tilde{\hat{P}})^t\tilde{\hat{P}}, h_4\right) + \mathcal{O}(|||h_1,h_2,h_3,h_4|||^2)
\end{aligned}$$

so the gradient of $\mathcal{L}$ for our scalar product is

$$-2\left((\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}})^t\tilde{\hat{Q}}, (\bar{Q} - A\tilde{\hat{Q}} - B\tilde{\hat{P}})^t\tilde{\hat{P}}, (\bar{P} - C\tilde{\hat{Q}} - D\tilde{\hat{P}})^t\tilde{\hat{Q}}, (\bar{P} - C\tilde{\hat{Q}} - D\tilde{\hat{P}})^t\tilde{\hat{P}}\right).$$

Using the same arguments, we find that

$$\nabla g_1(A,B,C,D) = (2a^t A_{psd} A, 0, 0, 0)$$

and that ∇g_2, ∇g_3 and ∇g_4 have similar expressions. Unfortunately, we also find that

$$\nabla g_5(A,B,C,D) = 4(C^tCA - C^tAC, 0, A^tAC - A^tCA, 0)$$

which equals to zero when $g_5(A,B,C,D) = 0$. Similar results hold for g_6 and g_7 so we can't use usual theoretical tools to characterise local minima.

Numerically, we will use a gradient descent to find a value of $\bar{A}$ which achieve a small value of the loss.

Results

Despite a large number of attempts, gradient descent did not produce a corrective term that did not worsen the solution obtained in low dimension.

A.3.2 Tentative 2

We now want to add crossed terms $y_i y_j$ for $i \neq j$ in the quadratic map. We then look for ϕ_λ of the form $y \mapsto \bar{A}\tilde{Y}$ where $\tilde{Y} = (y_1 y_1, y_1 y_2, ..., y_{2k} y_{2k})$ and $\bar{A}$ is in $\mathcal{M}_{2n,S}(\mathbf{R})$. Recall from a previous section that $S = 1 + k + \frac{k(k+1)}{2}$ is the dimension of $\tilde{Y}$.

Optimisation problem

Decompose $\bar{A}$ in $\binom{A}{B}$ with A and B in $\mathcal{M}_{n,S}$. Let G the matrix in $\mathcal{M}_{S,4k^2}(\mathbf{R})$ whose coefficients are 0 except for:

- $G_{P(1,i,i);2ki+i}$, which is equal to 2 for all $i \in [\![1, 2k]\!]$,
- $G_{P(1,i,j),2ki+j}$, which is equal to 1 for all $i, j \in [\![1, 2k]\!]$ such that $i < j$,
- $G_{P(1,i,j),2kj+i}$ which is equal to 1 for all $i, j \in [\![1, 2k]\!]$ such that $i < j$.

The Jacobian matrix of $f : y \mapsto \tilde{Y}$ at y is given by $\mathcal{J}f(y) = G\check{Y}$, where

$$\check{Y} = \begin{pmatrix} Y & 0 & ... & 0 \\ 0 & Y & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & ... & 0 & Y \end{pmatrix} \in \mathcal{M}_{4k^2,2k}(\mathbf{R}).$$

Then, for $D_{corr} : x = (p, q) \mapsto (A_{psd}p + Bf(x), A_{psd}q + Af(x))$, we have

$$\begin{cases} \nabla_q D^q_{corr}(\hat{x}) = A_{psd} + AG\check{\hat{Q}}, \\ \nabla_q D^p_{corr}(\hat{x}) = BG\check{\hat{Q}}, \end{cases} \quad \begin{cases} \nabla_p D^q_{corr}(\hat{x}) = AG\check{\hat{P}}, \\ \nabla_p D^p_{corr}(\hat{x}) = A_{psd} + BG\check{\hat{P}}, \end{cases}$$ where

$\check{\hat{Q}}$ and $\check{\hat{P}}$ in $\mathcal{M}_{4k^2,k}(\mathbf{R})$ are such that $\check{\hat{X}} = (\,\check{\hat{Q}} \mid \check{\hat{P}}\,)$. Therefore, the symplecticity conditions for D_{corr} are, for all $\check{\hat{Q}}$ and $\check{\hat{P}}$

$$\begin{cases} {}^tA_{psd}BG\check{\hat{P}} + {}^t\check{\hat{Q}}{}^tG({}^tAB - {}^tBA)G\check{\hat{P}} + {}^t\check{\hat{Q}}{}^tG^tAA_{psd} = 0, \\ {}^t(A_{psd} + AG\check{\hat{Q}})BG\check{\hat{Q}} - {}^t\check{\hat{Q}}{}^tG^tB(A_{psd} + AG\check{\hat{Q}}) = 0, \\ {}^t\check{\hat{P}}{}^tG^tA(A_{psd} + BG\check{\hat{P}}) - {}^t(A_{psd} + BG\check{\hat{P}})AG\check{\hat{P}} = 0. \end{cases}$$

If we take $\check{\hat{Q}} = 0$ in the first equation, we have ${}^tA_{psd}BG\check{\hat{P}} = 0$ for all $\check{\hat{P}}$. We can choose $\check{\hat{P}}$ with all but one coefficient equal to zero. This leads to ${}^tA_{psd}BG_i = 0$ for all column G_i of G, which means that ${}^tA_{psd}BG = 0$. The same argument with $\check{\hat{P}} = 0$ shows that ${}^tA_{psd}AG = 0$. Now, taking $\check{\hat{P}}$ and $\check{\hat{Q}}$ with all but one coefficients equal to zero, we see that all the coefficients of ${}^tG({}^tAB - {}^tBA)G$ are zero. Conversely, if ${}^tA_{psd}AG = {}^tA_{psd}BG = {}^tG({}^tAB - {}^tBA)G = 0$, then the three equation above are satisfied for all $\check{\hat{Q}}$ and $\check{\hat{P}}$ and D_{corr} is symplectic.

The optimisation problem that we want to solve is therefore

$$\min_{(A,B)\in K} \mathcal{L}(A, B, C, D).$$

with the loss function

$$\mathcal{L} : \begin{cases} \mathcal{M}_{n,S}(\mathbf{R})^2 \to \mathbf{R} \\ (A, B) \mapsto \|\bar{X} - \bar{A}\tilde{\hat{X}}\|^2_{F,2n,N}, \end{cases}$$

where $\bar{X}$ denotes the compression-decompression error made by the PSD on the samples, that is $X - D\hat{X}$ and

$$K = \{(A, B) \in \mathcal{M}_{n,S}(\mathbf{R})^2 \quad | \quad {}^tA_{psd}AG = {}^tA_{psd}BG = {}^tG({}^tAB - {}^tBA)G = 0\}.$$

Results

As in the previous case, and despite a large number of attempts, gradient descent did not produce a corrective term that did not worsen the solution obtained in low dimension.

B Generating functions

In this chapter, we present a summary of what we learned about generating functions in [2] and [16]. We expect to use these notions in future reduction methods for Hamiltonian systems. In fact, as we will see, generating functions are useful tools to build symplectomorphisms $f : V \to V$ on a given symplectic manifold (V, ω). With a generating function $S : V \to \mathbf{R}$, we can characterise f. When one wants to learn a Hamiltonian dynamic, this characterisation may be interesting. Here, we take $V = \mathbb{R}^{2n}$ and we use coordinates (p, q), with $q, p \in \mathbb{R}^n$. We endow $\mathbb{R}^{2n}$ with the usual symplectic structure, given by $\omega = d\lambda$ with $\lambda = pdq$.

We are here interested in isosymplectic maps, that is $f : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ such that $f^*\omega = \omega$.

In particular, Hamiltonian flows are isosymplectic transformations:

$$(\phi_H^t)^*\omega = \omega.$$

To show it, first note that this equation is equivalent to $L_{X_H}\omega = 0$. Then, use Cartan's formula:

$$L_z\alpha = \iota_z\alpha + d(\iota_z\alpha),$$

which is true for all p-form α and all vector field z in $\mathbf{R}^{2n}$. In this formula, $L_z\alpha$ represents the Lie derivative of the form α in the direction z and $\iota_z\alpha$ the interior product between z and α. This immediately gives

$$\iota_{X_H} d\omega + d(\iota_{X_H}\omega) = d(dH) = 0.$$

It is obvious that if f is isosymplectic, then the form $\lambda - f^*\lambda$ is closed. In fact, it it even exact: there exists $S : \mathbb{R}^{2n} \to \mathbb{R}$ such that

$$pdq(p, q) + P(p, q)dQ(p, q) = dS(p, q). \tag{B.1}$$

If we assume that the coordinates (q, Q) are independent, we can express S in this coordinate system. Note

$$S_1(q, Q(p, q)) = S(p, q).$$

We have

$$p = \frac{\partial S_1}{\partial q}(q, Q) \quad \text{and} \quad P = -\frac{\partial S_1}{\partial Q}(q, Q). \tag{B.2}$$

Conversely, if a function $S_1 : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ verifies $\det \frac{\partial^2 S_1}{\partial Q \partial q} \neq 0$, then the implicit function theorem applied to $\frac{\partial S_1}{\partial q}$ tells that we can express Q in terms of $p := \frac{\partial S_1}{\partial q}$ and q. If we set $P_1(q, Q) = \frac{\partial S_1}{\partial Q}(q, Q)$ and $P(p, q) = P_1(q, Q(p, q))$, we obtain an isosymplectic transformation $g : (p, q) \mapsto (P, Q)$. In fact, it verifies equation (B.1) so if we apply the exterior derivative and use the fact that $ddS = 0$, we have $g^*\omega = \omega$. The map f is such that p and P satisfy (B.2). We then say that S_1 is the *generating function of* f.

Note that we obtained a canonical transformation from a single map from $\mathbb{R}^{2n}$ to $\mathbb{R}$. Moreover, every isosymplectic transformation which verify the independence condition between q and Q can be obtained from a generating function.

It can happen that q and Q are not independent: this is for example the case in the identity function. This do not mean that previous computations can not be done any more. We can apply the same argument with the coordinates q and P instead of q and Q. We then have

$$p = \frac{\partial S_1}{\partial q}(P, q) \quad \text{and} \quad Q = \frac{\partial S_1}{\partial P}(P, q).$$

For example, a generating function for the identity function is given by $S : (P, q) \mapsto Pq$.

Actually, we can choose any partition $(i_1, ..., i_k), (j_1, ..., j_m)$ de $(1, ..., N)$ such that

$$\det \frac{\partial^2 S_1}{\partial (P_j, Q_i) \partial q} \neq 0.$$

For isosymplectic transformations close to the identity, we can choose generating functions of the form

$$S(P, q) = Pq + \epsilon \bar{S}(P, q, \epsilon).$$

We then have

$$p = P - \epsilon \frac{\partial \bar{S}}{\partial q} \quad \text{and} \quad Q = q + \epsilon \frac{\partial \bar{S}}{\partial P}$$

so if we set $H : (p, q) \mapsto S(p, q, 0)$, we have

$$\frac{dP}{d\epsilon}\Big|_{\epsilon=0} = -\frac{\partial H}{\partial q} \quad \text{and} \quad \frac{dQ}{d\epsilon}\Big|_{\epsilon=0} = \frac{\partial H}{\partial q}.$$

C Codes (Python)

Class Monome

```
def __init__(self,ind,dim):
  '''Parameters:
  (list of int) ind: coefficients that appear in the monom,
  (int) dim: dimension of the space on which is defined the monom
  Returns:
  nothing, creates an instance of Monom.
  '''
  for i in ind:
    assert i<dim,"indices and dimension are not consistent"
  self.ind = ind
  self.dim = dim

def __call__(self,x):
  '''Parameters:
  (array-like) x: point(s) on which we want the evaluation,
  Returns:
  (array-like) the value of the monom at x.
  '''
  assert x.shape[0]==self.dim,"x dimension = {} != dim = {}".format(x.shape[0],
    self.dim)
  res = 1
  for i in self.ind:
    res *= x[i]
  return res

def gradient(self,x):
  '''Parameters:
  (array-like) x: point on which we want the evaluation,
  Returns:
  (array-like) the value of the gradient of the monom at x.
  '''
  assert x.shape[0]==self.dim,"x dimension = {} != dim = {}".format(x.shape[0],
    self.dim)
  res = np.zeros_like(x)
  for i in self.ind:
    r = 1
    ind_mi = self.ind.copy()
    ind_mi.remove(i)
    for j in ind_mi:
      r *= x[j]
    res[i] += r
  return res

def hessian(self,x):
```

```
  '''Parameters:
  (array-like) x: point on which we want the evaluation,
  Returns:
  (array-like) the value of the hessian matrix of the monom at x.
  '''
  assert x.shape[0]==self.dim,"x dimension = {} != dim = {}".format(x.shape[0],
    self.dim)
  if len(x.shape)==1:
    res = np.zeros((self.dim,self.dim))
  else:
    res = np.zeros((self.dim,self.dim,x.shape[1]))
  for i in self.ind:
    ind_mi = self.ind.copy()
    ind_mi.remove(i)
    for j in ind_mi:
      r = 1
      ind_mij = ind_mi.copy()
      ind_mij.remove(j)
      for l in ind_mij:
        r *= x[l]
      res[i,j] += r
  return res
```

Class SinProd

```
def __init__(self,ind,period,dim):
  '''Parameters:
  (list of int) ind: coefficients that appear in the sinusoidal function,
  (float) period: period of the function
  (int) dim: dimension of the space on which is defined the sinusoid
  Returns:
  nothing, creates an instance of SinProd.
  '''
  self.ind = ind
  self.period = period
  self.dim = dim

def __prod_ex(self,x,indices):
  '''Parameters:
  (array-like) x: point on which we want the evaluation,
  (list of int) indices: indices for which the coefficients that does not
    appear in the product.
  Returns:
  (array-like) the product of the coefficients of x whose indices are in self.
    ind but not in indices.
  '''
  if len(x.shape)==1:
    prod = 1
  else:
    prod = np.ones((x.shape[1]))
  for i in self.ind:
    if i not in indices:
      prod *= x[i]
  return prod

def __call__(self,x):
  '''Parameters:
  (array-like) x: point on which we want the evaluation,
  Returns:
```

```
  (array-like) the value of the sinusoid at x.
  '''
  p = self.__prod_ex(x,[])
  return np.sin(2*np.pi*p/self.period)

def gradient(self,x):
  '''Parameters:
  (array-like) x: point on which we want the evaluation,
  Returns:
  (array-like) the value of the gradient of the sinusoid at x.
  '''
  if len(x.shape)==1:
    res = np.zeros((self.dim))
  else:
    res = np.zeros((self.dim,x.shape[1]))

  for i in self.ind:
    p = self.__prod_ex(x,[i])
    res[i] = 2*np.pi*p*np.cos(2*np.pi*p*x[i]/self.period)/self.period
  return res

def hessian(self,x):
  '''Parameters:
  (array-like) x: point on which we want the evaluation,
  Returns:
  (array-like) the value of the hessian matrix of the sinusoid at x.
  '''
  if len(x.shape)==1:
    res = np.zeros((self.dim,self.dim))
  else:
    res = np.zeros((self.dim,self.dim,x.shape[1]))

  for i in self.ind:
    for j in self.ind:
      p = self.__prod_ex(x,[i,j])
      res[i,j] = -x[i]*x[j]*((2*np.pi*p/self.period)**2)*np.sin(2*np.pi*p*x[i]*
   x[j]/self.period)
  return res
```

Class Family

```
def __init__(self,members,k):
  '''Parameters:
  (list) members: instances of Monom or SinProd,
  (int) k: half-dimension of the symplectic space on which the functions
    represented by the elements of members are defined,
  Returns:
  nothing, creates an instance of Family.
  '''
  for i in range(len(members)):
  assert members[i].dim==members[0].dim
  assert 2*k<=members[0].dim
  self.members=members
  self.K=len(self.members)
  self.dim=self.members[0].dim
  self.k=k

def X_F(self,x):
  '''Parameters:
```

```
    (array-like) x: point of which we want the evaluation.
    Returns:
    (array-like) X: matrix whose columns are the value of the Hamiltonian vector
      field induced by the elements of the family, evaluated at x.
    '''
    assert x.shape[0]==2*self.k,"x has wrong shape: {} != {}".format(x.shape
      [0],2*self.k)
    s=list(x.shape)
    s.append(self.K)
    X=np.zeros(s)
    for i in range(self.K):
    Df=self.members[i].gradient(x)
    X[:self.k,:,i]=-Df[self.k:2*self.k]
    X[self.k:,:,i]=Df[:self.k]
    return X

def H_theta(self,x,theta):
    '''Parameters:
    (array-like) x: point on which we want the evaluation,
    (array-like) theta: coefficients in front of each member of the family.
    Returns:
    (array-like) H: Hamiltonian given by the linear combination of the members of
       the family with the coefficients in theta, evaluated in x.
    '''
    assert len(theta)==self.K,"theta has wrong shape: {} != {}".format(len(theta)
      ,self.K)
    assert x.shape[0]==2*self.k,"x has wrong shape: {} != {}".format(x.shape
      [0],2*self.k)
    H=0
    for i in range(self.K):
    H += theta[i]*self.members[i](x)
    return H

def dH_theta(self,x,theta):
    '''Parameters:
    (array-like) x: point on which we want the evaluation,
    (array-like) theta: coefficients in front of each member of the family.
    Returns:
    (array-like) dH: gradient of the Hamiltonian given by the linear combination
      of the members of the family with the coefficients in theta, evaluated in x.
    '''
    assert len(theta)==self.K,"theta has wrong shape: {} != {}".format(len(theta)
      ,self.K)
    assert x.shape[0]==2*self.k,"x has wrong shape: {} != {}".format(x.shape
      [0],2*self.k)
    dH=np.zeros_like(x)
    for i in range(self.K):
    dH += theta[i]*(self.members[i].gradient(x)[:2*self.k])
    return dH

def DpH_theta(self,x,theta):
    '''Parameters:
    (array-like) x: point on which we want the evaluation,
    (array-like) theta: coefficients in front of each member of the family.
    Returns:
    (array-like) dpH: gradient with respect to the k first coordinates of the
      Hamiltonian given by the linear combination of the members of the family
      with the coefficients in theta, evaluated in x.
    '''
    return self.dH_theta(x,theta)[:self.k]
```

```
68
69 def DqH_theta(self,x,theta):
70   '''Parameters:
71   (array-like) x: point on which we want the evaluation,
72   (array-like) theta: coefficients in front of each member of the family.
73   Returns:
74   (array-like) dqH: gradient with respect to the k last coordinates of the
       Hamiltonian given by the linear combination of the members of the family
       with the coefficients in theta, evaluated in x.
75   '''
76   return self.dH_theta(x,theta)[self.k:]
77
78 def HessH_theta(self,x,theta):
79   '''Parameters:
80   (array-like) x: point on which we want the evaluation,
81   (array-like) theta: coefficients in front of each member of the family.
82   Returns:
83   (array-like) HessH: Hessian matrix of the Hamiltonian given by the linear
       combination of the members of the family with the coefficients in theta,
       evaluated in x.
84   '''
85   assert len(theta)==self.K,"theta has wrong shape: {} != {}".format(len(theta)
       ,self.K)
86   assert x.shape[0]==2*self.k,"x has wrong shape: {} != {}".format(x.shape
       [0],2*self.k)
87   s=[2*self.k] + list(x.shape)
88   HessH=np.zeros(s)
89   for i in range(self.K):
90   HessH += theta[i]*(self.members[i].hessian(x)[:2*self.k,:2*self.k])
91   return HessH
92
93 def XH_theta(self,x,theta):
94   '''Parameters:
95   (array-like) x: point on which we want the evaluation,
96   (array-like) theta: coefficients in front of each member of the family.
97   Returns:
98   (array-like) X: Hamiltonian vector field of the Hamiltonian given by the
       linear combination of the members of the family with the coefficients in
       theta, evaluated in x.
99   '''
100  dH=self.dH_theta(x,theta)
101  X=np.zeros_like(dH)
102  X[:self.k]=-dH[self.k:]
103  X[self.k:]=dH[:self.k]
104  return X
105
106 def X2H(self,x,z,theta):
107  '''Parameters:
108  (array-like) x: point on which we want the evaluation,
109  (array-like) z: direction in which we want the differential,
110  (array-like) theta: coefficients in front of each member of the family.
111  Returns:
112  (array-like) X2z: differential at x of Hamiltonian vector field of the
       Hamiltonian given by the linear combination of the members of the family
       with the coefficients in theta, evaluated in z.
113  '''
114  assert z.shape[0]==2*self.k,"z has wrong shape: {} != {}".format(z.shape
       [0],2*self.k)
115  X2z=np.zeros((2*self.k))
116  X2z[:self.k]=-z[self.k:]
```

```
117   X2z[self.k:]=z[:self.k]
118   X2z=self.HessH_theta(x,theta)@X2z
119   return X2z
120
```

Class Descent

```
1  def __init__(self,x_true,alpha,epsilon,dt,F,solver,v0,explicit=True):
2    '''Parameters:
3    (array-like) x_true: targeted trajectories,
4    (float) alpha: coefficient in front of the penalisation term,
5    (float) epsilon: regulation coefficient of the penalisation term,
6    (float) dt: time step used to compute x_true,
7    (Family) F: the generating family of the space in which we process the
       descent,
8    (function) solver: numerical solver used to compute the solutions,
9    (array-like) v0: initial conditions from which we compute the trajectories,
10   (bool) explicit: if solver is explicit or not.
11   Returns:
12   nothing, creates an instance of Descent.
13   '''
14   self.alpha=alpha
15   self.epsilon=epsilon
16   self.dt=dt
17   self.m=x_true.shape[1]
18   self.x_true=x_true.copy()
19   self.F=F
20   self.k=self.F.k
21   self.__solver=solver
22   self.v0=v0
23   self.explicit=explicit
24   self.nb_traj=len(self.v0)
25   self.__theta=None
26   self.__x_theta=None
27
28 def __solve_primal(self):
29   '''Parameters:
30   none.
31   Returns:
32   nothing, updates the solution of the primal system from self.v0, using self.
       solver, for Hamiltonian whose coefficients are self.__theta.
33   '''
34   if self.explicit:
35   DpH_hat=lambda p: self.F.DpH_theta(np.concatenate([p,np.zeros_like(p)]),self.
       __theta)
36   DqH_hat=lambda q: self.F.DqH_theta(np.concatenate([np.zeros_like(q),q]),self.
       __theta)
37   else:
38   DpH_hat=lambda p,q: self.F.DpH_theta(np.concatenate([p,q]),self.__theta)
39   DqH_hat=lambda p,q: self.F.DqH_theta(np.concatenate([p,q]),self.__theta)
40
41   self.__x_theta=np.empty((2*self.k,self.m,self.nb_traj))
42   for i in range(self.nb_traj):
43   self.__x_theta[:,:,i]=self.__solver(DpH_hat,DqH_hat,self.v0[i],self.k,self.m,
       self.dt,border=False)
44
45 def set_theta(self,theta):
46   '''Parameters:
47   (array-like) theta: coefficients in the sapce spanned by the members of the
```

```
    family self.F.
48  Returns:
49  nothing, updates self.__theta and the solution of the primal system using
      self.__solve_primal.
50  '''
51  self.__theta=theta.copy()
52  self.__solve_primal()
53
54 def get_theta(self):
55  '''Parameters:
56  none.
57  Returns:
58  (array-like) self.__theta: current value of theta.
59  '''
60  return self.__theta
61
62 def get_x_theta(self):
63  '''Parameters:
64  none.
65  Returns:
66  (array-like) self.__x_theta: current value of the primal solution.
67  '''
68  return self.__x_theta
69
70 def L(self,theta=None):
71  '''Parameters:
72  (array-like) theta (default=None): coefficients in the sapce spanned by the
      members of the family self.F.
73  Returns:
74  the value of the loss at theta. If theta=None, uses self.__theta.
75  '''
76  if not np.all(theta==None):
77    self.set_theta(theta)
78  return self.alpha*np.sqrt(np.sum(self.__theta**2+self.epsilon)) + self.dt*np.
      sum((self.__x_theta-self.x_true)**2)
79
80 def __f_dual(self,x,a,x_diff):
81  '''Parameters:
82  (array-like) x: primal solution at a given time t,
83  (array-ike) a: adjoint solution at t,
84  (array-like) x_diff: difference between self.x_true at t and x.
85  Returns:
86  numerical flow of the adjoint problem at t.
87  '''
88  return self.F.X2H(x,a,self.__theta) + x_diff
89
90 def __solve_dual(self,first,last,id):
91  '''Parameters:
92  (int) first: index at which starts the window for the computation of the
      gradient,
93  (int) last: index at which ands the window for the computation of the
      gradient,
94  (int) id: index of the trajectory considered.
95  Returns:
96  (array-like) a: adjoint solution of the problem between first*self.dt and
      last*self.dt,
97  (array-like) x: primal solution of the problem between first*self.dt and last
      *self.dt.
98  '''
99  if self.explicit:
```

```
    DpH_hat=lambda p: self.F.DpH_theta(np.concatenate([p,np.zeros_like(p)]),self.__theta)
    DqH_hat=lambda q: self.F.DqH_theta(np.concatenate([np.zeros_like(q),q]),self.__theta)
    else:
    DpH_hat=lambda p,q: self.F.DpH_theta(np.concatenate([p,q]),self.__theta)
    DqH_hat=lambda p,q: self.F.DqH_theta(np.concatenate([p,q]),self.__theta)
    x=self.__solver(DpH_hat,DqH_hat,self.x_true[:,first,id],self.k,last-first,self.dt,border=False)[:,::-1]

    x_diff=x - self.x_true[:,first:last,id][:,::-1]
    a=np.zeros((2*self.k,last-first))
    for i in range(last-first-1):
    a[:,i+1]=a[:,i] - self.dt*self.__f_dual(x[:,i],a[:,i],x_diff[:,i])
    return a[:,::-1],x[:,::-1]


def dL(self,first=0,last=None,theta=None):
    '''Parameters:
    (int) first (default=0): index at which starts the window for the computation of the gradient,
    (int) last (default=None): index at which ands the window for the computation of the gradient,
    (array-like) theta (default=None): coefficients in the space spanned by the members of the family self.F.
    Returns:
    (array-like) grad: gradient of the loss computed between first*self.dt and last*self.dt. If last=None, uses self.m. If theta=None, uses self.__theta.
    '''
    if last==None:
    last=self.m
    if not np.all(theta==None):
    self.set_theta(theta)

    grad=np.zeros((self.F.K))
    for id in range(self.nb_traj):
    a_theta,x=self.__solve_dual(first,last,id)
    X_x_theta=self.F.X_F(x)
    grad -= 2*self.dt*np.sum((X_x_theta*a_theta[:,:,None]),axis=(0,1))
    return grad


def descente(self,theta0,max_iter,rho,beta=0,gamma=0,width=None,Adam=False,if_mask=False,random=False):
    '''Parameters:
    (array-like) theta0: starting point of the descent in the parameter space,
    (int) max_iter: maximum number of iterations in the descent,
    (float) rho: learning rate,
    (float) beta (default=0): momentum coefficient in Adam descent,
    (float) gamma (default=0): regularisation coefficient in Adam descent,
    (int) width (default=None): width of the windows used to compute gradients,
    (bool) Adam (default=False): if uses Adam descent or not,
    (bool) if_mask (default=False): if masks some trajectories at each step,
    (bool) random (default=False): if the windows first point are randomly chosen at each step.
    Returns:
    (array-like) iteres: successive iterates reached during descent,
    (list of float) L_iteres: successive values of the loss reached during descent,
    (list of array) dL_iteres: successive gradients used in the descent.
```

```
150   If width=None, uses self.m.
151   '''
152   if width==None:
153   width=self.m
154   cpt=0
155   M=0
156   S=0
157   nu=1e-8
158   first=0
159   last=width
160   self.set_theta(theta0)
161   iteres=[self.__theta]
162   L_iteres=[self.L()]
163   dL_iteres=[]
164
165   while cpt<max_iter:
166   nbr_batches=self.m//width
167   for i in range(nbr_batches):
168     if cpt>max_iter:
169     break
170     cpt += 1
171     dL_theta=self.alpha*(self.__theta/np.sqrt(self.__theta**2+self.epsilon)) +
      self.dL(first,last)
172     M=beta*M + (1-beta)*dL_theta
173     R=rho*M
174
175     if Adam:
176     S=gamma*S + (1-gamma)*dL_theta**2
177     R=R / (np.sqrt(S) + nu)
178
179     if if_mask:
180     mask=np.random.randint(0,2,self.F.K)
181     R=R*mask
182
183     self.set_theta(self.__theta - R)
184     iteres.append(list(self.__theta))
185     L_iteres.append(self.L())
186     dL_iteres.append(dL_theta)
187
188     if random:
189     first=np.random.randint(0,self.m-w)
190     else:
191     first=(first+width)%self.m
192     last=min(first+width,self.m)
193
194   return np.array(iteres),L_iteres,dL_iteres
```