uc3m | Universidad Carlos III de Madrid

Degree in Applied Mathematics and Computing

2024-2025

Bachelor Thesis

Coordinate-independent integrator of differential equations on smooth manifolds

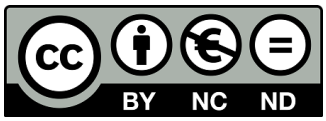
---

Alejandro Fernández-Paniagua

1st Florio Maria Ciaglia

2nd Fabio di Cosmo

Leganés, September 7th 2025



ABSTRACT

This thesis presents the development of the Coordinate-Adaptive method, a geometric numerical integrator for approximating differential equations on smooth manifolds in a coordinate-independent way. Unlike standard schemes that evolve solutions in the embedding space and risk drifting off the manifold, the Coordinate-Adaptive method exploits ideas from Differential Geometry to generate trajectories that respect the geometry of the solution space. The algorithm was implemented in MATLAB and validated mathematically with a rigorous convergence analysis as well as computationally, on the benchmark problem of predicting the state of a qubit in Quantum Mechanics.

Keywords:Differential Equations, Geometric Integrator,Smooth manifolds, Mathematical modeling, Numerical method, Quantum Computing,Differential Geometry

DEDICATION

I dedicate this thesis to my family and friends, whose support has been fundamental to my personal growth and self-discovery. Even if you may not fully understand what follows in these pages, the ideas presented here have only been possible thanks to the energy you have indirectly given me countless times through conversations, shared meals, and theater evenings. I would also like to express my gratitude to my supervisors, Florio and Fabio, for their invaluable guidance and insightful discussions from the very first day this project was introduced to me. You have made the world of Differential Geometry far less intimidating and much more inspiring.

CONTENTS

LIST OF FIGURES

LISTINGS

LIST OF TABLES

1. INTRODUCTION

In this project we develop the Coordinate-Adaptive algorithm, a geometric integrator programmed in MATLAB to approximate the solution of autonomous systems of ordinary differential equations on smooth manifolds. To do this, we learn about the field of Differential Geometry, which provides us with an accurate understanding on the description of dynamical systems on curved and higher-dimensional spaces. From this mathematical insight, we define object-oriented representations that contribute towards the final development of our integrator, which respects the underlying geometry of the solution; an important property missed by most numerical methods. Finally, we apply the algorithm to predict the state of a quantum bit in Quantum Computing, which serves as a benchmark test for assessing the accuracy and robustness of our integrator.

1.1. Geometry of state spaces

In many instances of mathematical modeling, one models the state of the system of interest with a vector $\mathbf{x} \in \mathbb{R}^n$ which obeys an Initial Value Problem of the form

$$\begin{cases} \mathbf{x}' = \mathbf{F}(\mathbf{x}) \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases} \tag{1.1}$$

Typically, finding a closed formula of the solution is not possible so one must resort to numerical integration to approximate it. Due to the abundant examples in science where problems like 1.1 arise, there is a long-established theory on the design of effective and efficient computational methods which tackle this problem [1],[2],[3].

These methods generate a sequence of estimates over a set of discrete times throughout a prescribed interval $[0, T]$, based on a small stepsize.

For instance, the Euler method applied to 1.1 is given by the recursive formula :

$$\mathbf{x}_n = \mathbf{x}_{n-1} + hF(\mathbf{x}_{n-1}), \quad n = 1, \ldots, N.$$

This is one example of a general family of methods known as one-step methods.

An essential requirements of any useful numerical method is that it must be convergent, meaning that the maximum error goes to zero as the step size goes to zero. Also, a measure of the efficiency of a method is given by its convergence rate, which is said to be of order p when the error made by taking a single step remains $O(h^{p+1})$ away from the exact solution. [1]

[1] This type of error is called local error.

In many cases, the solution of 3.1 is constrained to evolve on a distinguished subset $\mathcal{M} \subset \mathbb{R}^n$, known as a submanifold of $\mathbb{R}^n$. A rigorous definition of submanifolds will be given in section 2 but for the moment, a valid mental image is treating $\mathcal{M}$ as a smooth surface contained in $\mathbb{R}^n$.

In this setting, it is not sufficient for an integration scheme to ensure mere convergence. It is equally critical that the numerical trajectory remains confined to $\mathcal{M}$, since the geometric structure of $\mathcal{M}$ often has intrinsic properties of the underlying model such as invariants or conserved quantities carved into its geometry (like energy or angular momentum in classical mechanics).

The problem is that standard approaches will consist in applying a one-step method directly to 1.1, which will fail at giving predictions that stay in $\mathcal{M}$, revealing a need to design numerical structure-preserving methods, also known as geometric integrators which respect the geometry of the solution [4] . The need of having a geometric integrator is accentuated in the context of long-term prediction, where one wants to ensure that numerical errors do not compromise the qualitative properties of the system of interest. In astronomy for example, the author of [5] employed a geometric integrator to simulate the solar system for the next billion years with an accuracy in the order of 10^{-11}, obtaining evidence of the chaotic motion of Pluto, which had been previously observed through the use of another geometric integrator made to study asteroid belts formation and planetary motion [6].

1.2. Qubit problem

A concrete example of a physical system whose dynamics evolve on a submanifold of $\mathbb{R}^3$ arises in the prediction of the state of a qubit, which is the basic unit of computation in a Quantum Computer. The state of a qubit is described mathematically using a 2 dimensional Hilbert space where intuitively speaking, the dimensionality reflects the presence of a preferred physical quantity with only two possible measurement outcomes. For example, a qubit can be realized through the state of an electron bound to a single atom, which can be observed in ground state ($|0\rangle$) or excited state ($|1\rangle$). As we will explain with more precision in chapter 4, the state of the system can be represented by a unit complex vector $|\psi\rangle \in \mathbb{C}^2$ known as the state vector, expressed as a superposition of the computational basis states $|0\rangle, |1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

The complex numbers α and β are known as probability amplitudes as their modulus square are the probabilities of measuring each outcome. This is expressed mathematically by the condition:

$$|\alpha|^2 + |\beta|^2 = 1.$$

As a result of the laws of Quantum Mechanics, the state vector can be uniquely identified with a point in the unit sphere $S^2 \subset \mathbb{R}^3$ whose dynamics are described by the system:

$$\begin{cases} x' = h_z y - h_y z \\ y' = h_x z - h_z x \\ z' = h_y x - h_x y \end{cases}$$

For some constants $h_x, h_y, h_z \in \mathbb{R}$.

Examining the equations,[2] we see that

$$\frac{d}{dt}(x^2 + y^2 + z^2) = 0,$$

which implies that the trajectory is confined to a sphere of radius equal to that of the initial position.

Applying the Euler Method to the problem with $h_x = h_y = 0, h_z = -1$ from the qubit state associated with the point $(1, 0, 0)^T$ results in a trajectory which drifts off the sphere, as shown in the plot of figure 1.1, where the sphere is seen from above, leaving a circle on the $z = 0$ plane.

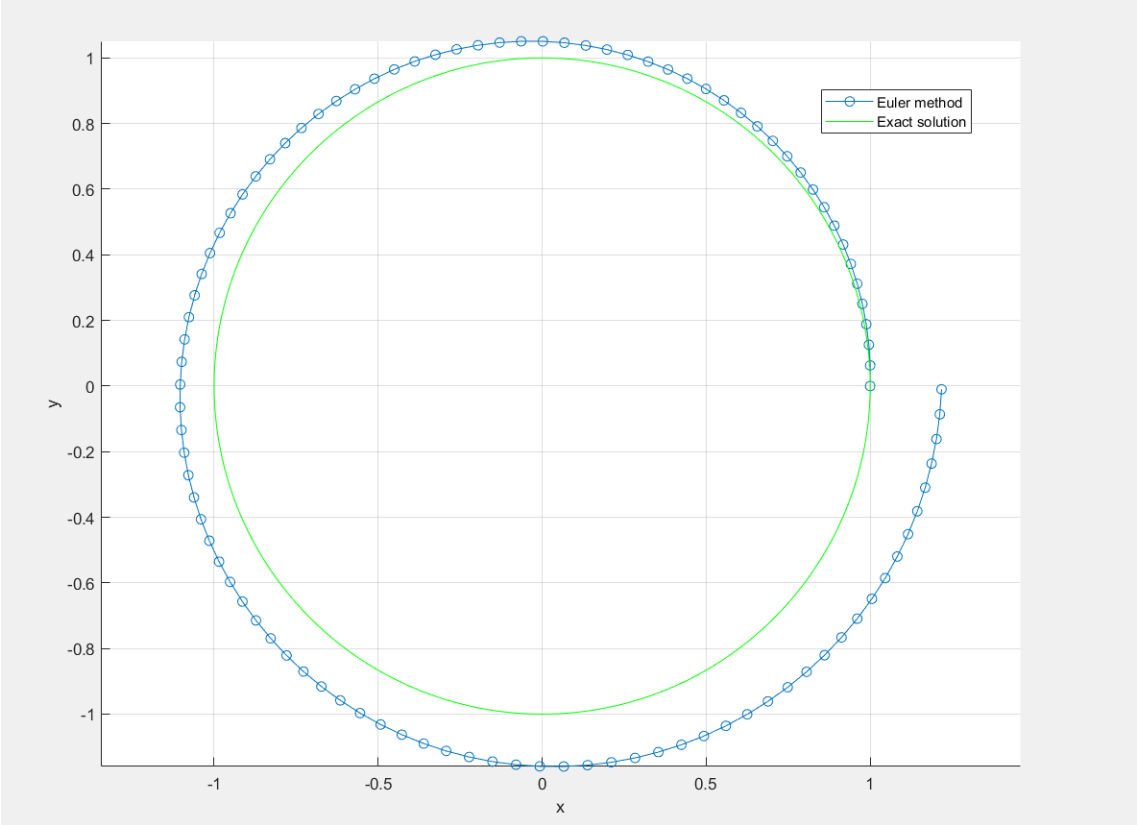


Fig. 1.1. The discrete trajectory obtained using Euler method visibly spirals off the circle, violating the geometric property of the solution.

[2]Note that x', y', z' denote derivatives with respect to t.

A possible solution would be to project the Euler estimates onto the sphere by normalization. However, this solution does not extend well to more complex state spaces, since the projection step will in general require solving a constrained minimization problem [4], only available when the solution space can be expressed as the level set of a smooth function.

1.3. Overview of the proposed solution

The fundamental reason why the estimates obtained by the Euler method do not lie directly inside the manifold has to do with the fact that the differential equations used with it are written with respect to the coordinates of the embedding space $\mathbb{R}^n$. Instead of applying the Euler method directly to those equations, we can leverage the smoothness of the solution space and cover it with a set of local coordinate systems, where we can find equivalent local equations of motion relative to the fundamental degrees of freedom of the space. This will correspond to finding for each coordinates system (U_i, φ_i), an associated "local coordinate vector field" $\hat{V}_i : \varphi_i(U_i) \subset \mathbb{R}^k \to \mathbb{R}^k$ describing motion intrinsically on $\mathcal{M}$[3]. Our algorithm will be able to perform this translation if needed and obtain estimates lying inside the solution space by numerically integrating over the coordinate spaces, aware of how and when to switch coordinates if needed. It is to our knowledge that this idea has been theoretically proposed by Hairer [4] and considered useful in the context of Neural Networks learning [7]. On the other hand, the author of [8] uses its principles as direct inspiration to mathematically formulate another geometric integrator applicable in a special subset of Hamiltonian systems. However, there are almost no works focused on the implementation details and addressing how to translate the required mathematical ideas from Differential Geometry into useful computer representations that can structured to produce an effective computer algorithm. After the development of our solution, we became aware of the doctoral thesis [9], which explores similar ideas to ours but through a fundamentally different implementation strategy. The author deliberately chose this alternative approach, arguing that it was less complex to design. While his method achieves accurate results and is shown to outperform classical schemes, the author acknowledges that the chosen implementation leads to severe inefficiencies and explicitly points toward the kind of approach we have taken as a promising future work direction worth taking.

[3]The sequence $\{U_i\}_{i=1}^n$ will completely cover $\mathcal{M}$ and each domain $U_i \subset \mathcal{M}$ will be paired with a coordinate space, $\varphi_i(U_i) \subset \mathbb{R}^k$ specified by a coordinate system given by some map φ_i.

1.4. Structure of the thesis

The contents of this thesis are structured according to the order which we followed to tackle the problem outlined above. Chapter 2 gives the necessary background in Differential Geometry, introducing key concepts such as the notion of a smooth manifold,atlases, the Tangent Bundle, smooth maps, vector fields and integral curves. As we cover these notions, we occasionally refer to our Matlab code to show how it is that we have implemented them. This section is also supported with concrete examples that convey the concepts learned clearly, as this area of mathematics was new to the author. Secondly, Chapter 3 gives a detailed explanation of how the algorithm is developed, beginning with an overview of how the algorithm operates that leads to a specific discussion of each of the separate components of its code and how they interrelate with the full algorithm's pseudo-code. Subsequently, we present an illustrative case highlighting potential difficulties of the algorithm and discuss how to prevent them. This is followed by a complete convergence analysis, where we prove the convergence of our integrator. Next, Chapter 4 introduces the mathematical formulation of Quantum Mechanics by presenting its fundamental postulates and illustrating them through the qubit case. The formalism based on state vectors is first examined, highlighting its limitations and motivating the formulation in terms of density operators. Building on this, the geometrical representation of a qubit state within the Bloch Ball is discussed, and the corresponding dynamics under a time-independent Hamiltonian are derived. Finally, the proposed algorithm is applied to this benchmark problem, with numerical experiments and plots provided to validate the effectiveness of the integrator. Chapters 5 and 6 address the socio-economic context and the regulatory framework of the project. Chapter 7 concludes the thesis by summarizing the main findings and outlining potential directions for future work. Finally, the code required to implement the Coordinate-Adaptive algorithm is included in the appendix of this document.

2. DIFFERENTIAL GEOMETRY

2.1. Topological manifolds

A manifold is a generalization of a set which locally looks like $\mathbb{R}^n$. For example, surfaces like the torus $\mathbb{T}^2$, the sphere $\mathbb{S}^2$ or $\mathbb{R}^n$ itself are some basic ones. But more abstract sets like a finite dimensional vector space, a 100-dimensional sphere and Lie groups like the group of rotations of space SO(3) can also be given the structure of a manifold. In order to define what a manifold is, we first need to understand some notions in topology, which are used to define the most basic kinds of manifolds, known as topological manifolds. We do this following the texts [10]–[12].

Definition 1. (**Topological space and continuity**) A topology over the set $\mathcal{M}$ is a collection of subsets of $\mathcal{M}$, $\tau \subseteq \mathcal{P}(\mathcal{M})$ verifying:

1. $\emptyset, X \in \tau$.
2. The union of any collection of sets in τ is also in τ.
3. The intersection of any finite collection of sets in τ is also in τ.

We also say that $(\mathcal{M}.\tau)$ is a topological space and the elements of τ are known as open sets. Additionally, we say that a function $f : X \to Y$ between the topological spaces (X, τ_x), (Y, τ_y) is continuous when $f^{-1}(V) \in \tau_x$ for every open $V \in \tau_y$.

Example 2.1. (**Standard topology on** $\mathbb{R}^n$) Perhaps the simplest case is $\mathcal{M} = \mathbb{R}^n$ with the topology induced by its standard euclidean metric d_2 where a set $U \subseteq \mathbb{R}^n$ is open if for each point $p \in U$ there is $\epsilon > 0$ with $B_\epsilon(p) = \{q \in \mathbb{R}^n : d_2(p, q) < \epsilon\} \subset U$.

Definition 2. Given a topological space (X, τ_z) and a subset S of X, the subspace topology on S is defined by $\tau_s = \{S \cap U, U \in \tau_x\}$. [a]

[a] This definition makes sense as it can be shown that τ_s satisfies definition 1.

Example 2.2. The subspace topology of the circle $S^1 = \{x^2 + y^2 = 1\} \subset \mathbb{R}^2$ consists of a set of curved pieces with no boundary. For instance, the northern hemisphere $S^1 \cap \{y > 0\}$ is open while the east pole, $(1, 0)$ or the union of both are not.

We remark that throughout the thesis, subsets of topological spaces will be assumed to be equipped with the subspace topology unless otherwise stated.

Definition 3 (**Topological manifold, chart and atlas** [11])**.** A topological space $(\mathcal{M}, \tau)$ is a **topological manifold** of dimension m if for every point $p \in \mathcal{M}$ there exists an open set $U \subset \mathcal{M}$ together with a map $\varphi : U \to \mathbb{R}^m$ such that $\varphi(U)$ is open, φ is continuous and it has a continuous inverse, φ^{-1}.[a] A pair (U, φ) is called a **chart** on M and φ is a **coordinate system** on U. This can also be written as $(U, (\xi^1, \xi^2, \ldots, \xi^n))$ or $(U, (\xi^i))$ to emphasize the coordinate component functions. Also, a collection of charts whose domains cover the manifold is known as an **atlas**.

[a] An equivalent but shorter way to say this is that φ is a homeomorphism between two open sets in $\mathcal{M}$ and $\mathbb{R}^m$ or that U is homeomorphic to an open set in $\mathbb{R}^m$.

Observe that based on definition 3, the map defined by $\varphi^{-1}(\theta) = (\cos(\theta), sin(\theta))$ with $\theta \in [0, 2\pi)$ does not classify as a valid coordinate system on S^1 since although φ^{-1} is continuous, φ is not: $[0, \pi)$ is open in $[0, 2\pi)$ while $\varphi^{-1}([0, \pi))$ is not open in S^1.

The terminology introduced above has a strong geometrical interpretation when looking at the special case where $\mathcal{M}$ is a surface like that of figure 2.1 , which can be covered by several patches (open domains of the charts) with coordinate grid lines on top (the coordinate systems of the charts).

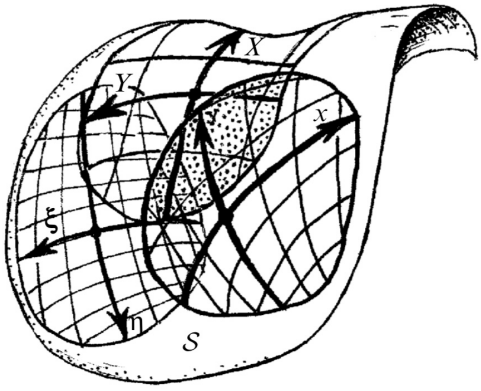


Fig. 2.1. A surface S with three overlapping charts covering it from "The road to reality" by Roger Penrose [13].

Furthermore, notice that on the overlaps there will be two ways of describing a point, which motivates the introduction of the transition functions, that allow changing coordinates from one system to another.

Definition 4 (**Transition functions**)**.** The transition functions between charts (U, φ) and (V, ψ) are :

$$\psi \circ \varphi^{-1} : \varphi(U \cap V) \to \psi(U \cap V) \quad \text{and} \quad \varphi \circ \psi^{-1} : \psi(U \cap V) \to \varphi(U \cap V)$$

2.2. Smooth atlases

We are particularly interested in a special class of topological manifolds where the tools of calculus can be extended, where the charts not only cover the manifold but also "glue" to each other smoothly on the overlaps, allowing easy translations of measures like the velocity of a particle from one chart into the other. These are called **smooth manifolds** and have the additional feature of admitting a **smooth atlas**, which is an atlas whose chart's transition functions are C^∞. [4]

This has a beautiful geometric interpretation, in the context of a surface like 2.1, meaning that one can (locally) transform one coordinate grid into the other by translating, scaling or rotating the first one by the right amount. In particular, the precise scaling and rotation amounts are encoded by the Jacobian matrix of the transition map.

Example 2.3 (**Plane slices atlas**). The n-th dimensional sphere $S^n = \{x_1^2 + ... + x_n^2 + x_{n+1}^2 = 1\}$ in $\mathbb{R}^{n+1}$ can be given a smooth atlas of $2(n+1)$ charts by splitting it into two hemispheres from the positive and negative halfspaces of a frozen component (see 2.2). Then, the coordinates of a point on each hemisphere are given by the other n components. For example, for the circle, S^1, the explicit form of the atlas is $\mathcal{A} = \{(N, \varphi_N), (S, \varphi_S), (E, \varphi_E), (W, \varphi_W)\}$ where :

$$\begin{aligned}
\varphi_N(x, y) &= x \quad N = \{(x, y) \in S^1 : y > 0\},\\
\varphi_S(x, y) &= x \quad S = \{(x, y) \in S^1 : y < 0\},\\
\varphi_E(x, y) &= y \quad E = \{(x, y) \in S^1 : x > 0\},\\
\varphi_W(x, y) &= y \quad W = \{(x, y) \in S^1 : x < 0\},
\end{aligned}$$

These are continuous maps from the 4 open hemispheres of S^1 onto the open interval $(-1, 1)$ and have continuous inverses:

$$\begin{aligned}
\varphi_N^{-1}(\xi) &= (\xi, \sqrt{1-\xi^2}), \quad |\xi| < 1,\\
\varphi_W^{-1}(\xi) &= (-\sqrt{1-\xi^2}, \xi), \quad |\xi| < 1,\\
\varphi_S^{-1}(\xi) &= (\xi, -\sqrt{1-\xi^2}), \quad |\xi| < 1,\\
\varphi_E^{-1}(\xi) &= (\sqrt{1-\xi^2}, \xi), \quad |\xi| < 1.
\end{aligned}$$

Up to here, this shows that $\mathcal{A}$ is an atlas for the circle. But it is also smooth as the

[4] Note that from the definition of chart , the domains and images of the transition maps between φ and ψ are open subsets of $\mathbb{R}^n$ and because $(\psi \circ \varphi^{-1})^{-1} = \varphi \circ \psi^{-1}$, in practice, verifying differentiability of one of them is enough to verify chart compatibility.

transitions are C^∞:

$$\varphi_W \circ \varphi_N^{-1}(\xi) = \sqrt{1-\xi^2}, \quad \xi \in (-1,0),$$

$$\varphi_S \circ \varphi_W^{-1}(\xi) = -\sqrt{1-\xi^2}, \quad \xi \in (-1,0),$$

$$\varphi_E \circ \varphi_S^{-1}(\xi) = -\sqrt{1-\xi^2}, \quad \xi \in (0,1),$$

$$\varphi_N \circ \varphi_E^{-1}(\xi) = \sqrt{1-\xi^2}, \quad \xi \in (0,1),$$

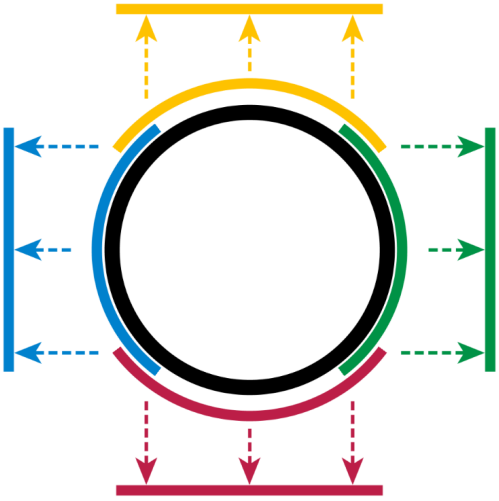


Fig. 2.2. A way to define a smooth atlas on the circle S^1 from [14].

Example 2.4 (**Stereographic projection atlas**). Another important atlas that can be constructed for the family of spheres and in particular for S^2 in $\mathbb{R}^3$ which will play an important role later in the thesis. One can imagine infinite light rays emanating from a light source at the north pole N and intersecting the sphere before going through the plane holding the sphere at $z = -1$. This defines a coordinate system on $\mathrm{S}^2 \setminus \{N\}$ whose image is the whole $\mathbb{R}^2$, as depicted by figure 2.3. Symmetrically, the same can be done by shooting rays from the south pole S, covering $\mathrm{S}^2 \setminus \{S\}$. Altogether these charts cover the sphere.

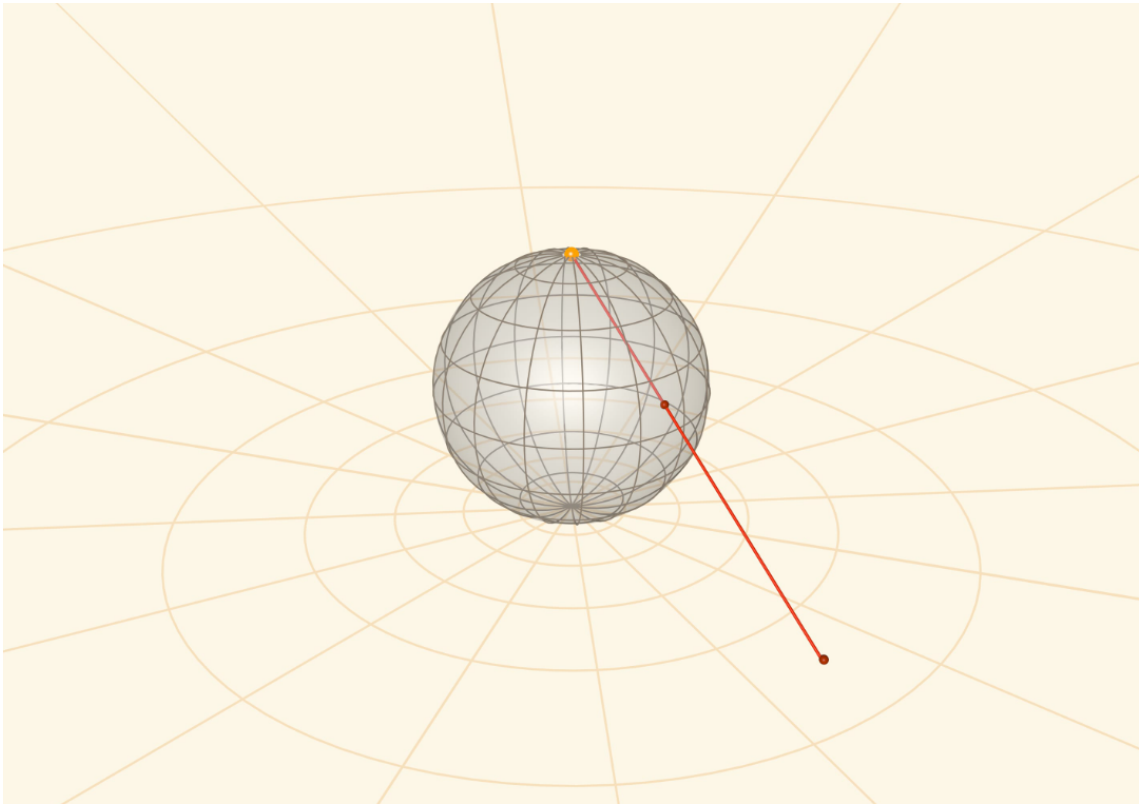


Fig. 2.3. Stereographic map from the north pole onto bottom plane, from the Mathematical Etudes Foundation website[15].

The form of these charts $(U_1 = S^2 \setminus \{N\}, \phi_1)$ and $(U_2 = S^2 \setminus \{S\}, \phi_2)$ is :

$$\phi_1(x, y, z) = (\frac{2x}{1-z}, \frac{2y}{1-z})$$

$$\phi_1^{-1}(\xi_1, \xi_2) = (\frac{4\xi_1}{\|\xi\|^2+4}, \frac{4\xi_2}{\|\xi\|^2+4}, \frac{\|\xi\|^2-4}{\|\xi\|^2+4})$$

$$\phi_2(x, y, z) = (\frac{2x}{z+1}, \frac{2y}{z+1})$$

$$\phi_2^{-1}(\xi_1, \xi_2) = (\frac{4\xi_1}{\|\xi\|^2+4}, \frac{4\xi_2}{\|\xi\|^2+4}, \frac{4-\|\xi\|^2}{\|\xi\|^2+4})$$

which can be obtained by parameterizing the line from pole to point and then solving for the time when it hits the plane (and the other way around). Interestingly, this is the least amount of charts needed to cover the 2-sphere (indeed for any dimension). The atlas can be seen to be smooth as the transition function :

$$\phi_1 \circ \phi_2^{-1}(\xi_1, \xi_2) = (\frac{4\xi_1}{\|\xi\|^2}, \frac{4\xi_2}{\|\xi\|^2})$$

is of class C^∞ on $\phi_2(U_1 \cap U_2) = \mathbb{R}^2 \setminus \{\mathbf{0}\}$.

Naturally, there are many smooth atlases and their unions may still be smooth.[5] For this reason, we need another definition to resolve this sort of redundancy among atlases and regard them as defining the same structure on the manifold.

Definition 5 (**Maximal atlas**)**.** A smooth atlas $\mathcal{A}$ is **maximal** when it is not a proper subset of any larger smooth atlas, that is, there is no chart outside of $\mathcal{A}$ compatible with all charts in $\mathcal{A}$.[a]

[a] Another name used in the literature is Differentiable Structure.

Moreover, a smooth manifold won't be strictly speaking defined with a finite smooth atlas like the ones given above but rather by **the unique maximal smooth atlas** which contains it, something that it is well defined according to the following constructive theorem:

Theorem 1. Let $\mathcal{M}$ be a topological manifold and $\mathcal{A}$ a smooth atlas for $\mathcal{M}$. Then, there exists a unique maximal atlas $\overline{\mathcal{A}}$ containing $\mathcal{A}$.

Proof. Let $\overline{\mathcal{A}}$ be the set of all charts smoothly compatible to every chart in the given atlas $\mathcal{A}$. We will show that $\overline{\mathcal{A}}$ is the unique maximal atlas containing $\mathcal{A}$.

First, let us show that $\overline{\mathcal{A}}$ is a smooth atlas by considering a pair of charts $(U, \varphi), (V, \psi)$ in $\overline{\mathcal{A}}$. Notice that these two will clearly be smoothly compatible with each other when they are "old" charts from $\mathcal{A}$, by the assumption that $\mathcal{A}$ is a smooth atlas. Also ,when one is an "old" chart from $\mathcal{A}$ while the other is not, compatibility holds from the definition of $\overline{\mathcal{A}}$. Now, the interesting question is whether two "new" charts in $\overline{\mathcal{A}}$ are smoothly compatible with each other or not. Indeed, they are smoothly compatible with each other, because for $z = \phi(p) \in \phi(U \cap V)$ there exists some "old" chart $(W, \theta) \in \mathcal{A}$ with $p \in U$, as $\mathcal{A}$ is an atlas for $\mathcal{M}$. Moreover, $p \in U \cap V \cap W$ and hence, the transition function can be rewritten as $\psi \circ \varphi^{-1} \equiv (\psi \circ \theta^{-1}) \circ (\theta \circ \varphi^{-1})$ for some open ball around z. Then, as $\psi \circ \theta^{-1}$ and $\theta \circ \varphi^{-1}$ are smooth by definition of $\overline{\mathcal{A}}$, it follows that $\psi \circ \varphi^{-1}$ is smooth at z. But then, the fact that z is arbitrary and $\mathcal{A} \subset \overline{\mathcal{A}}$ proves that $\overline{\mathcal{A}}$ is a smooth atlas.

Similarly, $\overline{\mathcal{A}}$ is maximal since the charts of any smooth atlas containing it must be in particular smoothly compatible to all those in $\mathcal{A}$ but then, they are in $\overline{\mathcal{A}}$.

Finally, uniqueness follows from the fact that if $\hat{\mathcal{A}}$ is a maximal atlas for $\mathcal{M}$ then, by maximality, $\overline{A} \subset \hat{\mathcal{A}}$. Conversely, every chart in $\mathcal{A}$ must be in particular smoothly compatible to every chart in $\mathcal{A}$ which means $\hat{\mathcal{A}} \subset \overline{A}$. Therefore, $\hat{\mathcal{A}} = \overline{A}$. □

[5] For example, one can see that the charts in the stereographic atlas are all compatible with those from the planes slicing atlas.

Illustrating the Size of a Maximal Atlas

To illustrate the size of the maximal atlas, consider the 4-chart smooth atlas proposed for the circle in example 2.3 and imagine using the same splitting of the circle into the north, south, east and west regions relative to a basis which corresponds to counterclockwise rotation by $\theta \in (0, 2\pi)$ radians of the canonical one. That is, using coordinates (a, b) defined via:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}}_{T_\theta} \begin{pmatrix} a \\ b \end{pmatrix} \quad \text{equivalent to} \quad \begin{pmatrix} a \\ b \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}}_{T_\theta^{-1}} \begin{pmatrix} x \\ y \end{pmatrix}$$

Then, the resulting atlas will be of the form:

$$\mathcal{A}_\theta = \{(U_{N,\theta}, \varphi_{N,\theta}), (U_{W,\theta}, \varphi_{W,\theta}), (U_{S,\theta}, \varphi_{S,\theta}), (U_{E,\theta}, \varphi_{E,\theta})\}$$

With coordinate functions and their inverses:

$$\varphi_{N,\theta}(x, y) = \cos(\theta)x + \sin(\theta)y \quad \text{where } U_{N,\theta} = S^1 \cap \{-\sin(\theta)x + \cos(\theta)y > 0\}$$

$$\varphi_{W,\theta}(x, y) = -\sin(\theta)x + \cos(\theta)y \quad \text{where } U_{W,\theta} = S^1 \cap \{\cos(\theta)x + \sin(\theta)y < 0\}$$

$$\varphi_{S,\theta}(x, y) = \cos(\theta)x + \sin(\theta)y \quad \text{where } U_{S,\theta} = S^1 \cap \{-\sin(\theta)x + \cos(\theta)y < 0\}$$

$$\varphi_{E,\theta}(x, y) = -\sin(\theta)x + \cos(\theta)y \quad \text{where } U_{E,\theta} = S^1 \cap \{\cos(\theta)x + \sin(\theta)y > 0\}$$

$$\varphi_{N,\theta}^{-1}(\xi) = T_\theta(\xi, \sqrt{1-\xi^2}) \quad \text{where } |\xi| < 1$$

$$\varphi_{W,\theta}^{-1}(\xi) = T_\theta(-\sqrt{1-\xi^2}, \xi) \quad \text{where } |\xi| < 1$$

$$\varphi_{S,\theta}^{-1}(\xi) = T_\theta(\xi, -\sqrt{1-\xi^2}) \quad \text{where } |\xi| < 1$$

$$\varphi_{E,\theta}^{-1}(\xi) = T_\theta(\sqrt{1-\xi^2}, \xi) \quad \text{where } |\xi| < 1$$

For identical reasons as the case of $\theta = 0$ radians rotation, the atlas is smooth.

Now for an arbitrary rotation, the change of coordinates from rotated north and any of the standard hemispheres is C^∞ :

$$\varphi_N \circ \varphi_{N,\theta}^{-1}(\xi) = \cos(\theta)\xi - \sin(\theta)\sqrt{1-\xi^2}$$

with

$$\xi \in \begin{cases} (-\cos(\theta), 1) & \text{if } \theta \in (0, \pi) \\ (-1, \cos(\theta)) & \text{if } \theta \in (\pi, 2\pi) \end{cases}$$

$$\varphi_W \circ \varphi_{N,\theta}^{-1}(\xi) = \sin(\theta)\xi + \cos(\theta)\sqrt{1-\xi^2}$$

with

$$\xi \in \begin{cases} (-1, \sin(\theta)) & \text{if } \theta \in \left(0, \frac{\pi}{2}\right) \cup \left(\frac{3\pi}{2}, 2\pi\right) \\ (-\sin(\theta), 1) & \text{if } \theta \in \left(\frac{\pi}{2}, \pi\right) \cup \left(\pi, \frac{3\pi}{2}\right) \end{cases}$$

$$\varphi_S \circ \varphi_{N,\theta}^{-1}(\xi) = \cos(\theta)\xi - \sin(\theta)\sqrt{1-\xi^2}$$

with

$$\xi \in \begin{cases} (-1, \cos(\theta)) & \text{if } \theta \in (0, \pi) \\ (-\cos(\theta), 1) & \text{if } \theta \in (\pi, 2\pi) \end{cases}$$

$$\varphi_E \circ \varphi_{N,\theta}^{-1}(\xi) = \sin(\theta)\xi + \cos(\theta)\sqrt{1-\xi^2}$$

with

$$\xi \in \begin{cases} (\sin(\theta), 1) & \text{if } \theta \in \left(0, \frac{\pi}{2}\right) \cup \left(\frac{3\pi}{2}, 2\pi\right) \\ (-1, -\sin(\theta)) & \text{if } \theta \in \left(\frac{\pi}{2}, \pi\right) \cup \left(\pi, \frac{3\pi}{2}\right) \end{cases}$$

By combining this fact with the "middle chart" argument used in proof of Theorem 1, it follows that any other relative hemisphere will be smoothly compatible with all standard ones, proving that $\bigcup_{\theta\in[0,2\pi)} \mathcal{A}_\theta$ is a smooth atlas for the circle. Observe that this atlas just constructed has uncountable infinite size, yet it is nowhere close to being maximal!

2.3. Smooth manifold

We are now ready to explain what a smooth manifold is.

Definition 6 (**Smooth manifold**). A smooth manifold of dimension m is a pair $(\mathcal{M}, \mathcal{A})$ where $\mathcal{M}$ is a topological manifold of dimension m as in definition 3, and $\mathcal{A}$ is a maximal smooth atlas on $\mathcal{M}$ as in definition 5.

Furthermore, we note that we will be concerned with the study of smooth manifolds satisfying the Hausdorff condition (which says that every pair of points can be contained in two disjoint domains) and which have a finite atlas. The first condition is for theoretical reasons as the results we will obtain are not completely true otherwise and the finite atlas condition is for computing purposes, as computers can only store finitely many things.

Besides, our algorithm solves general differential equations on a smooth manifold contained in $\mathbb{R}^n$, something which turns out not to be a restrictive assumption as the Strong Whitney Embedding Theorem [10, thm. 6.18] guarantees that every smooth manifold can be smoothly embedded into $\mathbb{R}^{2n}$ and thus recasted into the format that our algorithm accepts, given in definition 7.

Definition 7 (Submanifold of $\mathbb{R}^n$ [16])**.** A set $\mathcal{M} \subset \mathbb{R}^n$ is a submanifold of $\mathbb{R}^n$ if for every $p \in \mathcal{M}$ there are open sets $U, V \subset \mathbb{R}^n$ with $p \in U$ and a continuously differentiable mapping with continuously differentiable inverse , $F : U \to V$, such that $F(U \cap \mathcal{M}) = \mathbb{R}^m \times \{0\}^{n-m}$. Here, m is called the dimension of the submanifold and $n - m$ is its codimension.

We notice that if $\mathcal{M}$ is a submanifold of $\mathbb{R}^n$ with dimension m then it is also a topological manifold of the same dimension in the subspace topology by taking as a chart for $p \in \mathcal{M}$, $\hat{U} := U \cap \mathcal{M}$ open in $\mathcal{M}$ and $\varphi := \pi_m \circ F$ with π_m the function that extracts the first m out of n components from a vector. Additionally, since F was a diffeomorphism, the Jacobian of the inverse coordinate map, $D\varphi^{-1} \in \mathbb{R}^{n \times m}$, exists on the open image of φ and has full rank.[6] These observations will become important in the convergence analysis of the algorithm in Chapter 3.

2.3.1. Manifold implementation in `Matlab`

With the previous concepts defined earlier, we used object-oriented programming in `Matlab` to represent a chart (U, φ) using the `Chart` class as seen in Code 2.1.

```
classdef Chart
    properties
        % requested:
        domain      % domain of coord function
        image       % image of coord function
        phi         % manifold -> coords
        phi_inv     % coords -> manifold
        n           % dimension of embedding space
        k           % dimension of manifold
        name        % name of the chart (optional)
        % internally inferred:
        coords_sym  % Symbolic variables for the coordinates
        Jacobian    % Jacobian of inverse map
    end
```

Code 2.1. MATLAB class definition for Chart (constructor only).

It consists of `domain`, a boolean function that returns true when an input point $p \in \mathbb{R}^n$ is in U and false otherwise.

In an analogous way, but in this case for inputs in $\mathbb{R}^k$, the function `image` determines whether a given point lies in the image of the chart. We assume that the user is able to specify what the image is, otherwise, determining the image of an arbitrary function is a

[6] Observe that the examples 2.3 and 2.4 exhibit this property.

nontrivial problem which can be numerically approximated using random sampling and convex hulls generation. [17].

The computer representations of φ and φ^{-1} are stored as anonymous function handles in `phi` and `phi_inv`. Furthermore, we use other secondary metadata like the embedding and submanifold dimension `n` and `k`, an optional name for the chart and a symbolic list of variables `coords_sym` used to compute $D\varphi^{-1}$, which is stored as `Jacobian` in symbolic form, after using Matlab's symbolic differentiation capabilities. The class constructor is shown in Code 2.2.

```
function obj = Chart(d,i, p, pinv, n, k, name)
    % Parameterized Constructor
    obj.domain = d;
    obj.image = i;
    obj.phi = p;
    obj.phi_inv = pinv;
    obj.n = n;
    obj.k = k;

    % Define symbolic variables for the k-dimensional
       coordinate space
    % 'c' is used as a prefix for coordinate variables, e.g.,
       c1, c2, ...
    obj.coords_sym = sym('c', [1 k]);

    % Assign the symbolic inverse map
    phi_inv_sym = obj.phi_inv(obj.coords_sym);
    obj.Jacobian = jacobian(phi_inv_sym, obj.coords_sym);

    if nargin < 6
        obj.name = 'Unnamed';
    else
        obj.name = name;
    end
end
```

Code 2.2. MATLAB Chart class constructor.

The `Chart` class is also equipped with basic functions given in Code 2.3 that simply use the input function handles and evaluate the Jacobian at a given coordinate $\vec{c} \in \varphi(U) \subset \mathbb{R}^n$. In addition, it contains the function `Projector` . Its presence will add flexibility in the range of applicability of our method, admitting cases where the differential equation is not expressed intrinsically using a vector field in the manifold $\mathcal{M}$ but as a global vector field in the ambient space $\mathbb{R}^n$. We will discuss its usage in Chapter 3, after we cover the necessary notions like the Tangent Bundle and vector fields in the remaining of this

chapter, which are key concepts used to give a general formulation of dynamical systems on smooth manifolds.

```
function in = inside_domain(obj,p)
    % Check if the point in Rn lies in the domain
    in = obj.domain(p);
end

function in = inside_image(obj,c)
    in = obj.image(c);
    % Check if the point in Rk lies in the image of phi
end

function c = Coords(obj,p) % finds coordinates on chart
    c = obj.phi(p);
end

function p = Manifold(obj,c) % evaluates phi_inv
    p = obj.phi_inv(c);
end

function Jval = JacobianAt(obj,c)
    % Evaluates Jacobian at coordinate c
    Jval = double(subs(obj.Jacobian,obj.coords_sym,c));
end

function A = Projector(obj,p) % useful to convert a vector
   field of Rn into a vector field on M
    c = obj.phi(p);
    J = obj.JacobianAt(c);
    A = inv(transpose(J)*J)*transpose(J);
end
```

Code 2.3. MATLAB Chart class methods.

Example 2.5 (**Declaring a manifold with the `Chart` class**). Declaring the sphere S^2 with the maximal atlas induced by the stereographic atlas from example 2.3 can be done directly as illustrated in Code 2.4.

```
%% Defining stereographic atlas on 2-Sphere:

% North chart (exclude north pole)
U = @(p) (norm(p)==1 && ~(p(1)==0 && p(2)==0 && p(3)==1));
Im = @(c) (0==0);
phi = @(p) [(2*p(1))/(1-p(3)); (2*p(2))/(1-p(3))];
phi_inv = @(c) [(4*c(1))/(norm(c)^2+4); (4*c(2))/(norm(c)^2+4)
    ; (norm(c)^2-4)/(norm(c)^2+4)];
C1 = Chart(U, Im, phi, phi_inv, 3, 2, 'NorthChart');

% South chart (exclude south pole)
U = @(p) (norm(p)==1 && ~(p(1)==0 && p(2)==0 && p(3)==-1));
Im = @(c) (0==0);
phi = @(p) [(2*p(1))/(p(3)+1); (2*p(2))/(p(3)+1)];
phi_inv = @(c) [(4*c(1))/(norm(c)^2+4); (4*c(2))/(norm(c)^2+4)
    ; (4-norm(c)^2)/(norm(c)^2+4)];
C2 = Chart(U, Im, phi, phi_inv, 3, 2, 'SouthChart');

A = [C1; C2];
```

Code 2.4. Defining stereographic atlas on 2-Sphere.

Observe that the functions are inputted as anonymous function handles, using the @ symbol and a single letter to denote the input column vector, whose entries are referenced using indexing like $p(1), p(2), ..., p(n)$ for the vector $p \in \mathbb{R}^n$. For the case of the stereographic projection, $\varphi(U) = \mathbb{R}^2$, so we use the constant true function (0==0) in place of the `image` attribute. After having defined the charts to be used, the atlas is simply a column vector with the charts stored in its entries.

2.4. Maps on smooth manifolds

Given two manifolds $\mathcal{M}$ and $\mathcal{N}$, since they are also topological spaces, a map $f : \mathcal{M} \to \mathcal{N}$ is continuous if it is continuous as a map between topological spaces (definition 1).

However, for differentiability, one tracks the rate at which f changes relative to a change in a particular direction. This is done locally by using a pair of charts but in a way that does not depend on the actual charts chosen. The figure 2.4 provides a visual context of the setup involved in the study of the smoothness of f.

Definition 8 (**Smooth map**). Let $\mathcal{M}, \mathcal{N}$ be two smooth manifolds of dimensions m, n respectively. Then, the map $f : \mathcal{M} \to \mathcal{N}$ is *smooth* when for each $p \in \mathcal{M}$ there exists (U, φ) in $\mathcal{M}$ with $p \in U$ and (V, ϕ) in $\mathcal{N}$ with $f(U) \subseteq V$ such that:

$$f_{\phi\varphi} : \varphi(U) \subset \mathbb{R}^m \to \phi(N) \subset \mathbb{R}^n, \quad f_{\phi\varphi} := \phi \circ f \circ \varphi^{-1}$$

is smooth in the usual sense [a]. When f is smooth and has a smooth inverse, it is called a *diffeomorphism*.

[a] In the case where $\mathcal{M} = \mathbb{R}^m, \mathcal{N} = \mathbb{R}^n$ submanifolds of $\mathbb{R}^m$ and $\mathbb{R}^n$ notice that this definition implies that $\hat{f} : f \circ \varphi^{-1}$ is smooth in the usual sense, as we have that $\hat{f} = \phi^{-1} \circ f_{\phi\varphi}$ and ϕ^{-1} is smooth from the definition of submanifold 7

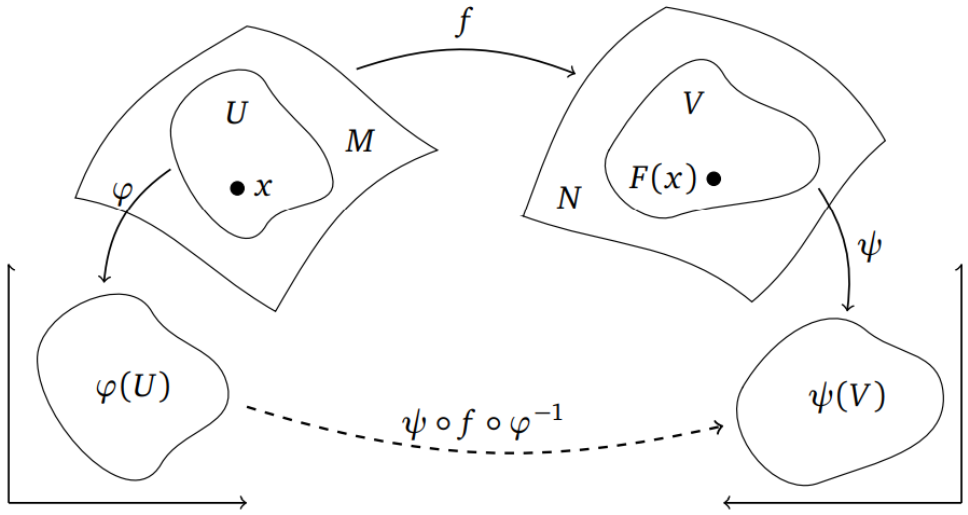


Fig. 2.4. The local coordinate representation of a map f between manifolds M, N using charts (U, φ) and (V, ψ) from Daniel Siemssen Phd thesis[18].

It is worth noting that the above definition is a generalization of the special case where $\mathcal{M}$ and $\mathcal{N}$ are open sets of $\mathbb{R}^m$ and $\mathbb{R}^n$. In that case, the charts are the entire spaces together with their identity maps. Another important remark is that smoothness in this sense implies continuity as one would expect, since f can be expressed locally on each open U as $f|_U = \phi^{-1} \circ f_{\phi\varphi} \circ \varphi$, which is a composition of continuous functions and hence continuous.

2.5. The tangent space

The tangent space and its elements, the tangent vectors, give a mathematical **coordinate independent** and **intrinsic description** of the different ways in which a point can "move" inside the manifold. **Coordinate independent** since a class of motion will be described without the need of referencing to a concrete chart velocity. It is only when we perform calculations that we may refer to tangent vectors using velocities in $\mathbb{R}^m$ from a concrete coordinate system. **Intrinsic** because it focuses on motions within $\mathcal{M}$, without assuming that $\mathcal{M}$ is embedded in $\mathbb{R}^n$. This last feature makes it a general definition that is meaningful for any kind of smooth manifold (even those which have infinite dimension which we have not covered in this thesis). Despite this generality, the motivation for the terminology used (tangent space, tangent vector) is attributed from the special case when $\mathcal{M} \subset \mathbb{R}^n$, in which case there is a direct correspondence with standard velocity vectors in $\mathbb{R}^n$ emanating from p and lying tangent to $\mathcal{M}$.

Suppose now that you wanted to classify the different ways to move away from $p \in \mathcal{M}$. As many curves can locally share the same motion, it is natural to consider all smooth curves initialized at $p \in \mathcal{M}$ and use a chart (U, φ) around $p \in \mathcal{M}$ to establish a relationship between them. In particular, we define a pair of curves as equivalent or constituting the same class of motion when their coordinate velocity on the chart matches.

Definition 9 (**Equivalent curves**)**.** Let $\mathcal{M}$ be a smooth manifold and $c, d : (-\varepsilon, \varepsilon) \to \mathcal{M}$ smooth curves initialized at $p \in \mathcal{M}$ that is, $c(0) = d(0) = p$. We say that c, d are equivalent if $(\varphi \circ c)'(0) = (\varphi \circ d)'(0)$ for some chart (U, φ) around $p \in \mathcal{M}$[a].[19]

[a]The symbol ' denotes differentiation with respect to each component

Despite defined in terms of coordinates, the latter relation from definition 9 is invariant under the chart used, due to the fundamental smoothness of the transition functions in the maximal atlas of $\mathcal{M}$. One can see this by taking another coordinate system (V, ϕ) containing p. Then, by the Chain Rule we have:

$$(\phi \circ c)'(0) = \left(D(\phi \circ \varphi^{-1})\right)_{\varphi(p)} \cdot (\varphi \circ c)'(0)$$

$$(\phi \circ d)'(0) = \left(D(\phi \circ \varphi^{-1})\right)_{\varphi(p)} \cdot (\varphi \circ d)'(0)$$

so if $(\varphi \circ c)'(0) = (\varphi \circ d)'(0)$ then, $(\phi \circ c)'(0) = (\phi \circ d)'(0)$.

It is true that the velocities change in another coordinates, in fact we see that they do so linearly through the Jacobian of the transition. However, the equality is still preserved.

Moreover, this relation is an **equivalence relation** among smooth curves at p, which allows one to partition the set of smooth curves initialized at p into a set of equivalence classes of curves known as the *tangent space of* $\mathcal{M}$ *at* p.

Definition 10 (**Tangent space**[19])**.** For a smooth manifold $\mathcal{M}$ and $p \in \mathcal{M}$ the tangent space to $\mathcal{M}$ at p is the set of equivalence classes of curves initialized at p:

$$T_p\mathcal{M} = \{[c]_p : c \in C_p^\infty(I, \mathcal{M})\}$$

where $[c]_p$ is an equivalence class of curves initialized at p and c is a representative of the class. [a]

[a] $C_p^\infty(I, \mathcal{M})$ denotes the set of smooth curves initialized at p.

2.5.1. Vector space structure

The set $T_p\mathcal{M}$ becomes a real vector space after introducing the suitable addition and scaling operations, given in definition 11, where sums of curves results in curves moving at the sum of velocities while scaling results in curves preserving direction but with scaled speed, similar to standard vectors in $\mathbb{R}^n$.

Definition 11 (**Vector space operations**)**.** Let $[c]_p, [d]_p \in T_p\mathcal{M}$ and $\lambda \in \mathbb{R}$. Define:

$$[c]_p + [d]_p := \left[\varphi^{-1}\left(\varphi(p) + ((\varphi \circ c)'(0) + (\varphi \circ d)'(0))t\right)\right]_p$$
$$\lambda \cdot [c]_p := \left[\varphi^{-1}\left(\varphi(p) + \lambda(\varphi \circ c)'(0)t\right)\right]_p$$

The operation of addition is independent on the choice of the representatives. This is because, using another pair of representatives, $\hat{c}, \hat{d}$:

$$\begin{aligned}
[\hat{c}]_p + [\hat{d}]_p &= \left[\varphi^{-1}(\varphi(p) + ((\varphi \circ \hat{c})'(0) + (\varphi \circ \hat{d})'(0))t)\right]_p \\
&= \left[\varphi^{-1}\left(\varphi(p) + ((\varphi \circ c)'(0) + (\varphi \circ d)'(0))t\right)\right]_p \qquad \text{since } c \equiv d \text{ and } d \equiv \hat{d}. \\
&= [c]_p + [d]_p.
\end{aligned}$$

So $[\hat{c}]_p + [\hat{d}]_p = [c]_p + [d]_p$. Also, if one considers another chart (V, ϕ) around p and writes

$$V := \frac{d}{dt}(\phi \circ \varphi^{-1}(\varphi(p) + ((\varphi \circ c)'(0) + (\varphi \circ d)'(0))t))\Big|_{t=0}$$

It holds that:

$$\begin{aligned}
V &= D(\phi \circ \varphi^{-1})((\varphi \circ c)'(0) + (\varphi \circ d)'(0)) && \text{(Chain Rule)} \\
&= D(\phi \circ \varphi^{-1})(\varphi \circ c)'(0) + D(\phi \circ \varphi^{-1})(\varphi \circ d)'(0) && \text{(Linearity)} \\
&= \frac{d}{dt}(\phi \circ c)\Big|_{t=0} + \frac{d}{dt}(\phi \circ d)\Big|_{t=0} && \text{(Chain Rule)} \\
&= \frac{d}{dt}(\phi \circ \phi^{-1}(\phi(p) + ((\phi \circ c)'(0) + (\phi \circ d)'(0))t))\Big|_{t=0},
\end{aligned}$$

and so

$$\left[\varphi^{-1}\left(\varphi(p) + ((\varphi \circ c)'(0) + (\varphi \circ d)'(0))t\right)\right]_p = \left[\phi^{-1}\left(\phi(p) + ((\phi \circ c)'(0) + (\phi \circ d)'(0))t\right)\right]_p .$$

This proves that + is well defined, invariant under the choice of representatives and the chart used. For the same reasons $\times$ is well defined too.

This justifies the name of **tangent vector** for the elements inside $T_p\mathcal{M}$. Its vector space dimension is the same as the dimension of $\mathcal{M}$, with a natural coordinate-dependent basis associated with a chart $(U, (\xi^i))$, where basis vectors correspond to uniform unit speed motions away from p through each of the coordinate lines of the chart:

$$B_\varphi = \{\frac{\partial}{\partial x^1}|_p := \left[\varphi^{-1}(\varphi(p) + e_1 t)\right], \ldots, \frac{\partial}{\partial x^n}|_p := \left[\varphi^{-1}(\varphi(p) + e_n t)\right]\}$$

To see why this is a basis of $T_p\mathcal{M}$ note that for a tangent vector $[c]_p \in T_p\mathcal{M}$, $(\varphi \circ c)'(0) = v_1\vec{e_1} + ... + v_n\vec{e_n}$ for a unique vector $\vec{v} = (v_1, ..., v_n)^T \in \mathbb{R}^n$ which implies $[c]_p = [\varphi^{-1}(\varphi(p) + (v_1\vec{e_1} + ...v_n\vec{e_n})t)] = v_1\frac{\partial}{\partial x^1}|_p + ... + v_n\frac{\partial}{\partial x^n}|_p$.

Using this basis, a tangent vector is represented by a particular coordinate velocity vector in $\mathbb{R}^m$ associated with the chart $(U, (\xi^i))$. Moreover, the change of basis with respect to another coordinate system is given by the Jacobian of the transition maps, which exists due to the smoothness of the manifold $\mathcal{M}$.

An important remark is that when $\mathcal{M}$ is a 2-dimensional submanifold in $\mathbb{R}^3$ then, there is a one-to-one correspondence between $T_p\mathcal{M}$ and velocity vectors of curves in $\mathbb{R}^3$ lying on the tangent plane at $p \in \mathcal{M}$ specified by $\mathrm{Im}(D\varphi^{-1})$, as figure 2.5 shows.

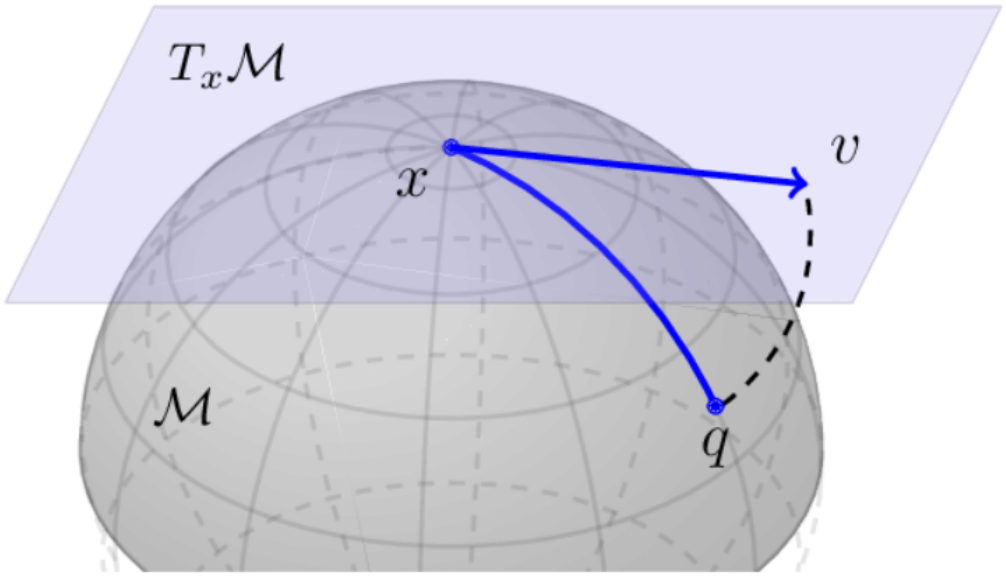


Fig. 2.5. The tangent space at the north pole of S^2 with the velocity vector v associated to the class of curves represented by q. Image from [20].

2.5.2. The push forward of a map

Given a smooth map between manifolds $f : \mathcal{M} \to \mathcal{N}$, the push forward of f at $p \in \mathcal{M}$, denoted df_p, is a linear map between tangent vectors in $p \in \mathcal{M}$ and tangent vectors at $f(p) \in \mathcal{N}$, that is: $df_p : T_p\mathcal{M} \to T_{f(p)}\mathcal{N}$, where $df_p([c]_p) := [f \circ c]_{f(p)}$. When a pair of coordinates $(U, \varphi), (V, \phi)$ over p and $f(p)$ are chosen, the local representation of the push forward is given by the Jacobian of the coordinate version of f on those charts, $f_{\phi\varphi} = \phi \circ f \circ \varphi^{-1}$. That is, the respective coordinate version of the differential becomes: $\hat{df}_p : \mathbb{R}^m \to \mathbb{R}^n$ with $\hat{df}_p(\vec{v}) = Df_{\phi\varphi}\vec{v}$. Under these representation of the tangent spaces using the chart basis , the differential is seen to be pushing velocities of curves at $p \in \mathcal{M}$ into velocities of curves at $f(p) \in \mathcal{N}$.

Observe that in the special case where $\mathcal{M} = \mathbb{R}^m$ and $\mathcal{N} = \mathbb{R}^n$ then, the differential of f is equal to the standard derivative or Jacobian of f.

2.6. The Tangent Bundle

After presenting $T_p\mathcal{M}$ and the fact that it is a vector space of dimension equal to that of $\mathcal{M}$, there is another important smooth manifold that can be obtained from these two spaces, known as the Tangent Bundle of $\mathcal{M}$, denoted $T\mathcal{M}$, which is the collection of all the tangent spaces at every point in the manifold. It has a natural topology, where an open set is the disjoint union across the tangent spaces over a common domain $U \subset \mathcal{M}$. More precisely, we define an open set of $T\mathcal{M}$ to be:

$$TU = \{[c]_p \mid p \in U, [c]_p \in T_p\mathcal{M}\}$$

where U is any chart from the maximal atlas of $\mathcal{M}$.

Theorem 2 (**The Tangent Bundle is a smooth manifold**)**.** Let $\mathcal{M}$ be a smooth manifold of dimension m. The Tangent Bundle of $\mathcal{M}$, $T\mathcal{M} = \{[c]_p \mid [c]_p \in T_p\mathcal{M}, p \in \mathcal{M}\}$ is a smooth manifold of dimension $2m$.

Proof. For each of open set TU in $T\mathcal{M}$, we can identify each tangent vector by its tangency point and velocity in that chart. In other words, the pair (TU, T_φ) with $T_\varphi : TU \to \mathbb{R}^{2m}$ defined as $T_\varphi([c]_p) := (\varphi(p), (\varphi \circ c)'(0))$ is a chart on $T\mathcal{M}$. From the definition of $\mathcal{M}$, this implies that for every $[c]_p \in T\mathcal{M}$ there is some open TU containing it which is homeomorphic to an open set in $\mathbb{R}^{2m}$. Thus, $T\mathcal{M}$ is a topological manifold. Moreover, for every pair of charts on $(U, (\xi^i)), (V, (\phi^j))$ on $\mathcal{M}$ whose domain overlap we see that:

$$(T_\phi \circ T_\varphi^{-1})(\vec{\xi}, \vec{v}) = ((\phi \circ \varphi^{-1})(\vec{\xi}), D(\phi \circ \varphi^{-1})\vec{v})$$

which is smooth by definition of $\mathcal{M}$ and hence $T\mathcal{M}$ is smooth of dimension $2m$. □

Example 2.6. In the case of $\mathcal{M} = \mathbb{R}^n$, using the canonical representation from the chart $(\mathbb{R}^n, \varphi = id)$, then, $T\mathbb{R}^n$ becomes $\mathbb{R}^{2n}$.

A pictorial illustration of TS^1 is given in figure 2.6, where every tangent line to the circle, represents a particular tangent space.

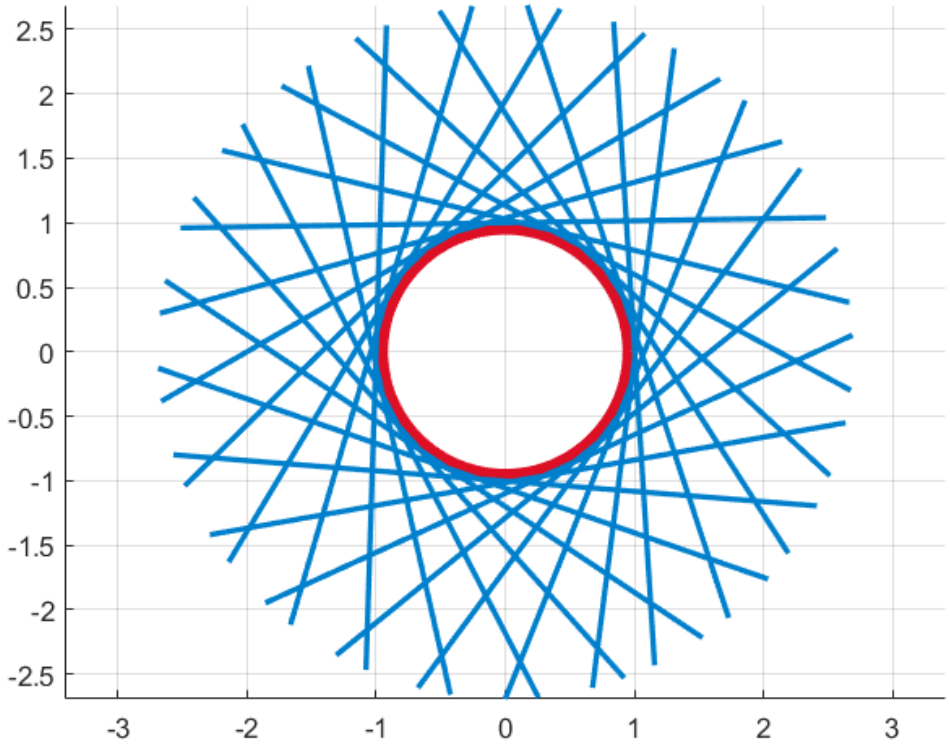


Fig. 2.6. The tangent bundle to the circle, TS^1 can be visualized by drawing every tangent line to the circle. To each point there corresponds a tangent line, corresponding to the one dimensional tangent space of the circle at that point.

2.7. Smooth vector fields

A smooth vector field is a smooth map which associates a point in a manifold with a tangent vector from the tangent space at that point. It is a central concept in mathematical modeling, used to describe any system which undergoes change in some form, like the motion of Earth around the Sun, the life-death cycle of a pair of species competing for the same resources or the spread of a contagious disease.

Definition 12 (Smooth vector field)**.** Let $\mathcal{M}$ be a smooth manifold of dimension m. A **smooth vector field** on $\mathcal{M}$ is a smooth map

$$V : \mathcal{M} \to T\mathcal{M}, \quad p \mapsto V(p)$$

with $V(p) \in T_p\mathcal{M}$ for each $p \in \mathcal{M}$.

Using coordinates $(U, \varphi = (\xi^1, \ldots, \xi^m))$, the restriction of the vector field to a domain $U \subset \mathcal{M}$ can be written in terms of the chart basis vectors:

$$V|_U(p) = \sum_{i=1}^{m} V^i(p) \frac{\partial}{\partial \xi^i}|_p,$$

where the V^i's are the components of the vector field in the chart (U, φ), which defines the velocity at each point in U through the coordinate system $\varphi = (\xi^1 ..., \xi^m)$. We denote its coordinate representation by $V_\varphi : \varphi(U) \to \mathbb{R}^m$.

From the definition of a smooth map (definition 8), it follows that $V|_U$ is smooth when V_φ is smooth and V is smooth when its restriction to any domain in the maximal atlas of $\mathcal{M}$ is smooth.

Example 2.7. $V(p) = \theta(p) \frac{d}{d\theta}|_p$ is a smooth vector field on $S^1 \setminus (1, 0)$ but its extension to the full circle given by setting $V(p) = (\hat{\theta}(p) + \frac{3\pi}{2} \mod 2\pi) \frac{d}{d\hat{\theta}}|_p$ (where $\hat{\theta}$ is the counter-clockwise angle coordinate measured from $(0, -1)$) is not smooth on S^1 . This is because the field fails to be continuous at $(1, 0)$ because $\widetilde{V}(\hat{\theta}) = \hat{\theta} + \frac{3\pi}{2} \mod 2\pi, \hat{\theta} \in (0, 2\pi)$ has a jump discontinuity at $\frac{\pi}{2}$.

2.7.1. Integral curve of a vector field

Fundamentally, given a smooth vector field V on $\mathcal{M}$ with an initial position $p \in \mathcal{M}$, we are interested in knowing what curve (if any) moves according to the laws of motion imposed by V. More specifically, a curve $c : I \subset \mathbb{R} \to \mathcal{M}$ with $c(0) = p_0$ whose velocity vector after time t satisfies [10]

$$dc_t \left(\frac{d}{dt} \right) = V(c(t)). \tag{2.1}$$

is known as an **integral curve** of the vector field V starting at p_0.

This integral curve question can be rephrased locally, as for sufficiently small t, the curve c remains inside the domain of the chart $(U, \varphi = (\xi^i))$ so the coordinate independent condition in equation (2.1) is equivalent to the following equation of tangent vectors, expressed in terms of the coordinate representation of the curve, $\varphi \circ c \equiv \vec{z}$:

$$\sum_{i=1}^{m} z_i'(t) \frac{\partial}{\partial \xi_i}\bigg|_{c(t)} = \sum_{i=1}^{m} V_\varphi^i(z_1(t), \ldots, z_m(t)) \frac{\partial}{\partial \xi_i}\bigg|_{c(t)}$$

Which by definition of a basis, it is equivalent to the following system of ODEs in the coordinate space $\varphi(U) \subset \mathbb{R}^m$ of the chart :

$$\begin{cases} \mathbf{z}'(t) = V_\varphi(\mathbf{z}(t)), \\ \mathbf{z}(0) = \varphi(p_0) \end{cases}$$

Where $V_\varphi : \varphi(U) \subset \mathbb{R}^n \to \mathbb{R}^n$ is a smooth vector-valued function from the smoothness of V.

Therefore, finding an integral curve during a small enough interval of time is equivalent to solving an autonomous ODE system of m equations, on a chart around the initial position. Moreover, as V_φ is smooth, it verifies the hypothesis of the elementary result of autonomous ODEs inside $\mathbb{R}^n$:

Theorem 3 (Classical results of autonomous ODES in $\mathbb{R}^n$ [10])**.** Suppose $U \subseteq \mathbb{R}^n$ is open, and $V : U \to \mathbb{R}^n$ is a smooth vector-valued function. Consider the initial value problem

$$\dot{y}^i(t) = V^i(y^1(t), \ldots, y^n(t)), \qquad i = 1, \ldots, n, \tag{2.2}$$

$$y^i(t_0) = c^i, \qquad i = 1, \ldots, n, \tag{2.3}$$

for arbitrary $t_0 \in \mathbb{R}$ and $c = (c^1, \ldots, c^n) \in U$.

1. **Existence:** For any $t_0 \in \mathbb{R}$ and $x_0 \in U$, there exist an open interval J_0 containing t_0 and an open subset $U_0 \subseteq U$ containing x_0 such that for each $c \in U_0$, there is a C^1 map $y : J_0 \to U$ that solves (2.2)–(2.3).

2. **Uniqueness:** Any two differentiable solutions to (2.2)–(2.3) agree on their common domain.

3. **Smoothness:** Let J_0 and U_0 be as in **(a)**, and let $\theta : J_0 \times U_0 \to U$ be the map defined by $\theta(t, x) = y(t)$, where $y : J_0 \to U$ is the unique solution to (2.2) with initial condition $y(t_0) = x$. Then θ is smooth.

Theorem 3 guarantees the existences and uniqueness of integral curves of smooth vector fields on any smooth manifold.

Example 2.8. An interesting illustration of the integral curves of a vector field on an non-trivial smooth manifold comes from studying the dynamics on the torus $\mathbb{T}^2 = S^1 \times S^1$, which is a product of two circles and thus a manifold (see page 137 of [19]). Specifically, consider the vector field

$$V(p) = \omega_1 \frac{\partial}{\partial\theta}|_p + \omega_2 \frac{\partial}{\partial\phi}|_p$$

for constants $\omega_1, \omega_2 > 0$. Its integral curve at $p(\theta_0, \phi_0)$ is given by:

$$c(t) \;=\; ((R + r\cos(\phi_0 + w_2 t))\cos(\theta_0 + w_1 t),\ (R + r\cos(\phi_0 + w_2 t))\sin(\theta_0 + w_1 t),\ r\sin(\phi_0 + w_2 t))\,.$$

As it is depicted in figure 2.7, the resulting dynamics depends on the ratio of the velocities: when $\frac{\omega_1}{\omega_2} \in \mathbb{Q}$ the integral curves are periodic closed curves getting tangled around the torus, otherwise, the integral curves lose their periodicity and are dense in the torus, getting arbitrarily close to every point in it; a behavior known as quasiperiodic motion [21].

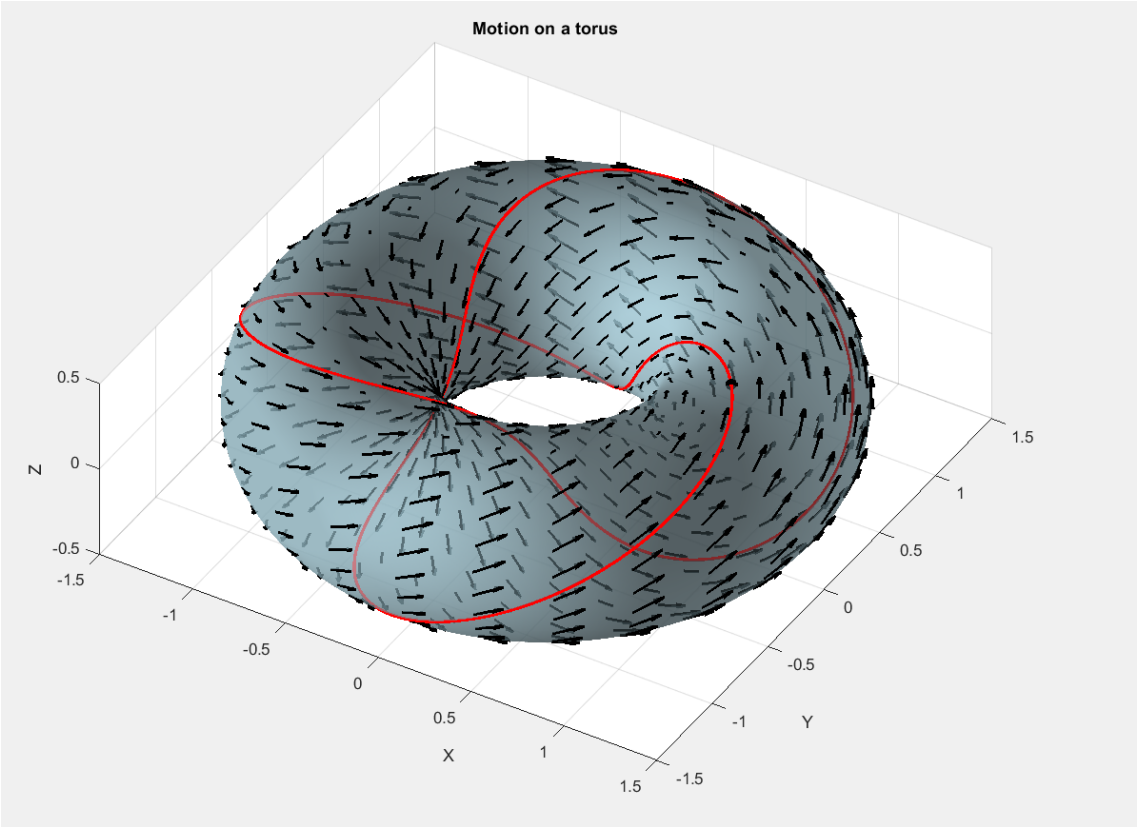


Fig. 2.7. Illustration of the vector field $V(p) = 2\dfrac{\partial}{\partial\theta} + 3\dfrac{\partial}{\partial\phi}$ on $\mathbb{T}^2$ using Matlab. Different tangent vectors are plotted together with a closed integral curve starting at the point $(0, 1, 0.5)$.

2.7.2. Flows of vector fields

A convenient way to encode the dynamics generated by a vector field V on a smooth manifold $\mathcal{M}$ is given by the flow map $\phi : D \to \mathcal{M}$ defined over $D = \{(t, po) \in \mathbb{R} \times \mathcal{M} : t \in I(p)\}$, where $\phi_t(p) : I(p) \to \mathcal{M}$ corresponds to the integral curve starting at p, existing throughout some maximal interval $I(p)$.

We remark that vector fields whose integral curves exist for all $\mathbb{R}$ are said to be **complete** and are special cases of smooth vector fields. For instance, $x^2\dfrac{\partial}{\partial x}$ is not complete on $\mathbb{R}$ nor $\dfrac{\partial}{\partial x}$ on $(0, 1)$.

Example 2.9 (Logistic Population Growth)**.** A famous model proposed by Verhulst in 1838 to study the poulation in France is given by:

$$\begin{cases} P'(t) = \lambda P(t)\left(1 - \dfrac{P(t)}{K}\right), \\ P(0) = p_0. \end{cases}$$

where the term $\lambda\left(1 - \frac{P}{K}\right)$ corresponds to the average birth rate per capita, which is maximal with rate λ and varies linearly based on the current population size P, decreasing as the population approaches some carrying capacity K.

The state space of this model is the submanifold $\mathbb{R}_+ \cup \{0\}$, as population size is a nonnegative quantity. Furthermore, the associated vector field is given by:

$$V(x) = \lambda\left(x - \frac{x^2}{K}\right)\frac{\partial}{\partial x}|_x$$

which is smooth. The problem can be solved analytically for an arbitrary initial population p_0 via separation of variables and leads to the **smooth** flow :

$$\phi(p_0, t) = \frac{K p_0 e^{\lambda t}}{K + p_0(e^{\lambda t} - 1)},$$

defined on:

$$D = \left\{(p_0, t) \,\middle|\, p_0 \geq 0, \quad t \in \begin{cases} \mathbb{R}, & p_0 \leq K \\ (\lambda^{-1}\ln(1 - K/p_0), +\infty), & p_0 > K \end{cases}\right\}$$

The graph of the flow when $\lambda = 1, K = 10$ is given in figure 2.8, superposed by some of its vertical sections, which can be seen as the graphs of different integral curves on the $t-z$ plane, revealing how all initial populations approach the carrying capacity K, as $t \to +\infty$.

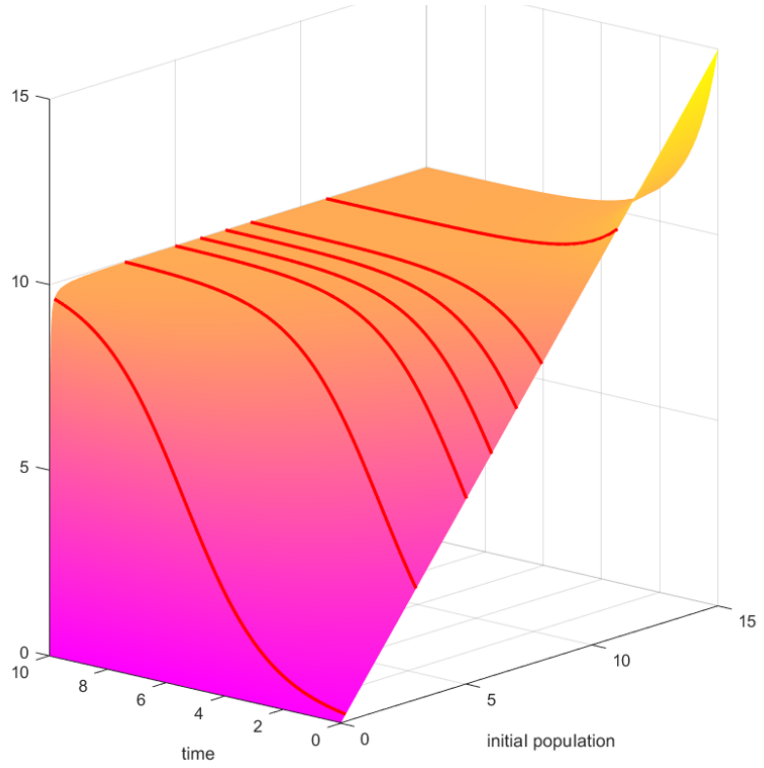


Fig. 2.8. Flow map of the population growth model, with a carrying capacity $K = 10$ and birth rate per capita $\lambda = 1$.

In general, we know by the following theorem that the flow generated by a smooth vector field is always smooth (in the sense of definition 8) implying that solutions of autonomous differential equation on a smooth manifold vary proportionally with respect to changes on their initial conditions.

Theorem 4 (Smoothness of the flow map)**.** The flow map $\phi : D \subset \mathcal{M} \times \mathbb{R} \to \mathcal{M}$ induced by a smooth vector field $V : \mathcal{M} \to T\mathcal{M}$ on a submanifold $\mathcal{M} \subset \mathbb{R}^n$ is smooth.

Proof. For some initial "space-time" point $(p_0, 0) \in D$, we know by the classical result in $\mathbb{R}^n$ that the coordinate problem on some chart (U, φ) containing p_0 has solution over a coordinate neighborhood of $\varphi(p_0) = z_0$ and the flow $\varphi \circ \phi_t \circ \varphi^{-1}$ is smooth on a neighborhood of $(z_0, 0)$ (see conclusion 3 of Theorem 3), proving smoothness of ϕ at $(p_0, 0)$. From this fact , we extend the proof of smoothness at a more general point $(p, T) \in D$ by dividing $[0, T]$ into finitely many sub-intervals $[0, t_1], ..., [t_{n-1}, t_n = T] \subset [0, T]$ during which the solution is contained on a single chart. This allows us to express [7] the flow map as a finite composition of itself over each of this smaller time stamps: $\phi(p, t) = (\phi_{t-t_{n-1}} \circ ... \circ \phi_{t_1})(p)$. Therefore, from the initial argument, we see that $\phi_{t_i - t_{i-1}}$ is smooth at $\phi_{t_{i-1}}(p)$ and $\phi_{t-t_{n-1}}$ is smooth at (p, T). □

[7] This is a crucial step in the proof, where we use that $\phi_{a+b} = \phi_a \circ \phi_b$ for $a, b \in \mathbb{R}^n$- a property enjoyed by integral curves of time independent vector fields. This would not be possible for time dependent vector fields, which we are not concerned with.

3. THE COORDINATE-ADAPTIVE ALGORITHM

3.1. Overview

After understanding the mathematical details outlined in the previous chapter, we are now ready to explain how the integrator works. The main idea of the algorithm is to use the observation that integrating an autonomous vector field on a smooth manifold reduces to solving a chain of ODEs[8], describing the dynamics locally in each chart. Based on this principle and from a given starting point, the integrator chooses an appropriate chart covering it, takes steps in the respective coordinate space using the coordinate vector field and obtains manifold estimates from these coordinate estimates, thus guaranteeing an estimated trajectory inside of the manifold. Naturally, the solution will probably move across different regions not covered by a single chart so at some time the coordinate estimates will step off the coordinate space. For this reason, when an overshot situation of this kind is faced, the algorithm has to dynamically assess the situation by adapting to the solution curve using another coordinate system from which the problem can be translated, before resuming the integration on the new coordinates. The image in figure 3 represents in a pictorial way the main workflow of the algorithm.

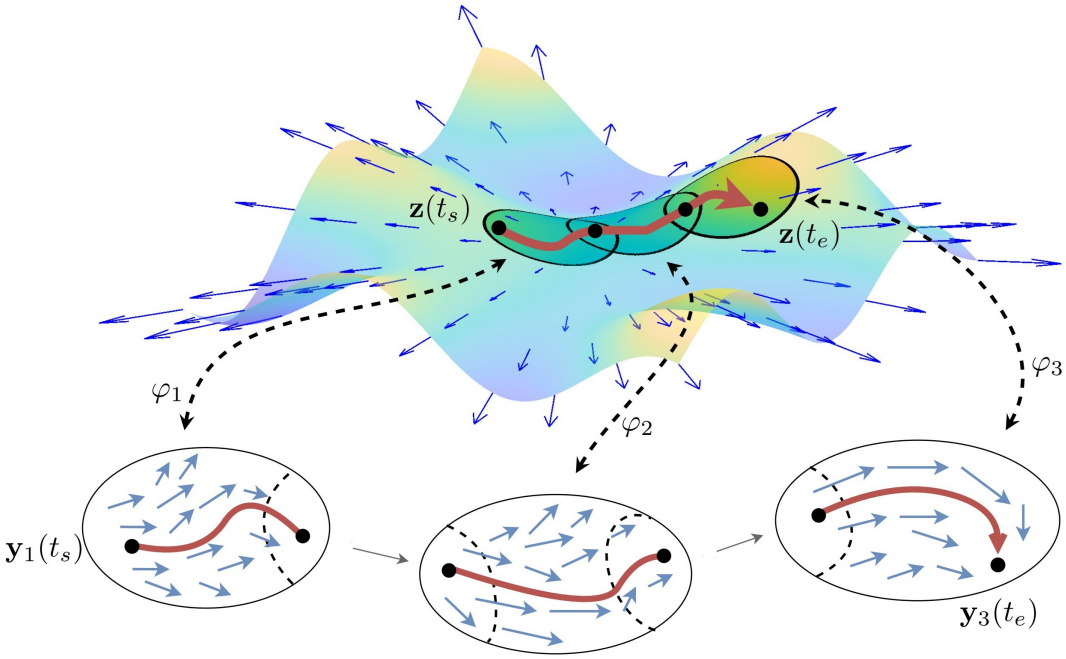


Fig. 3.1. Illustration of the Coordinate-Adaptive algorithm [7]. Each chart induces some vector field on an open subset of $\mathbb{R}^n$. Coordinates near the boundary of the chart image describe points inside overlapping regions.

[8]This is thanks to the locally euclidean property of manifolds, which we have already discussed.

We will explain the implementation of the Coordinate-Adaptive algorithm by identifying the key actions that the integrator must carry out. First, it must have a way of understanding which charts are currently useful, that is, those from the atlas which contain the current point to take the next step from. To do this, we use the function `GetCharts(A,p)`, given in Code 3.1, which returns the list of indexes corresponding to charts in the input atlas `A` whose domain contains the input point `p` in the manifold. It does this by using the `inside_domain` function from each `Chart` class object inside A.

```
function list = GetCharts(A,p)
    list = [];
    for i = 1:size(A,1)
        if A(i,1).inside_domain(p) == 1
            list = [list i];
        end
    end
end
```

Code 3.1. GetCharts function

After picking a valid chart surrounding the current point, the integrator constructs the associated local ODE using the vector field. In order to add flexibility to the integrator, we allow the user to provide the vector field indirectly, by means of a vector field on the ambient space which is tangent to the submanifold. That is, the user may only have at its disposal a map $f : \mathbb{R}^n \to \mathbb{R}^n$ whose integral curves on $\mathcal{M}$ remain in $\mathcal{M}$.

For this situation, we program the necessary logic to obtain the corresponding vector field on $\mathcal{M}$ in terms of charts. The way in which we do this conversion comes from reasoning as follows [16]: If c is an integral curve starting at $p \in \mathcal{M}$ and (U, φ) covers it, then, by the Chain Rule:

$$c'(0) = D\varphi^{-1}(\varphi \circ c)'(0) = f(p) \in \mathbb{R}^n.$$

This is an overdetermined system of n equations 3.1 on $\dim \mathcal{M} = m$ unknowns which has a unique solution [9] corresponding to the coordinate velocity $(\varphi \circ c)'(0) \in \mathbb{R}^m$. After rearrangement of the equation we can see that $V_\varphi(\vec{\xi}) = (D\varphi^{-1})^+ f_\varphi(\vec{\xi})$[10].

The internal `Chart` Class function `Projector(p)` from Code 2.3 computes the matrix $(D\varphi^{-1})^+ \in \mathbb{R}^{m\times n}$, known as the pseudo-inverse of φ^{-1}.

We remark that this step is not necessary if the user knows the explicit chart representation of the vector field, in which case the algorithm simply stores this information.

For example, an arbitrary vector field $f : \mathbb{R}^3 \to \mathbb{R}^3$ that is tangent to the sphere S^2 can be directly expressed in terms of the 6 chart-atlas from example (2.3) as the solutions to the system given in 3.1 are verified to be given by all components of f excluding the

[9] This is due to the fact that f is tangent to $\mathcal{M}$. Indeed, f is tangent to $\mathcal{M} \iff f(p) \in \mathrm{Im} D\varphi^{-1}(\varphi(p)))$

[10] $(D\varphi^{-1})^+ = (D\varphi^{-T} D\varphi^{-1})^{-1}(D\varphi^{-1})^T \in \mathbb{R}^{m\times n}$

one corresponding to the slicing plane. In the algorithm file, we would then fill up the function, `V_chart(A,chart_index,coord)` as given in Code 3.2 below.

```
function v = V_chart(A, chart_index, coord)
    % User can define the vector field through the charts of
       the input atlas A here (intrinsic form)
    vect = f(A(chart_index).Manifold(coord));

    if chart_index == 1 || chart_index == 2
        v = [vect(1); vect(2)];
    end
    if chart_index == 3 || chart_index == 4
        v = [vect(1); vect(3)];
    end
    if chart_index == 5 || chart_index == 6
        v = [vect(2); vect(3)];
    end
end
```

Code 3.2. The V_chart function stores the vector field definition in terms of the charts.

Using the coordinate velocity on the chosen chart, the integrator is then free to apply any one step method. In the case of no overshot then the integrator will take the coordinate estimate and invert it to obtain the respective point estimate on the manifold. In contrary, if an overshot occurs, the integrator quickly transforms the local equations into the equivalent ones on a new chart that also covers the current estimate. This is accomplished through the use of the function[11] `TranslateVelocity(A,i,j,c_i,v_i)` (see Code 3.3) which takes a tangent vector v_i on chart i at the point with coordinate c_i and outputs the equivalent representation v_j on chart j, exploiting Matlab symbolic differentiation capabilities.

```
function vj = TranslateVelocity(A, i, j, c_i, vi)
    % Obtains coordinate in chart j for point with coord i in
       chart i
    % Expresses a tangent vector from basis of chart i with
       respect to basis
    % of chart j.
    coords_i = sym('c', [1 2]);
    transition = A(j).Coords(A(i).Manifold(coords_i));
    J = jacobian(transition, coords_i);
    vj = double(subs(J, coords_i, c_i)) * vi;
    % Change of basis is the Jacobian of transition at c_i
end
```

[11] The i and j correspond to the indexes of the charts from the input atlas `A`.

Code 3.3. TranslateVelocity function: transforms a tangent vector from chart i to chart j.

Overall, the algorithm execute these taks iteratively, producing manifold estimates every $h > 0$ units of time, until reaching a final simulation time T. The final way in which these tasks are assembled is presented in the following pseudo-code:

Algorithm 1 The Coordinate-Adaptive Algorithm

Require: Smooth atlas $\mathcal{A}$, initial position $p_0 \in \mathcal{M}$, vector field $V : \mathcal{M} \to T\mathcal{M}$, time step $h > 0$, final time T

Ensure: Sequence $\{p_n\} \subset \mathcal{M}$ with $n \cdot h \leq T$.

$t \leftarrow 0$
$p \leftarrow p_0$
for $n = 1 \dots T/h$ **do**
    Choose $(U_i, \varphi_i) \in \text{AvailableCharts}(p) \subset \mathcal{A}$
    $A \leftarrow \mathcal{A} \setminus \{(U_i, \varphi_i)\}$
    $C_{n-1} \leftarrow \varphi_i(p_{n-1})$
    $C_n \leftarrow OneStep(C_{n-1}, V_{\varphi_i}(C_{n-1}))$
    $x \leftarrow i$
    while $C_n \notin \varphi_x(U_x)$ **do**
        if $A \neq \emptyset$ **then**
            Pick $(U_j, \varphi_j) \in A$
            $A \leftarrow A \setminus \{(U_j, \varphi_j)\}$
            $V_{\varphi_j} \leftarrow D(\varphi_j \circ \varphi_i^{-1})(C_{n-1}) \cdot V_{\varphi_i}(C_{n-1})$
            $C_{n-1} \leftarrow \varphi_j(p_{n-1})$
            $C_n \leftarrow OneStep(C_{n-1}, V_{\varphi_j}(C_{n-1}))$
            $x \leftarrow j$
        else
            repeat with smaller h
        end if
        $p_n \leftarrow \varphi^{-1}(C_n)$
    end while
end for

Notice that this pseudo-code uses the Euler Method for taking a step in the coordinate space of the chosen chart. However, any other preferred one-step method could be applied. As remarked in the introduction, all the source code of the integrator is available in a public Github repository [22] and appended at the end of this document.

3.2. Choosing the best chart

An interesting question that has arisen throughout the development of the integrator is what chart to use when there is more than one possible choice. There are occasions, where asking this question is not just a matter of optimization or preference but a necessity to be considered, as we observed that there exists the possibility that the chart used and the vector field are very incompatible with each other, leading to dangerous instabilities in the integration that cannot be numerically handled. More precisely, a chart that maps a proper subset of the manifold onto the whole plane, $\mathbb{R}^n$ (like a stereographic chart, which covers everything but a pole) may introduce local coordinate vector fields whose solutions blow up in finite time, when the integral curves of the vector field described only travels around the chart domain temporarily. Observe that that this kind of situation can occur even when the vector field is smooth and complete since the source of this singular behavior comes as an artifact introduced by the coordinate system used to describe the vector field.

To illustrate it with a concrete example, consider the vector field on the circle corresponding to uniform counterclockwise rotation and south pole as initial condition. Then, using the stereographic atlas given in example 2.4, the integrator will start integrating the local system on the chart $(S^1 \setminus N, \varphi_1)$:

$$\begin{cases} x'(t) = \dfrac{x(t)^2 + 4}{4}, \\ x(0) = 0 \end{cases}$$

but note that the analytic solution is $x(t) = 2\tan\left(\frac{t}{2}\right)$ and so $\lim_{t\to\pi} x(t) = +\infty$, hence the coordinate problem has a finite time blow-up.

From the numerical standpoint, this translates to very inaccurate predictions as velocity grows arbitrarily fast, demanding an infeasible small timestep to keep the error from quickly accumulating. Actually, no matter how small the time step used is, at some point the velocity becomes too large for the computer to handle, causing integration crashes due to overflow errors[12]. For this concrete example, a solution is to obtain estimates using the chart whose source point at infinity is the furthest away, that is, if the current estimate is on the lower hemisphere use φ_1 otherwise, use φ_2.

We devised a possible choosing criteria that could be used by the integrator to avoid this issue in a general setting, using a heuristic argument to minimize the local error in each chart, which we see that depends on the distortion of the coordinate map and the local properties of the vector field under question. We reason as follows.

Let $p_n \in \mathcal{M}$ be the estimate at the n-th step and consider the local error made by the Coordinate-Adaptive method with Euler as underlying one-step and a chart (U_i, φ_i),

[12] In Matlab the program does not literally crash but the estimated coordinates are assigned the value NaN but in any case, the method stops giving useful estimations.

defined as

$$d_{n,i} = p_{n+1,i} - c(h),$$

where c is the exact solution starting at $p_n \in \mathcal{M}$, that is,

$$c'(t) = V(c(t)), \quad c(0) = p_n.$$

Then, let $z_{n,i} = \varphi_i(p_{n,i})$, and so

$$p_{n+1,i} = \varphi_i^{-1}(z_{n+1,i}) = \varphi_i^{-1}(z_{n,i} + V_{\varphi_i}(z_{n,i})h).$$

The last equality being true because we are using the Euler method in the coordinate space, although the argument works for any other one-step method.

Now, because V is a smooth vector field on $\mathcal{M} \subset \mathbb{R}^n$ then, V_{φ_i} and φ_i^{-1} are smooth, so we can expand $\varphi_i \circ c \equiv z_i$ about $t = 0$ and φ_i^{-1} about $z_{n+1,i}$ to get:

$$\begin{aligned}
|d_{n,i}| &= \left|\varphi_i^{-1}(z_{n+1,i}) - \varphi_i^{-1}\Big(z_{n,i} + V_{\varphi_i}(z_{n,i})h + V'_{\varphi_i}(z_{n,i})V_{\varphi_i}(z_{n,i})\tfrac{h^2}{2} + O(h^3)\Big)\right| \\
&= \left|\varphi_i^{-1}(z_{n+1,i}) - \varphi_i^{-1}\Big(z_{n+1,i} + \tfrac{1}{2}V'_{\varphi_i}(z_{n,i})V_{\varphi_i}(z_{n,i})\tfrac{h^2}{2} + O(h^3)\Big)\right| \\
&= \left|D\varphi_i^{-1}(z_{n+1,i})\, V'_{\varphi_i}(z_{n,i})V_{\varphi_i}(z_{n,i})\frac{h^2}{2} + O(h^3)\right|
\end{aligned}$$

So,

$$d_{h,i} \simeq \Big|\underbrace{D\varphi_i^{-1}(z_{n+1,i})\, DV_{\varphi_i}(z_{n,i})V_{\varphi_i}(z_{n,i})}_{C_{i,n}}\Big|\frac{h^2}{2}.$$

Then, the choosing criteria is to obtain the manifold estimate using the system $\varphi_\star$ with

$$\varphi_\star := \underset{i\,:\,p_n \in \mathrm{Dom}(\varphi_i)}{\mathrm{Argmin}}\ C_{i,n}.$$

In the case of the stereographic example given above, this method is equivalent to choosing the chart whose missing pole is further away from the current point. Although it is a theoretically plausible approach, it seems to be too inefficient when there is a lot of overlapping charts in the input atlas as it requires applying the one-step method to each one available. Also, it requires knowing or at least approximating the jacobians of the chart and the coordinate representation of the vector field which could incur relevant evaluation and storage costs.

3.3. Convergence Analysis

In this section we present the necessary theorems and proofs to study the convergence of the Coordinate-Adaptive algorithm. These results are adapted from [16] which is the central reference for this chapter.

The fact that smooth vector fields induce smooth flows as we saw from Proposition 4 of section 2, can be used to derive a particularly relevant lemma for the proof of convergence of our algorithm. In particular, we are able to find a quantitative upper bound on the rate at which initial perturbations propagate over an interval of time.

Lemma 1. [**Controlling error growth**] Given a smooth vector field $V : \mathcal{M} \to TM$ on a submanifold $\mathcal{M} \subset \mathbb{R}^n$ whose integral curve at $p_0 \in \mathcal{M}$ exists on the interval $[0, T]$ there exists $\delta, C > 0$ such that:

$$\|\phi_{t-\tau}(p_1) - \phi_{t-\tau}(p_2)\| \leq C \cdot \|p_1 - p_2\| \quad \text{for } 0 \leq \tau \leq t \leq T$$

and

$$p_1, p_2 \in K_\tau(\delta) = \{p \in \mathcal{M} : \|p - \phi_\tau(p_0)\| \leq \delta\}.$$

The above lemma tells us that points dropped off close enough to the solution curve starting at p_0 will not deviate by more than a factor of C, relative to their initial separation. But more generally, it addresses the propagated separation for pairs of points dropped around any point near the image of the curve throughout the interval $[0, T]$. This is why the parameter τ is used, as a way of fixing the initial position from which to measure the growth of perturbations from, in the next $t - \tau$ units of time.

Proof. Using the same interval splitting argument given in the previous proof and by the openness of the chart domains, we know that there exists $\delta_0 > 0$ such that $K_\tau(\delta_0) \subset U_i$ for all $\tau \in [t_i, t_{i+1}]$ and $i = 0, ...n - 1$. In particular, we may take $\delta \leq \delta_0$ small enough to guarantee that $\phi_{t-\tau}(K_\tau(\delta)) \subset K_t(\delta)$. This corresponds to building a compact (closed and bounded) tubular region trapping the flow around the solution during $[0, T]$, where we can study the perturbations using the smoothness of the flow and apply standard results from $\mathbb{R}^n$. Indeed, consider $p_1, p_2 \in K_\tau(\delta)$ where $\tau \in [t_i, t_{i+1}]$ and look at the propagation for the next t units of time, $\|\phi_{t-\tau}(p_1) - \phi_{t-\tau}(p_2)\|$ through chart i. From the smoothness of the flow and the compactness of $K(\delta)$ we see that:

$$\|\phi_{t-\tau}(p_1) - \phi_{t-\tau}(p_2)\| = \|(\phi_{t-\tau} \circ \varphi_i^{-1})(\varphi_i(p_1)) - (\phi_{t-\tau} \circ \varphi_i^{-1})(\varphi_i(p_2))\| \leq C_i D_i \|p_1 - p_2\|$$

where we have applied the Mean Value and Extreme Value Theorems twice : first to the map $\phi_{t-\tau} \circ \varphi_i^{-1}$ which is smooth on $\varphi_i(U_i)$ from the smoothness of ϕ and second to φ_i which is smooth because recall that $\varphi_i := \pi_m \circ F_{|U_i \cap \mathcal{M}}$ for a diffeomorphism F from an open set $U \subset \mathbb{R}^n$, obtaining the constants C_i and D_i respectively. Then, the lemma is proven after defining $C := \max_{i=0,...,n-1} \; C_i D_i$. □

Lemma 1 is used to prove the convergence of our Coordinate-Adaptive method, where the idea of the proof consists on expressing the global error as an accumulation of the propagated local errors of the previous steps.

Theorem 1. Consider the differential equation $c'(t) = V(c(t))$ on a submanifold $\mathcal{M} \subset \mathbb{R}^n$ with a smooth vector field V and initial condition $p_0 \in \mathcal{M}$ with solution c defined on $[0, T]$. Then, the Coordinate-Adaptive method with an underlying one-step method of order p converges to the solution c. More specifically:

$$\|p_n - c(nh)\| \leq Ch^p, \quad \text{for } nh \leq T,$$

where C is independent of h but depends on the length of the interval T.

Proof. The magnitude of the global error at the nth step is given by:

$$|E_n| = |p_n - p(t_n)|$$

We can rewrite this in terms of the previous estimates as follows:

$$\|E_n\| = \|\phi_0(y_n) - \phi_h(y_{n-1}) + \phi_h(y_{n-1}) + \phi_{2h}(y_{n-2}) - \phi_{2h}(y_{n-2}) + \ldots + \phi_{(n-1)h}(y_1) - \phi_{nh}(y_0)\|$$

So by the triangle inequality we see that:

$$\|E_n\| \leq \sum_{j=0}^{n-1} \|\phi_{t_n - t_{j+1}}(y_{j+1}) - \phi_{t_n - t_j}(y_j)\|$$

Since $\phi_{t_n - t_j}(y_j) = \phi_{t_n - t_{j+1}}(\phi_h(y_j))$, we can use Lemma 1 and the uniform bound for the local error as long as y_n and $\phi_h(y_{n-1})$ are in $K_{nh}(\delta)$, which is possible by taking h small enough. This allows us to deduce that:

$$\|E_n\| \leq CC_0(nh)h^p \leq CC_0Th^p$$

Therefore, the method is convergent with maximum error of order p.

□

From these results, we have effectively established the convergence of the Coordinate-Adaptive algorithm for smooth vector fields on general smooth manifolds, satisfying the finite atlas condition mentioned after definition 6. The key idea is that local convergence properties of the chosen one-step method (like the Euler method) extend globally by exploiting the smoothness of the flow and the fact that manifolds are locally identical to $\mathbb{R}^n$. By switching between charts and adapting to the local problems in a consistent way, the global error does not accumulate uncontrollably, and thus the method converges on the manifold at same rate as that of the underlying one-step method used in the coordinate space.

4. QUANTUM MECHANICS

4.1. The postulates of quantum mechanics

In this chapter, we will test the Coordinate-Adaptive algorithm from chapter 3 against the problem of predicting the evolution of a state of a quantum bit (qubit), which is the fundamental quantum system of interest in the fields of Quantum Computating and Information Geometry. In order to appreciate how our algorithm can prove useful in this context, some background knowledge on the physics that describe the behaviour of microscopic systems, that is, Quantum Mechanics, is required. The theory has been developed in the beginning of the 20-th century and its formalization and development has been progressing ever since. An introductory and basic mathematical formulation of Quantum Mechanics may be given in terms of the four postulates below [23]:

Postulate 1: State Space

Associated to any isolated physical is a complex Hilbert space $\mathcal{H}$ known as the state space of the system. The system is completely described by a vector of unit length, $|\psi\rangle \in \mathcal{H}$, known as the state vector.

It is important to observe that this postulate does not say anything about the specific form of $\mathcal{H}$, which is the task of the modeler (physicist) to specify. Also, it does not specify the dimension of the space, which in principle could be infinite.

In the case of the qubit, its state space is two dimensional i.e. $\mathcal{H} \simeq \mathbb{C}^2$ and the state vector $|\psi\rangle$ can be written in terms of an orthonormal basis known as the computational basis, given by $|0\rangle, |1\rangle$ where :

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

The numbers $\alpha, \beta \in \mathbb{C}$ are called probability amplitudes and the unit length condition implies that their modulus squared, sum up to 1: $|\alpha|^2 + |\beta|^2 = 1$.

Postulate 2: Unitary Evolution

The evolution of a closed quantum system is described by a unitary transformation. In particular, the state of the system at time t_2, $|\psi(t_2)\rangle$ is related to the state at time t_1, $|\psi(t_1)\rangle$ as follows:

$$|\psi(t_2)\rangle = U(t_2 - t_1)|\psi(t_1)\rangle,$$

where U is unitary i.e. $U^\dagger = U^{-1}$.

*Equivalently, the state satisfies **Schrödinger's Equation**:*

$$\begin{cases} i\hbar \dfrac{d}{dt} |\psi(t)\rangle = H\,|\psi(t)\rangle, \\ |\psi(0)\rangle = \psi_0 \in \mathcal{H} \end{cases}$$

where H is a Hermitian operator, i.e. $H^\dagger = H$, known as the Hamiltonian of the system.

For instance, if the Hamiltonian does not depend on time, one can verify that the solution is :

$$|\psi(t)\rangle = \exp\left(\frac{-i}{\hbar}Ht\right)\psi_0$$

where $exp(A) := \sum_{n=0}^{+\infty} \frac{A^n}{n!}$. We see how the equivalence with the discrete version by using the properties of the exponential and letting $U(t_2 - t_1) = exp(\frac{-i}{\hbar}H(t_2 - t_1))$.

Example 4.1 (Time evolution under $H = \sigma_z$)**.** To illustrate the use of the Schrödinger equation, let us look at the special case where the Hamiltonian is given by the Pauli z-matrix

$$H = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

As an initial state, we choose the equal superposition (denoted the $|+\rangle$ state)

$$|\psi_0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Since σ_z is already diagonal, this exponential can be written down immediately:

$$U(t) = \exp\left(-\tfrac{i}{\hbar}Ht\right) = \begin{pmatrix} e^{-it/\hbar} & 0 \\ 0 & e^{it/\hbar} \end{pmatrix}.$$

Then, the state after t units of time is determined by the action of $U(t)$ on $|\psi_0\rangle$ which results in:

$$|\psi(t)\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} e^{-it/\hbar} \\ e^{it/\hbar} \end{pmatrix}.$$

In the same spirit of the first postulate, the evolution postulate only gives a general property of the evolution of a closed system, by saying that it undergoes unitary motion. Finding the Hamiltonian of the underlying system is a problem which must be addressed by the particular theory under study. However, this postulate only applies for the situations where the system has not been affected by an external observer, that is, it only describes how the state changes while kept in isolation. Interfering by observing the system and making a measurement breaks this assumption, but postulate 3 is in charge of explaining how the evolution continues immediately after measurement, a subtle phenomenon known as the collapse of the wave function, which has been an open source of discussion by many important physicists over the last decades.

Postulate 3: Measurement

Measurements in Quantum Mechanics are described by a collection of measurement operators $\{M_m\}$ acting on $\mathcal{H}$. The probability of observing measurement outcome m from state $|\psi\rangle$ is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle.$$

If such an outcome is measured, the state immediately collapses to the state

$$|\psi'\rangle = \frac{M_m|\psi\rangle}{\sqrt{p(m)}}.$$

A compact way to express the fact that $\{p(m)\}_{m\in I}$ is a probability distribution for the measurement outcomes is done through the "Completeness Equation":

$$\sum_m M_m^\dagger M_m = I.$$

In the case of a qubit, the measurement operators are given by

$$M_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \text{ and } M_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$

which extract the amplitudes of the state vector from each component separately. In the case that the qubit state is

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

and a 0, or 1, is measured, then the postulate affirms that the qubit immediately changes to the state

$$M_0|\psi\rangle = \frac{\alpha}{|\alpha|}|0\rangle, \text{ or } M_1|\psi\rangle = \frac{\beta}{|\beta|}|1\rangle.$$

What this means physically is that the qubit loses its randomness features and degenerates into a deterministic system which constantly gives the same measurement outcome afterwards. This is true unless an experimenter intervenes by applying an external action on it.

In the same fashion as classical computers alter bit values through circuits and logic gates, quantum computers are designed to be capable of altering the state of qubits by means of quantum gates, which correspond to unitary transformations on the state vector. The most common one is the Hadamard gate, whose action is defined by the matrix

$$A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

When acting, it resets a qubit from its "deterministic" states $|0\rangle$, or $|1\rangle$, into the "coin flip" states

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \text{ or } |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

and vice versa.

An important observation is that state vectors that are constant multiples of each other have the same observable properties. Specifically, the state vector $e^{i\gamma}|\psi\rangle$, which differs only by a phase factor $e^{i\gamma}$ from $|\psi\rangle$, possesses the same probability distribution as $|\psi|$. Indeed, because of the statement of Postulate 3, it holds

$$\langle\psi|e^{i\gamma}\, M_m^\dagger M_m\, e^{i\gamma}|\psi\rangle = \langle\psi|M_m^\dagger M_m|\psi\rangle.$$

This is true for any general pair of states whose amplitudes have the same magnitude,entry-wise, like $|+\rangle$ and $|-\rangle$, which have identical magnitudes of 0.5 but differ by local phase factors 1 and -1 in their second amplitudes. However, the effect of a unitary operator on them will lead to observable differences in the future. For example, applying the Hadamard gate to $|+\rangle$ and $|-\rangle$ results in $|0\rangle$ and $|1\rangle$, whose statistical meaning is radically different.

This observation regarding the physical indistinguishability among state vectors, equal up to global phase factors shows that this formulation is superfluous , an issue that has motivated the development of a more accurate and complete description of a state, which we will see later. Although not directly used, understanding this first formulation was a necessary step towards understanding the details behind the prediction problem where our algorithm is used.

Postulate 4: Composite Systems

The state space of a composite system consisting of subsystems A and B is the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$. If the two systems are in states $|\psi_A\rangle$ and $|\psi_B\rangle$, the joint state is

$$|\psi_{AB}\rangle = |\psi_A\rangle \otimes |\psi_B\rangle.$$

Even though it is not of interest for the development of this thesis, this postulate is one of the most important for the whole theory of Quantum Mechanics, since one of the main consequences is the existence of the known entangled states (see the main reference [23] for further details).

4.2. Density operators

A more advanced formulation of quantum mechanics requires the notion of a density operator to replace that of a state vector. Specifically, the space $\mathcal{D}(\mathcal{H})$ of density operators is defined as

$$\mathcal{D}(\mathcal{H}) = \left\{\rho \in \mathcal{B}(\mathcal{H}) \,\middle|\, \rho = \rho^\dagger,\ \rho \geq 0,\ \mathrm{Tr}(\rho) = 1\right\}$$

where $\mathcal{B}(\mathcal{H})$ is the space of linear operators on $\mathcal{H}$.[13] Elements in $\mathcal{D}(\mathcal{H})$ are the quantum states of the system under investigation. Note that $\mathcal{D}(\mathcal{H})$ is a convex set, so that it makes sense to take convex combinations of quantum states.

The objective of this formulation is two-fold:

Firstly, it fixes the redundancy introduced by the state vector perspective that we showed earlier. This is done by mapping the state vectors to their rank-one projector via:

$$|\psi\rangle \longmapsto \rho_\psi = |\psi\rangle\langle\psi|$$

[13] Necessarily bounded because $\mathcal{H}$ has finite dimension.

Which effectively implies that:

$$\rho_{e^{i\theta}\psi} = e^{i\theta}|\psi\rangle\langle\psi|e^{-i\theta} = e^{i\theta}e^{-i\theta}|\psi\rangle\langle\psi| = |\psi\rangle\langle\psi| = \rho_\psi.$$

This type of operators are called the **pure states** of the system.

Secondly, this framework is more general as it allows for the representation of a broader class of situations. Apart from the inherent probabilistic nature of a Quantum system, this formulation can also describe cases where there is an extra source of uncertainty coming from the observer's ignorance about the state vector describing the system, (for example, this can happen right after making a measurement) which contemplates looking at a non-singleton set of state vectors $\{(|\psi_i\rangle, p_i)\}_{i=1}^N$, where p_i is the probability that $|\psi_i\rangle$ is the current state vector of the system. In this situation, we say that the system is in a **mixed state**, given by the operator:

$$\rho = \sum_{i=1}^{N} p_i\, |\psi_i\rangle\langle\psi_i| \quad \text{with} \quad \sum_{i=1}^{N} p_i = 1.$$

Within this framework, each pure state is one to one related to a class of state vectors that are equal up to global phase factor whereas there is more than one ensemble of state vectors associated to one mixed state (in fact, infinitely many). For example,the ensembles of states $E_1 = \{(\frac{1}{2}, |0\rangle), (\frac{1}{2}, |1\rangle)\}$ and $E_2 = \{(\frac{1}{2}, |+\rangle), (\frac{1}{2}, |-\rangle)\}$ both give the same mixed state, $\rho = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$ even though they refer to very different physical instances: In E_1 there is only external noise and no randomness coming from the qubit's state vector, while E_2 has both. It turns out that understanding the kind of vectors that share a density operator is a fruitful idea for devising error correction algorithms that keep accurate information transmission (read Chapter 8 from [23] for more details)

In addition, it can be proven that a sufficient and necessary condition for a state ρ to be pure is that, its purity, defined as $Tr(\rho^2) \in [1/2, 1]$ equals 1.

The postulates we explained for the state vector can be reformulated in terms of the density operator [23] and look as follows:

Postulate 1: State Space

A quantum mechanical system is described through a complex Hilbert space $\mathcal{H}$. The physical states of the system are identified with elements in the space $\mathcal{D}(\mathcal{H})$, which is referred to as the state space of the system.

Postulate 2: Evolution

The evolution of a closed quantum system is described by a unitary transformation. In particular, the state of the system at time t_2, $\rho(t_2)$ is related to the state at time t_1, $\rho(t_1)$ as follows:

$$\rho(t) = U(t_2 - t_1)\rho(t_1)U(t_2 - t_1)^\dagger$$

Equivalently, the state satisfies the operator differential equation:

$$\begin{cases} \frac{d\rho}{dt} = -\frac{i}{\hbar}[H,\rho], \\ \rho(0) = \rho_0 \in \mathcal{D}(\mathcal{H}) \end{cases}$$

where $[H,\rho] = H\rho - \rho H$ *and* H *is the Hamiltonian of the system.*

To see how this is deduced from the state vector formulation, note that, with probability p_i, the qubit's state vector is $|\psi_i\rangle$ and changes to $U(t_2 - t_1)|\psi_i(t_1)\rangle$ at time t_2. Hence, the new state becomes

$$\begin{aligned} \rho(t_2) &= \sum_i p_i U(t_2 - t_1)|\psi_i(t_1)\rangle\langle\psi_i(t_1)|U(t_2 - t_1)^\dagger \\ &= U(t_2 - t_1)(\sum_i p_i|\psi_i(t_1)\rangle\langle\psi_i(t_1)|)U(t_2 - t_1)^\dagger \\ &= U(t_2 - t_1)\rho(t_1)U(t_2 - t_1)^\dagger \end{aligned}$$

Similarly, the operator differential equation can be obtained using Schrodinger's equation as follows:

$$\begin{aligned} \frac{d}{dt}\rho &= \frac{d}{dt}(\sum_i p_i|\psi_i\rangle\langle\psi_i|) \\ &= \sum_i p_i \frac{d}{dt}|\psi_i\rangle\langle\psi_i| + \sum_i p_i|\psi_i\rangle\frac{d}{dt}\langle\psi_i| \\ &= \sum_i p_i \frac{-i}{\hbar}H|\psi_i\rangle\langle\psi_i| + \sum_i p_i|\psi_i\rangle\frac{i}{\hbar}H\langle\psi_i| \\ &= \frac{-i}{\hbar}(H\sum_i p_i|\psi_i\rangle\langle\psi_i| - \sum_i p_i|\psi_i\rangle\langle\psi_i|H) \\ &= \frac{-i}{\hbar}[H,\rho] \end{aligned}$$

Furthermore, the solution is given by $\rho(t) = exp(\frac{-i}{\hbar}Ht)\rho_0 exp(\frac{i}{\hbar}Ht)$.

Postulate 3: Measurement

Measurements in Quantum Mechanics are described by a collection of measurement operators $\{M_m\}$ *acting on* $\mathcal{H}$*. The probability of outcome m when the system is in state* $|\psi\rangle$ *is given by*

$$p(m) = \mathrm{Tr}(M_m^\dagger M_m \rho).$$

and the state of the system after the measurement is

$$\frac{M_m \rho M_m^\dagger}{\mathrm{Tr}(M_m^\dagger M_m \rho)}.$$

These measurement operators satisfy the Completeness Equation:

$$\sum_m M_m^\dagger M_m = I.$$

Postulate 4: Composite Systems

The state space of a composite physical system is the tensor product of the component physical systems. Moreover, if we have systems numbered 1 *through n, and the system number i is prepared in the state ρ_i, then the joint state of the total system is $\rho_1 \otimes \rho_2 \otimes ... \rho_n$.*

4.3. Geometry of the Qubit state space

The state space of a qubit system $\mathcal{D}(\mathcal{H})$ with $\mathcal{H} \cong \mathbb{C}^2$, can be interpreted geometrically after introducing the basis for the real vector space of self adjoint matrices (and hence for $\mathcal{D}(\mathcal{H})$) . In particular, the state ρ can be expressed uniquely with respect to the basis given by the Identity matrix and the three Pauli matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

$$\rho = \frac{I + \vec{r} \cdot \vec{\sigma}}{2} = \frac{1}{2} \begin{pmatrix} 1+z & x-iy \\ x+iy & 1-z \end{pmatrix}$$

for a unique vector $\vec{r} = (x, y, z)^T = (\mathrm{Tr}(\rho\sigma_x), \mathrm{Tr}(\rho\sigma_y), \mathrm{Tr}(\rho\sigma_z))^T$.

Verifying that a matrix with this form is self- adjoint and has unit trace is straightforward. On the other hand, figuring out what constraints $\vec{r} = (x, y, z)^T \in \mathbb{R}^3$ so that the non negativity of eigenvalues is ensured ($\rho \geq 0$) can be deduced by solving the characteristic equation:

$$\det(\rho - \lambda I) = \begin{pmatrix} \frac{1+z}{2} - \lambda & \frac{x-iy}{2} \\ \frac{x+iy}{2} & \frac{1-z}{2} - \lambda \end{pmatrix} = \frac{1}{4}(1-z^2) - \frac{\lambda}{2}(1+z) - \frac{\lambda}{2}(1-z) + \lambda^2 - \frac{1}{4}(x^2+y^2) = 0$$

Multiplying by 4 and simplifying reduces the equation to the quadratic:

$$4\lambda^2 - 4\lambda + \|\vec{r}\|^2 + 1 = 0$$

Whose solutions are:

$$\lambda_{1,2} = \frac{1 \pm \|\vec{r}\|}{2}$$

So the condition $\rho \geq 0$ is equivalent to the constraint $\|\vec{r}\| \leq 1$.

Therefore, the state of a qubit can be identified with a point in the closed unit ball $\overline{B}(0,1) = \{r \in \mathbb{R}^3, \|r\| \leq 1\} \subset \mathbb{R}^3$. [14] A further geometrical property from this representation is that the higher the purity of the state, the larger the associated bloch vector. Indeed, all points lying in an inner sphere of radius $p \in [0, 1]$ correspond to states of constant purity. To see this, we note that:

$$\rho^2 = \frac{1}{4}\left(I + 2\,\vec{r} \cdot \vec{\sigma} + (\vec{r} \cdot \vec{\sigma})^2\right).$$

[14] It can be thought of as the union of the open ball forming the interior and the sphere as the boundary, which are smooth manifolds of dimensions 3 and 2 respectively.

Hence

$$\mathrm{Tr}(\rho^2) = \mathrm{Tr}\left(\frac{I + 2\vec{r}\cdot\vec{\sigma} + (\vec{r}\cdot\vec{\sigma})^2}{2}\right) = \frac{1}{4}\,\mathrm{Tr}\Big(I + 2\vec{r}\cdot\vec{\sigma} + (\vec{r}\cdot\vec{\sigma})^2\Big).$$

And since $(\vec{r}\cdot\vec{\sigma})^2 = \|\vec{r}\|^2 I$, we have

$$\mathrm{Tr}(\rho^2) = \frac{1}{4}\,\mathrm{Tr}\Big(I + 2\vec{r}\cdot\vec{\sigma} + \|\vec{r}\|^2 I\Big) = \frac{1}{4}(2 + 0 + 2\|\vec{r}\|^2).$$

That is,

$$\mathrm{Tr}(\rho^2) = \frac{1 + \|\vec{r}\|^2}{2}.$$

In this way, states with constant purity $\mathcal{D}(\mathbb{C}^2) \cap \{\mathrm{Tr}(\rho^2) = \mathrm{p}\}$, $p \in [\frac{1}{2}, 1]$, are mapped to a sphere $\mathcal{S}_r^2$ of radius $r = \sqrt{2p-1}$.

More properties can be studied from this geometrical representation when focusing on its boundary, famously known as the **Bloch Sphere**.

Without loss of generality, a pure state can be written as $\rho = |\psi\rangle\langle\psi|$ with:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \mathrm{e}^{\phi i}\,\sin\left(\frac{\theta}{2}\right)|1\rangle \text{ for } \phi \in (0, 2\pi),\ \theta \in (0, \pi)$$

Which means that:

$$\begin{aligned}
\rho_\psi &= \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{pmatrix}\begin{pmatrix} \cos\frac{\theta}{2} & e^{-i\phi}\sin\frac{\theta}{2} \end{pmatrix} \\
&= \begin{pmatrix} \cos^2\frac{\theta}{2} & \cos\frac{\theta}{2}\sin\frac{\theta}{2}e^{-i\phi} \\ \cos\frac{\theta}{2}\sin\frac{\theta}{2}e^{i\phi} & \sin^2\frac{\theta}{2} \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 1 + cos\theta & \sin\theta(\cos\phi - i\sin\phi) \\ \sin\theta(\cos\phi + i\sin\phi) & 1 - \cos\theta \end{pmatrix},
\end{aligned}$$

and thus

$$\rho = \frac{1}{2}(I + \vec{r}(\phi, \theta)\cdot\vec{\sigma}),$$

for $\vec{r}(\phi, \theta) = (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta)) \in S^2$.

Hence, the pure state ρ with state vector $|\psi(\phi, \theta)\rangle$ is represented by the point with longitude ϕ and latitude θ in the unit sphere S^2. This additional observation gives further physical insight about the geometrical representation of pure states through the Bloch Sphere as the north and south poles refer to the states with state vectors $|0\rangle$ and $|1\rangle$ respectively (see figure 4.1). Also, higher latitude points correspond to states whose probability distribution is biased towards measurement outcome 1 and the equator represents all pure states with a uniform distribution across the 2 outcomes.

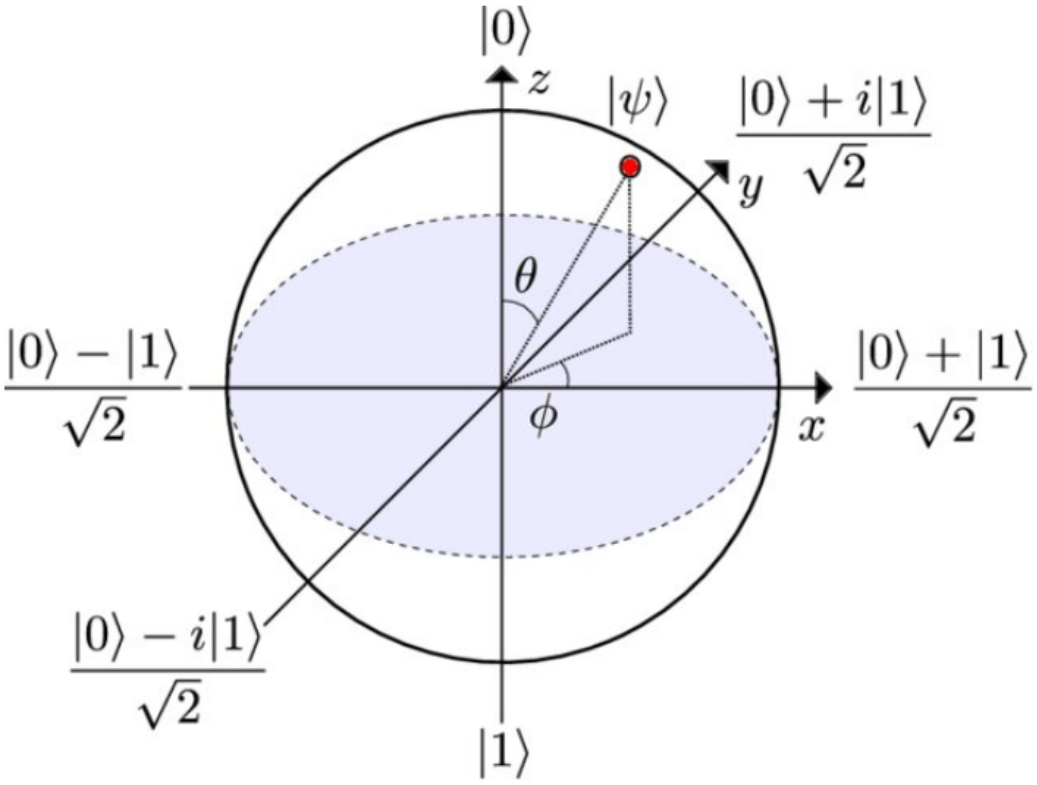


Fig. 4.1. The Bloch Sphere is a geometrical representation of the pure states of a qubit [24].

4.4. Test against Qubit Dynamics

It is possible to find the dynamics on the ball associated to the density operator differential equation [15] :

$$\frac{d\rho}{dt} = i[H, \rho]$$

To do this, we can write $H = \dfrac{h_0 I + h_x\sigma_x + h_y\sigma_y + h_z\sigma_z}{2}$ for a particular choice of real numbers h_0, h_x, h_y, h_z. Similarly, ρ can be expressed in terms of the components x, y, z which will be functions of time. The left hand side corresponds to the derivative of ρ which is given by:

$$\rho' = \frac{1}{2}(x'\sigma_x + y' + y'\sigma_y + z'\sigma_z)$$

[15] We have redefined $H := \frac{-1}{\hbar}$ for convenience

And the right hand side can be expressed as:

$$\begin{aligned} i[H,\rho] &= \frac{i}{4}[h_0 I + h_x\sigma_x + h_y\sigma_y + h_z\sigma_z,\ I + x\sigma_x + y\sigma_y + z\sigma_z] \\ &= \frac{i}{4}[h_x\sigma_x + h_y\sigma_y + h_z\sigma_z,\ x\sigma_x + y\sigma_y + z\sigma_z] \\ &= \frac{i}{4}\Big((h_xy - h_yx)[\sigma_x,\sigma_y] + (h_xz - h_zx)[\sigma_z,\sigma_x] + (h_yz - h_zy)[\sigma_z,\sigma_y]\Big) \\ &= \frac{1}{2}\Big((h_zy - h_yz)\sigma_x + (h_xz - h_zx)\sigma_y + (h_yx - h_xy)\sigma_z\Big) \end{aligned}$$

Therefore, equating both sides and matching the components of the basis gives a vector field (independent of h_0) on $\overline{B}(0,1)$:

$$\begin{cases} x' = h_zy - h_yz \\ y' = h_xz - h_zx \\ z' = h_yx - h_xy \\ x^2 + y^2 + z^2 = 1 \end{cases}$$

This vector field is indeed also tangent to any sphere of radius less than or equal to 1, something that can be checked by direct differentiation or through the purity of the solution $\rho = U\rho_0U^\dagger$:

$$\begin{aligned} \mathrm{Tr}(\rho^2) &= \mathrm{Tr}(\mathrm{U}\rho_0\mathrm{U}^\dagger\mathrm{U}\rho_0\mathrm{U}^\dagger) \\ &= \mathrm{Tr}(\mathrm{U}\rho_0^2\mathrm{U}^\dagger) \\ &= \mathrm{Tr}(\rho_0^2) \end{aligned}$$

The last equality follows from the cyclic property of the trace. As we know that states of constant purity are represented in the ball with a common radius, then, we confirm that the vector field on $\overline{B}(0,1)$ is in particular tangent to every purity sphere S^2_r where $r \in [0,1]$. Qualitatively speaking, the exact state evolution consist on clockwise rotations at a constant frequency around the axis specified by the Hamiltonian vector $\vec{h} = (h_x, h_y, h_z)^T \in \mathbb{R}^3$, with rotational speed given by $\|\vec{h}\|$.

We applied the Coordinate-Adaptive algorithm to predict the approximate state trajectories of a qubit starting at various initial states in the Bloch Sphere and subject to 3 different time-independent Hamiltonian's over the interval $[0, 2\pi]$, using a time step of $h = 0.025$ and the "plane-slicing" atlas for S^2 discussed in example 2.3.

The plots of the trajectories obtained are displayed in figures 4.2,4.3 and 4.4.

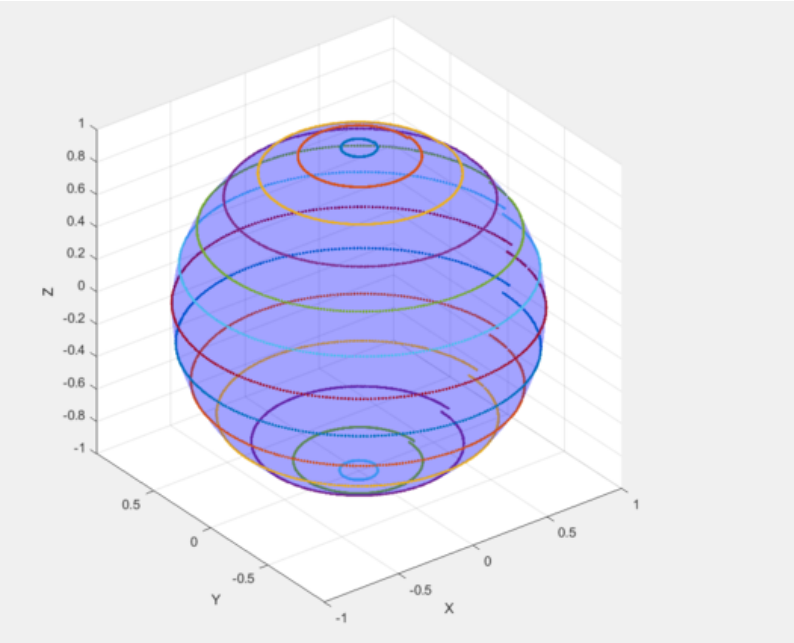


Fig. 4.2. Estimated qubit's state trajectories, using $h = 0.025$ and $H = \sigma_z$.

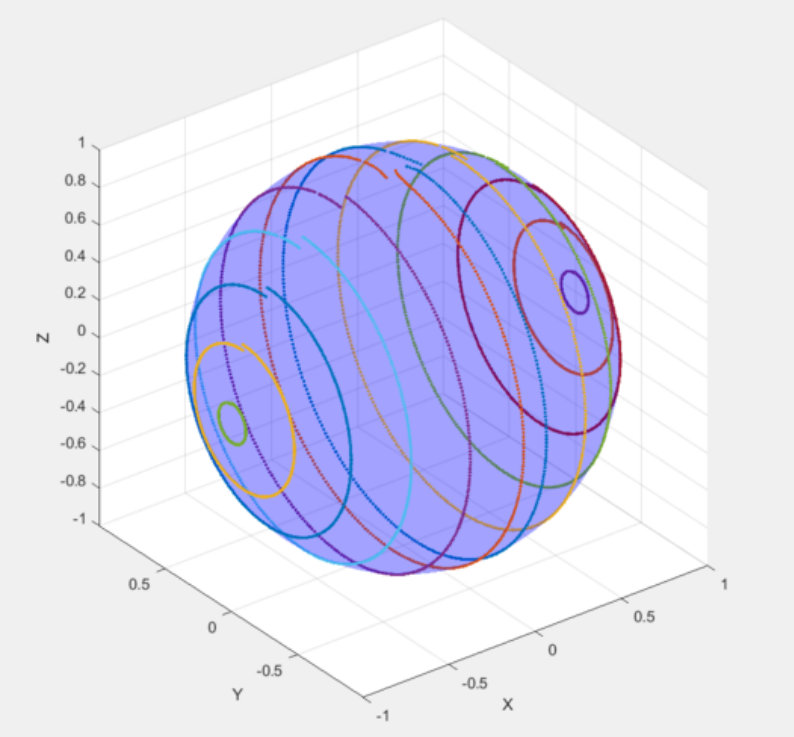


Fig. 4.3. Estimated qubit's trajectories, using $h = 0.025$ and $H = \sigma_x$.

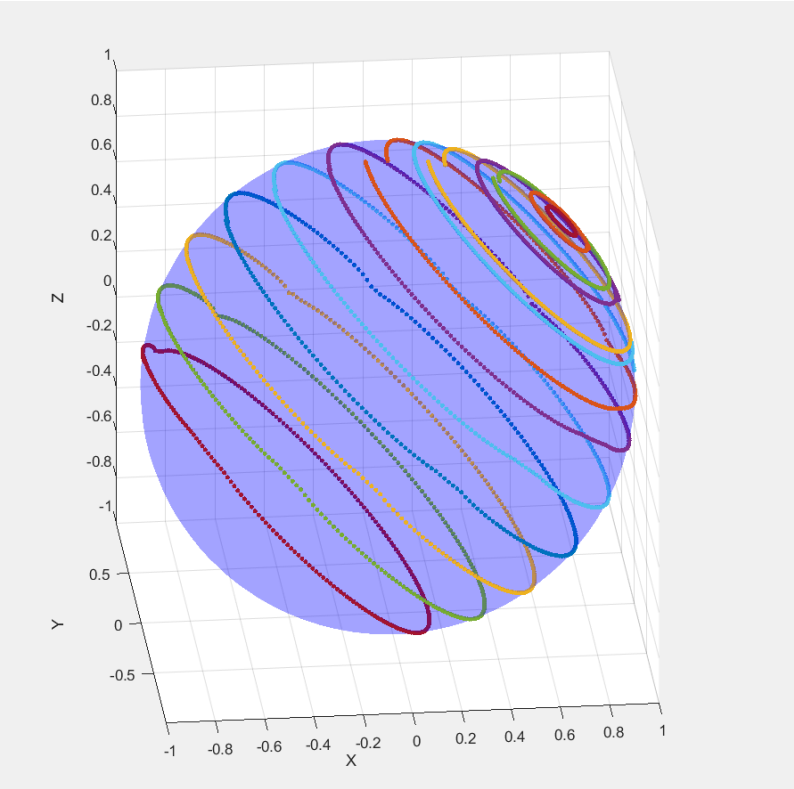


Fig. 4.4. Estimated qubit's trajectories, using $h = 0.025$ and $H = \frac{1}{2\sqrt{2}}(\sigma_x + \sigma_z)$.

As desired, predictions preserve the initial purity by staying on the sphere and remain very close to the exact solutions (great circles). Furthermore, table 4.1 summarizes the largest error of the Coordinate-Adaptive method when applied to the qubit prediction problem with Hamiltonian $H = \sigma_Z$ and initial pure state $\rho_0 = |+\rangle\langle+|$ using different values of h and underlying one-step methods.

h	Euler	RK2	RK4
2^{-3}	1.72e-01	1.69e-01	5.33e-02
2^{-4}	1.19e-01	8.54e-02	1.93e-02
2^{-5}	8.95e-02	5.23e-02	1.21e-02
2^{-6}	3.48e-02	2.28e-02	6.58e-03
2^{-7}	2.08e-02	1.15e-02	3.40e-03
2^{-8}	1.25e-02	5.57e-03	1.26e-03
2^{-9}	7.99e-03	3.33e-03	3.68e-04
2^{-10}	3.41e-03	1.46e-03	3.30e-05

TABLE 4.1. SUMMARY OF NUMERICAL ERRORS WITH DIFFERENT UNDERLYING METHODS, FOR THE CASE OF $H = \sigma_Z$ AND VARIOUS STEP SIZES.

Aside from the qualitative look of the plots, these nummerical data also provides us with empirical evidence that reinforces our theoretical analysis on the convergence of the method, as errors approach 0 across all columns. In particular, we also get nummerical confirmation that the order of convergence of the Coordinate-Adaptive method preserves the order of the underlying one-step methods (1,2,4), visible to a certain degree in the table. For more details about the proofs on the orders of convergence of these methods, check out [1].

Additionally, the graph from figure 4.5 illustrates the evolution of the error made by the integrator at each step, as well as providing a visual record of the different phases from the integration cycle, helping us to verify its correctness.

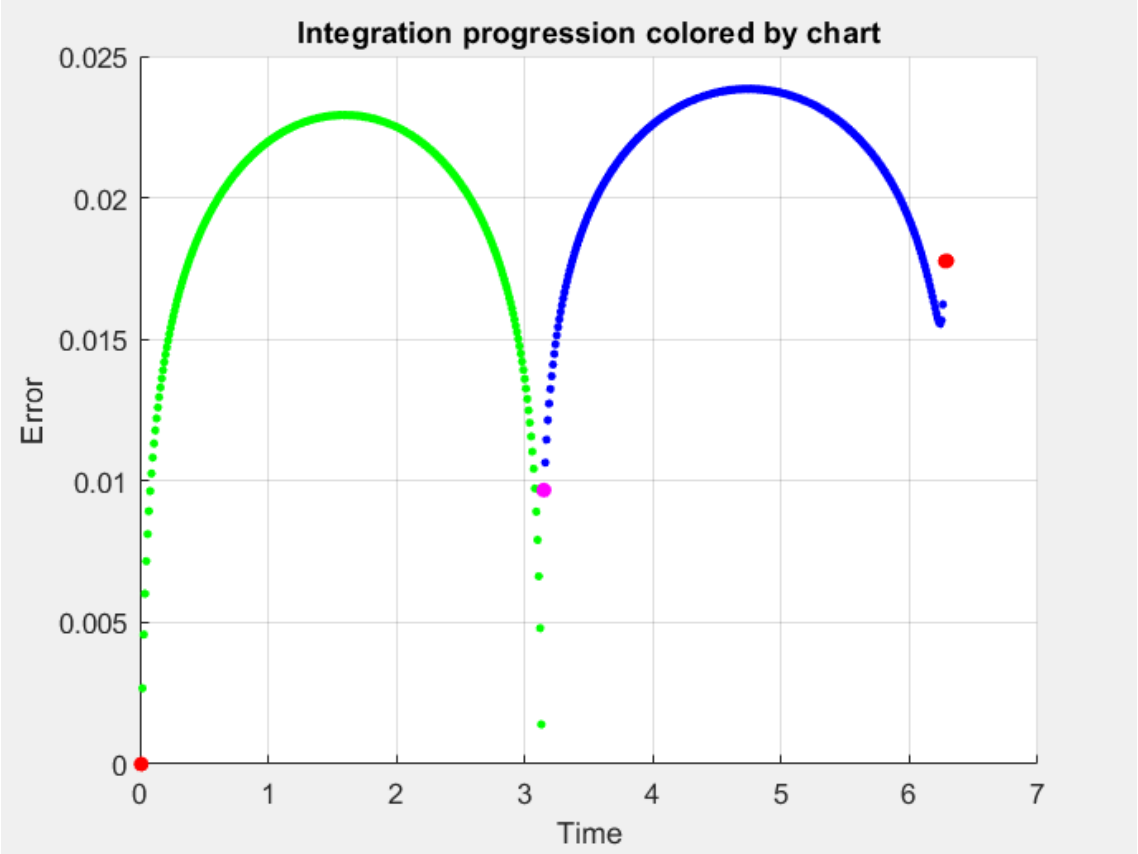


Fig. 4.5. Visual representation of how the error evolves with time, where the color of a point $(t_n, |e_n| = p_n - p(t_n))$ is determined by the chart used to find p_n.

Starting from the initial condition at $(1, 0, 0)$, which is only covered by the positive x chart, the next estimate is obtained based on the projected velocity onto the $y - z$ plane. This is represented by the red colored point right at the start. Then, the output estimate is now inside this previously used chart and also the one for the hemisphere with $y < 0$, which is the one the algorithm changes to, colored in green. It remains in this chart for approximately π radians, until experiencing an overshot around $(-1, 0, 0)$, where it resorts momentarily to the "left" chart, corresponding to the hemisphere with $x < 0$, visible by a single pink point on the graph. But immediately after obtaining the associated estimate

from this chart, which lies on the second quarter of the $x-y$ plane, the algorithm settles for the rest of the simulation on the chart covering the hemisphere with $y > 0$ shown by the blue region on the right-half of the graph. In the end, just about to loop back to $(1, 0, 0)$, a second overshot similar to the first one occurs and the final estimate is obtained using the same chart from the the beginning of the integration, reflected by the red color at $T = 2\pi$.

5. SOCIO-ECONOMIC CONTEXT

5.1. Applications and Sector Relevance

The Coordinate-Adaptive algorithm is a geometry-preserving numerical integrator of differential equations on smooth manifolds and so it is virtually applicable to any areas where mathematical modeling using autonomous systems of differential equations is present. Some of the places lying in this broad spectrum are:

- **Prediction Problems in Science and Engineering**
  Many models in physics, robotics, and mechanics involve dynamical systems evolving on manifolds. Some examples in this class include celestial orbits, rigid body motion like trajectory optimization and motion planning in robotics, and constrained mechanical systems. As we saw in Chapter 1, standard integrators may lead to drift and violation of conserved quantities which are of great importance in all of the above scenarios. By design, our integrator will maintain these invariants, offering superior accuracy and stability.

- **Quantum Computing—Scaling to Multiple Qubits**
  The method has been tested on a single qubit evolving under a time-independent Hamiltonian and it is scalable to multiple qubit systems whose state space will be a higher dimensional space. This is particularly relevant to quantum computing, where accurate prediction of the quantum state trajectory (unitarity, fidelity) is essential for the correct execution of quantum algorithms and in the field of Quantum Information Theory, which studies studies how information can be represented, transmitted, and manipulated using quantum systems.

  We should also highlight that the quantum computing industry is experiencing an explosive growth with a current market valuation of around USD 1.4 billion in 2024, projected to surpass USD 4 billion by 2030 (CAGR ∼19%–30%). Other forecasts anticipate it could reach tens of billions by the 2030s. Moreover, the development of quantum computing will benefit multiple sectors like healthcare, finance, materials science, and cybersecurity. Also, it has potential to contribute to quantum simulation tools, algorithm design frameworks, and quantum software ecosystems which are critical enablers of the emerging quantum industry.

5.2. Broader Socio-Economic Benefits

1. **Accessibility and Open Source Value:** The implementation is fully open-source, democratizing access to advanced geometric integration methods. This stands in

contrast to proprietary solvers, which are often expensive or inaccessible to smaller institutions.

2. **Educational Impact:** The integrator serves as a potent educational resource, bridging abstract differential geometry with computational practice. It can be directly used in graduate coursework or research in numerical analysis, geometry, and physics.

Therefore, the project has a positive impact on the development of the open science and the interaction between science and society. It is important to recall that both are fundamental principles for the modern scientific community.

5.3. Cost Estimation

To assess the economic value of this work, a costing model has been constructed using standard academic and commercial rates.

5.3.1. Intellectual Resources

The primary source of expenses comes from the intellectual labour required to develop the solution. We therefore estimated the salaries of the student (junior engineer), the thesis supervisor, who is an "Investigador distinguido" in the Department of Mathematics at the Universidad Carlos 3 de Madrid, and the co-supervisor, who is a "Profesor Permanente Laboral", including in this estimation, individual expenses like taxes, insurance, and other social contributions.

Personnel Costs			
Role	**Time (h)**	**Hourly Wage (€ / h)**	**Compensation (€)**
Student Researcher	400	25	10,000
Supervisor (Investigador distinguido)	90	40	3600
Co-Supervisor	90	40	3600
Subtotal Personnel			17200

TABLE 5.1. PERSONNEL COSTS.

As displayed on Table 5.1, the total personnel costs equals 17200€.

5.3.2. Software and Hardware

The development of the algorithm was done using Matlab programming language, where the symbolic differentiation toolbox played an important role. This software requires an academic license which was paid by the university. Additionally, an HP Pavilion laptop

with an AMD Ryzen 7 processor, 16 GB RAM, and 512 GB SSD was used to run numerical simulations and generate the project's computational results, which assuming a useful life-expectancy of 4 years and a depreciation cost of around 20€ per month translates to 160€ for the approximate duration of the project. Aside from these two, Overleaf enhanced plan was used in the last month to allow for a fluent interaction between the student and the other two members of the team. This amounts to a partial cost of summarized in Table 5.2 below :

Name	**Price (€)**
MATLAB academic license + Symbolic Math Toolbox	500
HP Pavilion Laptop (depreciation)	160
Overleaf student (monthly)	9.68
Total	669.68

TABLE 5.2. SOFTWARE AND HARDWARE COSTS.

5.3.3. Transportation

During the first six months of the project, biweekly in-person meetings were conducted to review progress and discuss new material between the student and the supervisors. The student traveled approximately 68 km round trip per meeting (12 meetings total), resulting in the following relative cost estimate:

$$\text{Transportation cost} = 68\ \text{km/meeting} \times 12\ \text{meetings} \times 0.25\ €/\text{km} = 204\ €.$$

5.3.4. Total Costs

Using the different costs of each source studied, we obtain the final costs summary of our project described in Table 5.3 below:

Source	**Total (€)**
Intellectual	17,200
Software and Hardware	669.68
Transportation	204
Total Project Cost	18,073.68

TABLE 5.3. TOTAL PROJECT COSTS.

6. REGULATORY FRAMEWORK

The work presented in this thesis is fundamentally theoretical in nature and does not involve experimental procedures, personal data, or any sensitive material subject to ethical or legal regulation.

Regarding reproducibility and future use, the source code developed in this thesis is also publicly available in the GitHub repository [22] under the MIT License apart from being appended at the end of this document. Users are welcome to use and adapt the code, with the expectation that appropriate reference to this work will be made in academic or applied contexts.

The resources used in this thesis, like figures, tables and published literature, have been properly cited and referenced in accordance with the citation standards established by the university thesis writing guidelines. All figures included in this thesis were either generated directly from the author's own implementations or obtained from publicly available sources. In cases where external sources were used, full and accurate citations have been provided, ensuring proper attribution of intellectual property.

7. CONCLUSIONS AND FUTURE WORK

This thesis has concluded with the design and implementation of a numerical algorithm capable of accurately solving differential equations on smooth manifolds, a problem that arises in applied mathematics, every time a mathematical model is proposed. The development of the algorithm required learning advanced concepts from Differential Geometry, a field in which the author had no prior background at the beginning of the project. Through a process of gradual learning, these theoretical foundations were acquired and successfully applied to the construction of the Coordinate-Adaptive algorithm. Its implementation was supported by a formal convergence analysis and validated against a benchmark problem in quantum mechanics, demonstrating both robustness and accuracy. The project has therefore represented a genuine integration of mathematics, computer science, and physics.

Looking ahead, we would like to announce three promising directions for future research. First, the study of chart selection criteria requires further exploration. Although a theoretical framework was outlined, time constraints prevented testing it numerically and observing its performance with an input atlas having many overlaps. The algorithm could be significantly enhanced if systematic research were conducted on efficient and stable chart-selection strategies.

Second, following the positive results obtained with a single qubit, it is natural to investigate the algorithm's performance on arbitrary finite-dimensional pure quantum systems, such as multi-qubit configurations. These can be described within complex projective spaces, which are well-known smooth manifolds equipped with explicit atlases for any dimension. This would extend the reach of the method to realistic scenarios in quantum computation.

Third, an important step is to test the algorithm against quantum systems with time-dependent Hamiltonians and eventually extend it to open quantum systems governed by the GKLS master equation. These generalizations would substantially increase the algorithm's applicability in quantum physics, where environmental interactions and temporal variations play a central role.

BIBLIOGRAPHY

[1] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Philadelphia: Society for Industrial and Applied Mathematics, 1998.

[2] D. F. Griffiths and D. J. Higham, *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. London: Springer London, 2010.

[3] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*. Cambridge, UK: Cambridge University Press, 2008.

[4] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations* (Springer Series in Computational Mathematics), 2nd. Springer-Verlag, 2006, vol. 31, Structure-preserving algorithms for ordinary differential equations.

[5] J. Wisdom and M. Holman, "Symplectic maps for the n-body problem," *Astronomical Journal*, vol. 102, pp. 1528–1538, 1991.

[6] J. Wisdom and G. Sussman, "Numerical evidence that the motion of pluto is chaotic," in *Bulletin of the American Astronomical Society, Vol. 20, p. 901*, vol. 20, 1988, p. 901.

[7] L. Falorsi and P. Forré, "Neural ordinary differential equations on manifolds," *arXiv:2006.06663*, 2020.

[8] G. Benettin, A. M. Cherubini, and F. Fasso, "A changing-chart symplectic algorithm for rigid bodies and other hamiltonian systems on manifolds," *SIAM Journal on Scientific Computing*, vol. 23, no. 4, pp. 1189–1203, 2001.

[9] K. K. Lin, "Coordinate-independent computations on differential equations," Ph.D. dissertation, Massachusetts Institute of Technology, 1997.

[10] J. M. Lee, *Introduction to Smooth Manifolds* (Graduate Texts in Mathematics). New York: Springer Science & Business Media, 2003, vol. 218.

[11] M. D. Spivak, *A Comprehensive Introduction to Differential Geometry*. Berkeley, CA: Publish or Perish, Inc., 1979.

[12] V. I. Arnold, *Ordinary Differential Equations*. Berlin & Heidelberg; New York: Springer-Verlag, 1992.

[13] R. Penrose, *The road to reality*. Random house, 2006.

[14] Wikipedia contributors. "Manifold," Wikipedia. (), [Online]. Available: https://en.wikipedia.org/wiki/Manifold.

[15] Etudes, *Stereographic projection – interactive 3d model*, https://en.etudes.ru/models/stereographic-projection/4/.

[16] E. Hairer, *Solving differential equations on manifolds*, Universitè de Genève, Course Notes, 2011.

[17] T. Lew, R. Bonalli, L. Janson, and M. Pavone, "Estimating the convex hull of the image of a set with smooth boundary: Error bounds and applications," *arxiv:2302.13970*, 2024.

[18] D. Siemssen, "The semiclassical einstein equation on cosmological spacetimes," Ph.D. dissertation, Università degli Studi di Genova, 2015.

[19] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, tensor analysis, and applications*. Springer Science & Business Media, 2012, vol. 75.

[20] Z. Lin, D. Kong, and Q. Sun, "Modeling symmetric positive definite matrices with an application to functional brain connectivity," *arXiv:1907.03385*, 2019.

[21] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering (studies in nonlinearity)*. Westview press, 2001, vol. 1.

[22] A. Fernández-Paniagua. "Sourcecode." GitHub repository. (2025), [Online]. Available: https://github.com/Panithecracker/Coordinate-independent-integrator-of-differential-equations-on-smooth-manifolds.git.

[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[24] A. F. Kockum, *Quantum optics with artificial atoms*. Chalmers Tekniska Hogskola (Sweden), 2014.

A. SOURCE CODE

A.1. File: Chart.m

```
1
2  classdef Chart
3      properties
4          %requested:
5          domain      %domain of coord function
6          image       % image of coord function
7          phi         % manifold -> coords
8          phi_inv     % coords -> manifold
9          n           % dimension of embedding space
10         k           % dimension of manifold
11         name        % name of the chart (optional)
12         %internally inferred:
13         coords_sym  % Symbolic variables for the coordinates
14         Jacobian    % Jacobian of inverse map
15     end
16
17     methods
18         function obj = Chart(d,i, p, pinv, n, k, name)
19             % Parameterized Constructor
20             obj.domain = d;
21             obj.image = i;
22             obj.phi = p;
23             obj.phi_inv = pinv;
24             obj.n = n;
25             obj.k = k;
26
27             % Define symbolic variables for the k-dimensional
                   coordinate space
28             % 'c' is used as a prefix for coordinate variables, e.g
                   ., c1, c2, ...
29             obj.coords_sym = sym('c', [1 k]);
30
31             % Assign the symbolic inverse map
32             % The user provides an anonymous function that accepts
                   these symbolic variables
33             % and returns the symbolic expression for phi_inv.
34             phi_inv_sym = obj.phi_inv(obj.coords_sym');
35             obj.Jacobian = jacobian(phi_inv_sym, obj.coords_sym);
36
37             if nargin < 6
38                 obj.name = 'Unnamed';
39             else
40                 obj.name = name;
```

```
            end
        end

        function in = inside_domain(obj,p)
            % Check if the point in Rn lies in the domain
            in = obj.domain(p);
        end

        function in = inside_image(obj,c)
            in = obj.image(c);
            % Check if the point in Rk lies in the image of phi
        end

        function c = Coords(obj,p) %finds coordinates on chart
            c = obj.phi(p);
        end

        function p = Manifold(obj,c) %evaluates phi_inv
            p = obj.phi_inv(c);
        end

        function Jval = JacobianAt(obj,c)
            % Evaluates Jacobian at coordinate c
            Jval = double(subs(obj.Jacobian,obj.coords_sym,c'));
        end

        function A = Projector(obj,p)
            % Useful to convert a vector field of Rn into a vector
                field on M
            % Returns the "projector" onto tangent space at p of M
                with
            % respect to this chart basis (k by n matrix)
            c = obj.phi(p);
            J = obj.JacobianAt(c);
            A = inv(J'*J)*J';
        end
    end
end


function vj = TranslateVelocity(A,i,j,c_i,vi)
    % Expresses a tangent vector given in chart i into chart j basis
        .
    coords_i = sym('c',[1 2]);
    transition = A(j).Coords(A(i).Manifold(coords_i'));
    J = jacobian(transition,coords_i);
    vj = double(subs(J,coords_i,c_i'))*vi;
end

% That is, given a particle moving at coordinate velocity v_i away
    from c_i
```

```
% on chart i, it finds the respective coordinate velocity v_j from
% coordinate c_j in chart j.
% The change of basis matrix is the Jacobian of transition at c_i.

function list = GetCharts(Atlas,p)
    % For a given point and atlas, it returns the list of indexes
    % from charts on the atlas that contain the point.
    list = [];
    for i = 1:size(Atlas,1)
        if Atlas(i,1).inside_domain(p) == 1
            list = [list i];
        end
    end
end
```

A.2. File: CoordinateAdaptive.m

```
function M = CoordinateAdaptive(A,M0,T,h)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% input:
%  A: Atlas
%  M0: Initial condition vector
%  T: Final time
%  h : time step

%% Output: Matrix of estimates ordered by columns, in steps of
    dt units of
%time from M0 included.

%initialization:
n = ceil(T/h);
M = zeros(3,n+1);
M(:,1) = M0;

for i=1:n

    valid_charts = GetCharts(A,M(:,i)); %valid charts (indexes
       ) for current point
    chart_index = valid_charts(1); %pick chart
    C = A(chart_index).Coords(M(:,i)); %coords on current
       chart
    % 0-USE EXTERNAL OR INTERNAL VECTOR FIELD DESCRIPTION (
       USER CHOICE, JUST UNCOMMENT THE ONE NEEDED):
```

```
23      % V = A(chart_index).Projector(M(:,i))*f(M(:,i)); %
            external description
24      V =  V_chart(A,chart_index,C); %intrisically, using chart
            basis
25
26      %% 1-INTEGRATE ON CHART :
27      Cnext = C+V*h; %Euler e.g (any works)
28
29      valid_charts = valid_charts(valid_charts~=chart_index); %
            update valid charts
30
31      while (~A(chart_index).inside_image(C)) %outside of coord
            space, change charts
32          if (~isempty(valid_charts)) %alternatives available
33              new_index = valid_charts(1); %pick new chart
34              %% 2-CHANGE TANGENT SPACE BASIS USING TRANSITION
                    MAP AND INTEGRATE ON THE ADAPTED NEW CHART
35              V = TranslateVelocity(A,chart_index,new_index,C,V)
                    ;
36              chart_index = new_index;
37              C = A(chart_index).Coords(M(:,i));
38              Cnext = C+V*h; %Euler (any other works)
39              valid_charts = valid_charts(valid_charts~=
                    chart_index); %update "stack" of valid charts
40          else
41              fprintf("Tested with all charts, decrease step
                    size!\n") %charts consumed, notify and exit
42              return
43          end
44      end
45      %% 3-COORDINATE ESTIMATED, NOW INVERT TO OBTAIN THE POINT
            ESTIMATE ON THE MANIFOLD
46      M(:,i+1) = A(chart_index).Manifold(Cnext);
47  end
48  end
49
50  function vj = TranslateVelocity(A,i,j,c_i,vi)
51      coords_i = sym('c',[1 2]);
52      transition = A(j).Coords(A(i).Manifold(coords_i'));
53      J = jacobian(transition,coords_i);
54      vj = double(subs(J,coords_i,c_i'))*vi;
55  end
56
57  function list = GetCharts(A,p)
58      list = [];
```

```
    for i = 1:size(A,1)
        if A(i,1).inside_domain(p) == 1
            list = [list i];
        end
    end
end

function v = f(x)
    h2 = 0; h3 = 0; h4 = 1;
    v = [h4*x(2)-h3*x(3); h2*x(3)-h4*x(1); h3*x(1)-h2*x(2)];
end

function v = V_chart(A,chart_index,coord)
    vect = f(A(chart_index).Manifold(coord));
    if chart_index == 1 || chart_index == 2
        v = [vect(1);vect(2)];
    end
    if chart_index == 3 || chart_index == 4
        v = [vect(1);vect(3)];
    end
    if chart_index == 5 || chart_index == 6
        v = [vect(2);vect(3)];
    end
end
```