Statistical Shape Analysis
Analysis of sample time series trajectories on the sphere

MATH4021

MSc Dissertation in

Statistics and Applied Probability

2022/23

School of Mathematical Sciences

University of Nottingham

Ka Leuk, Hui

Supervisor: Dr. Yordan Raykov

I have read and understood the School and University guidelines on plagiarism. I confirm that this work is my own, apart from the acknowledged references.

Abstract

This dissertation explores the statistical analysis of the sample of time series trajectories of a moving object on the surface of a unit sphere. We first apply the method of unrolling, unwrapping, and wrapping, to map the trajectory onto a related Euclidean plane. Then, statistical methods were used to analyze the mapped data on this plane. And finally, we map the results back to the sphere using wrapping to perform future position predictions.

The statistical techniques applied in the dissertation are the Cubic Smoothing Spline and Gaussian Processes Regression (GPR). We apply both methods to fit data on the Euclidean plane, and compare their pros and cons. After that, we use the fitted GPR to perform forecasting, and use it to simulate multiple trajectories to see the possible position of the particle on the sphere in the future time step.

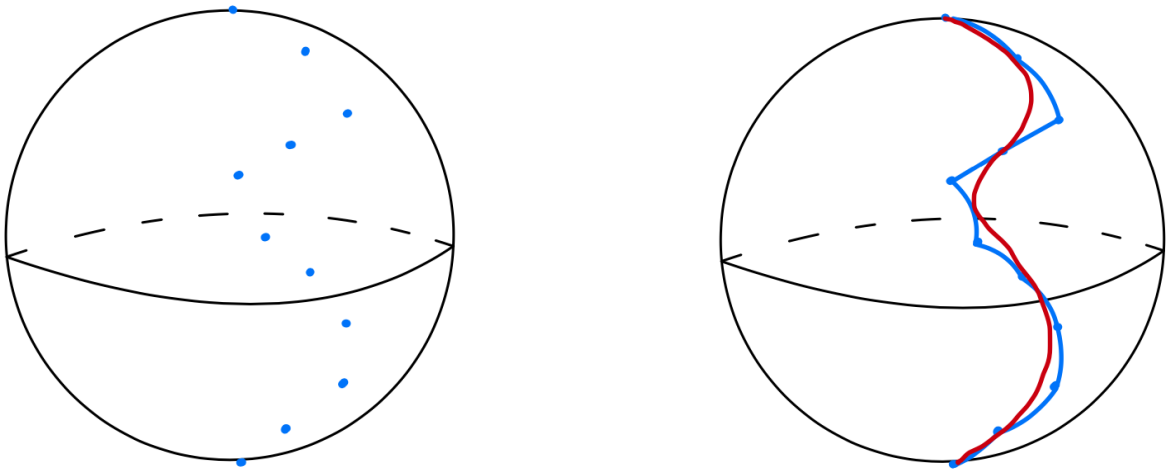
Potential applications can be in Geostatistics area, which analyzes the path of a moving object on the Earth's surface; or in the Biological area, which studies the birds' migration behavior. For example, find the mean path they fly, and predict their position in the next few days.

Contents

1 Introduction

Statistical Shape Analysis is an inter-discipline topic that requires knowledge of mathematics like differential geometry, linear algebra, and differential equations. Statistical methods include regression, probability distribution, and various multivariate statistical methods. Also, we need to have knowledge of programming languages like R and Python to perform statistical analysis and simulations.

Traditional statistical shape analysis would intensively study the actual shape of underlying objects, like the analysis of skulls of specific species of mammals or chemical atoms. On the other hand, another direction of this topic is to study the shape of the space in which the object resides. For example, an ant is walking on a toy, or a bird's flying path.



(a) Time series trajectory

(b) connected using piece-wise geodesic and best fitted smooth curve

Figure 1: Example of time series trajectory

The objective of the analysis is based on the time series trajectory[1]*, which means the observed position of the particle at successive time step. We wish to find a best fit curve.*

Because of the analysis depends on the shape of the underlying space, so further methods has been developed in this century. One of the most famous method is the method of unrolling, unwrapping, and wrapping technique.

Unrolling, unwrapping, and wrapping methods was initially proposed by Jupp and Kent

[1]In this dissertation, we refer *trajectory to be a time series of observations at successive time; and curve be a usual continuous curve*

(1987) [2]*, that is used to find a smooth curve on the sphere given the trace of discrete points. This method has been extensively studied and successfully applied to more general manifolds like Kim et al. (2021)* [3] *apply on smoothing spline on Riemannian manifold.*

Because the previous research commonly performs fitting based on a single sample fitting method on the tangent space. We would be interested in how multiple samples may increase the accuracy of the mean curve estimation on the sphere. Furthermore, we would also like to know the capability of performing predictions, including the predicted value, assumptions, and limitations.

The structure of this dissertation contains several sections: first, we will briefly review the essential differential geometry concepts and formulas in Section 2; secondly, the necessary methodologies, including the use of unrolling, unwrapping, and wrapping method, will be discussed in Section 3; In Section 4, we will list the outlines of the analysis, defining all notations and give the assumptions for later sections; In Section 5, this perform all the analysis and simulations in this dissertation, including the curve fitting methods based on different sample size; forecasting; and the discussions of the problems. Finally, there are further discussions, including possible future works to extend our findings.

For the programming part, we used Python to perform all of our simulations and statistical analyses. We used Jupyter Notebook to prepare all the contents and prepared Google Colab links for reference.

Jupyter Notebook is a powerful tool for Python developers to organize the project. The structure of the notebook are as follows: Section 1 defines all differential geometry related functions (for dissertation section 2); Section 2 defines the unrolling, unwrapping, and wrapping (dissertation section 3.1; Section 3 contains a helper function for 3D graph plotting; Section 4 performs the true curve definition, sample curves generation, and curve fitting method (dissertation section 3.4, 3.5, 4 and 5.1). The last one Section 5 is the section for predictions (dissertation section 5.2). There is also an extra Jupyter Notebook to contain standalone examples used in this dissertation.

[2] Jupp and Kent (1987) [20]: *Fitting Smooth Paths to Spherical Data*

[3] Kim et. al. (2021) [23]: Smoothing splines on Riemannian manifolds, with applications to 3D shape space

2 Differential geometry review

Differential geometry is a mathematical area that studies the geometry of smooth shapes and smooth spaces. For example, smooth curves, smooth surfaces, or other more abstract objects like Möbius strip.

The term smooth (or smoothness of a function) describes the parameterization of the underlying object that has continuous derivatives over the domain. Smoothness provides many convenience properties, such as continuity and differentiability. We can then define various methods like covariant derivative, parallel transport, and geodesic, without concern about any points of discontinuity or non-differentiable regions.

Although some statistical analysis has been developed in more general manifolds (like general Riemannnian Manifolds), due to the simple structure of the sphere and its convenience properties, this dissertation will focus on the analysis on unit sphere. So, in this section, we will first introduce each term and notation for general differential geometry and with examples for spherical cases.

This section will list formulas for each concept, using the results listed in Kim et. al. (2021) [23] p.116. Furthermore, we will give an intuitive idea and simple derivation if possible. Most results can be found in standard differential geometry textbooks like Elementary differential geometry [4] *and Differential geometry of curves and surfaces* [5].

2.1 Manifolds and curves

2.1.1 Manifolds

A manifold M is a topological space that is locally Euclidean space. That is a generic term to describe the surface of an object. A manifold can be high dimensional. In particular, for a 2-dimensional case, the manifold is called a surface (living in a 3-dimensional space). By definition, M is called a differentiable manifold in $\mathbb{R}^3$, if $\forall p \in M, \exists$ open neighborhood V of M at p, an open set $U \subset \mathbb{R}^2$, and a smooth map $f : U \to V \cap M$. We denote the tuples (U, f, V) be the local parameterization of M.

[4] Pressley (2010)[33] : Elementary differential geometry
[5] Do Carmo (2016)[9]: Differential geometry of curves and surfaces

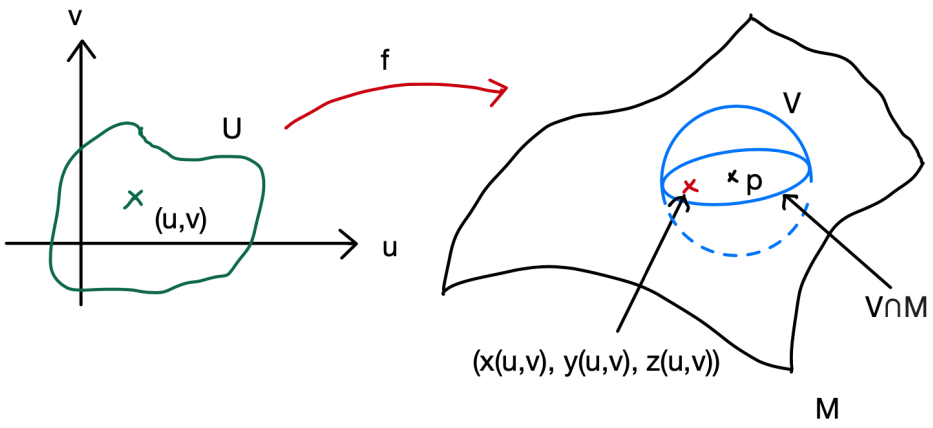


Figure 2: Local parameterization of a differential manifold

The dimension of a manifold can be determined by the number of parameters of the mapping f

For example, the surface of the sphere, S^2 (assumed radius r and centered at the origin) is a 2-dimensional manifold because, for each point $p = (x, y, z) \in S^2$, we can define a local parameterization:

1. *local Euclidean coordinates $U = \{(u, v)|u \in [0, \pi], v \in [0, 2\pi]\}$*
2. *an open set V of S^2 at the point p*
3. *a mapping $f : U \to V \cap S^2$, with $f(u, v) = (rsin(u)cos(v), rsin(u)sin(v), rcos(u))$*

Since f has two inputs u and v, therefore the surface of the sphere S^2 is a 2-dimensional manifold.

In this dissertation, we will simply refer the surface of the sphere as Sphere if it is clear in context.

2.1.2 Curves

A curve is defined in a similar way in Euclidean spaces.

Let I be an interval $\subset \mathbb{R}$. let $\gamma : I \to M$ as a parameterized curve, mapped from an interval of $\mathbb{R}$ to M.

For example, to define an arc on S^2, starts from the North pole (0,0,1) and ends at the "East" of the equator (1,0,0), i.e., a quarter arc viewed on the x-z plane let $\gamma : [0, \pi/2] \to S^2$

such that $\gamma(t) = (sin(t), 0, cos(t))$.

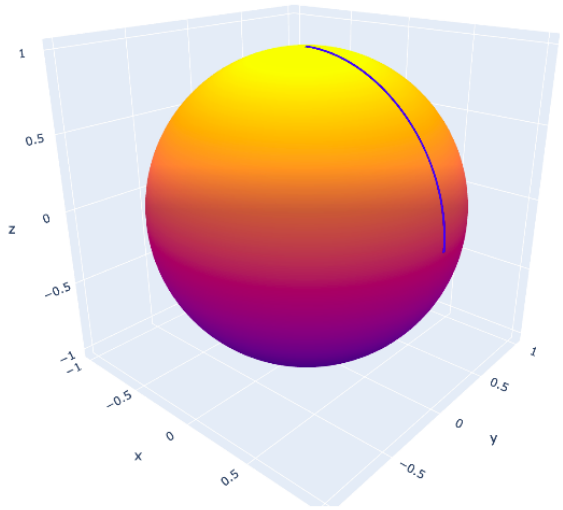


Figure 3: simple curve on the sphere

2.2 Tangent space

The concept of tangent spaces plays a vital role in deriving parallel transport, geodesic, and exponential map. All statistical analyses conducted in this dissertation rely heavily on tangent space.

Tangent space is a vector space of all tangent vectors of M *at point* $p \in M$*, denoted as* T_pM*. The dimension of the tangent space equals the number of basis vectors in the vector space (it can be shown that the dimension of tangent space is always equal to the dimension of the manifold).*

Furthermore, suppose there is a parameterized curve $\gamma : (-\epsilon, \epsilon) \to M$*, with* $\gamma(0) = p$ *and initial velocity* $\gamma'(0) = w$*. Then, we can also define a tangent space to the manifold* M *at the contact point* $\gamma(0)$*, where the components of the linearly independent vectors of* $\gamma'(0)$ *spans this tangent space.*

For example, for the sphere, because S^2 *is a smooth manifold, there is a local parameterization* (U, f, V) *at* p*, such that* U *has basis vectors* $\{u, v\}$*. The basis vectors of the tangent space are* $(\frac{\partial f}{\partial u}$ *and* $\frac{\partial f}{\partial v})$*. Thus,* w *can be expressed as the linear combination of these basis vectors by* $w = af_u + bf_v$*, where* $f_u = \frac{\partial f}{\partial u}$*,* $f_v = \frac{\partial f}{\partial v}$*. Here, we can see that the tangent space*

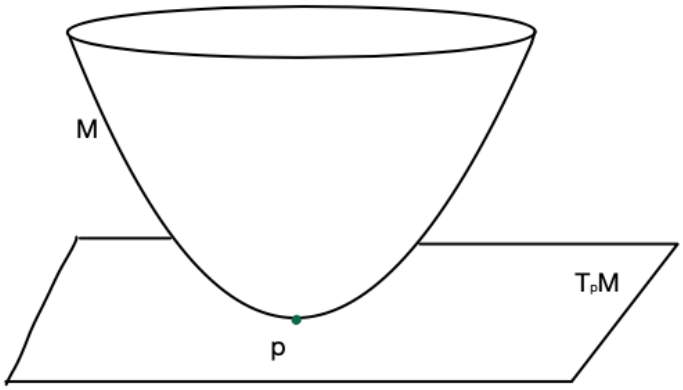


Figure 4: Tangent space

of S^2 at p has 2 basis vectors, so we also call this a tangent plane.

2.3 Covariant derivative

Covariant derivative is an important concept in deriving parallel transport and geodesic. Suppose there is a curve γ on the manifold M, and a vector field [6] w, assigning each points on the manifold M a tangent vector. The covariant derivative is defined as the normal projection of the derivative of the vector field restricted to the curve γ on the tangent plane.

Mathematically, first, let the curve $\gamma : I \to M$, and by local parameterization, so that points on M can be written as $f(u, v)$, then we can rewrite $\gamma(t) = f(u(t), v(t))$ Secondly, we assign the vector field w as the tangent vector to the manifold, i.e., $w(\gamma(t)) = w(t) = \gamma'(t) \forall t \in I$. Because $w(t)$ is a tangent vector to the manifold M, so $w(t) \in T_pM$, and hence it can be rewritten as the linear combination of the basis vectors of the tangent space $\{f_u, f_v\}$ gives

$$w(t) = a(t)f_u + b(t)f_v$$

So, by differentiating both sides with respect to t, we obtain

$$\begin{aligned}\frac{dw(t)}{dt} = &\frac{da(t)}{dt}f_u + a(t)[f_{uu}\frac{du(t)}{dt} + f_{uv}\frac{dv(t)}{dt}] \\ &+ \frac{db(t)}{dt}f_v + b(t)[f_{vu}\frac{du(t)}{dt} + f_{vv}\frac{dv(t)}{dt}]\end{aligned}$$

[6] Vector Field is a map that assigns each point on the manifold a vector.

To simplify this, first note that f_{uu}, f_{uv}, and f_{vv} are 3d vectors, so each of them can be written as the linear combinations of f_u, f_v, and a normal vector N that is perpendicular to the tangent plane. i.e.,

$$f_{uu} = \Gamma^1_{11} f_u + \Gamma^2_{11} f_v + L_1 N$$
$$f_{vv} = \Gamma^1_{22} f_u + \Gamma^2_{22} f_v + L_2 N$$
$$f_{uv} = f_{vu} = \Gamma^1_{12} f_u + \Gamma^2_{12} f_v + L_3 N$$

where the symbol Γ^k_{ij} is known as the Christoffel Symbols, which can be determined by performing inner-product[7] to each above equations by $f_u, f_v respectively$; and using the fact that N is orthogonal [8] to the tangent plane (i.e. $< N, f_u >=< N, f_v >= 0$.

Hence, by combining all these equations, by projecting those vectors onto the tangent plane (performing orthogonal projection of $\frac{dw(t)}{dt}$ onto the tangent plane), this ends up with the definition of the covariant derivative, denoted as $\frac{Dw(t)}{dt}$:

$$\frac{Dw(t)}{dt} = \frac{dw(t)}{dt} - < \frac{dw(t)}{dt}, N > N$$

It can then be written as

$$\begin{aligned}\frac{Dw(t)}{dt} &= [a'(t) + \Gamma^1_{11} a(t)u'(t) + \Gamma^1_{12} a(t)v'(t) + \Gamma^1_{12} b(t)u'(t) + \Gamma^1_{22} b(t)v'(t)] f_u \\ &\quad + [b'(t) + \Gamma^2_{11} a(t)u'(t) + \Gamma^2_{12} a(t)v'(t) + \Gamma^2_{12} b(t)u'(t) + \Gamma^2_{22} b(t)v'(t)] f_v\end{aligned}$$

If the covariant derivative $\frac{Dw(t)}{dt} = 0 \forall t \in I$, then this vector field w is called a parallel vector field.

2.4 Parallel transport

Parallel transport is the process of transporting a vector on the manifold along a smooth curve that keeps the vector "parallel in some sense."

[7] Inner-product can be expressed as $< a, b >$ for vectors a and b, and this inner-product $= 0$ if a and b are *orthogonal* (perpendicular in usual 3d Euclidean space)

[8] Orthogonal: in usual 3d Euclidean space, this is also known as perpendicular

Imagine an ant is standing on the North Pole. It needs to keep walking in a specific direction. So, at any time, from the ant's perspective, it is always pointing in the same direction. However, for an observer viewing the ant, its direction changes from horizontal to vertical downward. Suppose further that the ant needs to walk along the equator in a parallel sense. It is like moving to the right. For an observer outside of the Earth, the ant is moving in the direction pointing downwards (See Figure 5).

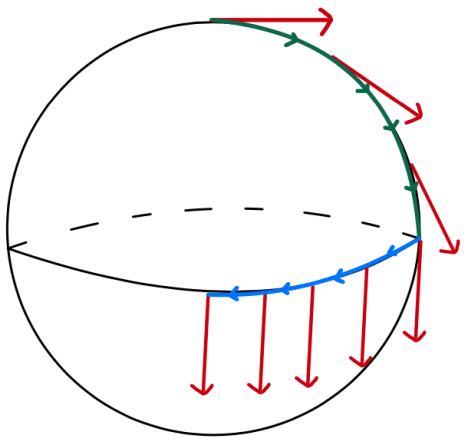


Figure 5: Parallel Transport

We denote the notation of parallel transporting a vector along a curve $\gamma(t)$ *from* $t = t_0$ *to* $t = t_1$ *is a map* [9] $P^{\gamma(t_1)}_{\gamma(t_0)} : T_{\gamma(t_0)}M \to T_{\gamma(t_1)}M$. *And this is defined as a vector* $w(t_0)$ *along the smooth curve* γ *on a parallel vector field with* $\frac{D\gamma'(t)}{dt} = 0 \ \forall t \in [t_0, t_1]$. *Note that the inverse of this map is to reverse the starting and ending points.*

$$(P^{\gamma(t_1)}_{\gamma(t_0)})^{-1} = P^{\gamma(t_0)}_{\gamma(t_1)}$$

We shall further define the parallel transport as an identity map if the starting point and ending point are the same.

$$P^{\gamma(t)}_{\gamma(t)}(x) = x \ \forall t \in I \text{ and } x \in M$$

For the derivation of parallel transport using the above definitions on the sphere, please

[9] This mapping is also known as Levi-Civita connection

see Appendix A.1.

In general, parallel transport does not have a closed-form solution. There are ways to solve parallel transport numerically [10].

For the spherical case, there is a simple close form [11] *to transport a vector* w *along a curve connecting 2 distinct points on the sphere.*

$$w(t) = w - < w, v > (v - \gamma'(t))$$

2.5 Unit speed geodesic

To find the geodesic curve γ *on the manifold* M*, we need to find the curves that the tangent vector* $\gamma'(t)$ *is parallel along* γ $\forall t$*, i.e.*

$$\frac{D\gamma'(t)}{dt} = 0 \ \forall t$$

In other words, we need to find curves where the acceleration vector $\gamma''(t)$ *is normal to the surface* $\forall t$*. Sometimes, it may be hard to compute for general manifold (or do not have closed-form). In this case, we also needed to use numerical methods to solve.*

For the sphere case, there is a simple closed-form solution for the geodesic. It used the fact that the arc joining the points on the great circle of the sphere would be the shortest distance. Thus, by the symmetry property of the sphere, we can conclude that the great circles of the sphere S^2 *are geodesics.*[12]

And the closed-form solution is, given the starting point x *and the direction vector* v*, the unit speed geodesic curve is*

$$\gamma(t) = xcos(t) + vsin(t) \tag{2.1}$$

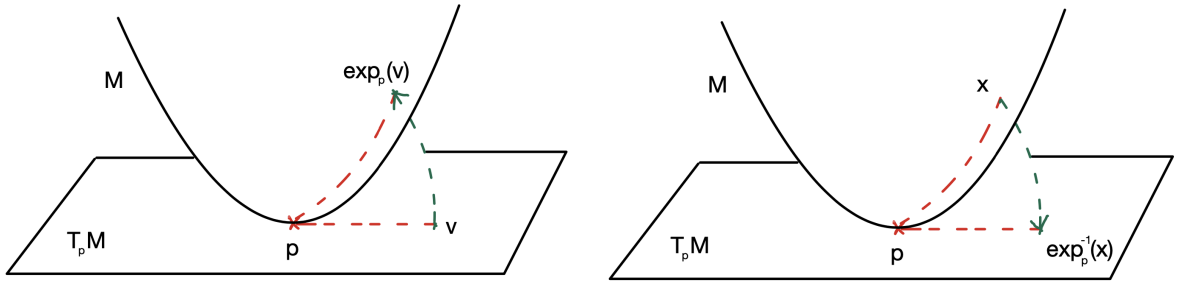
[10] see [26] Parallel transport in shape analysis: a scalable numerical scheme

[11] see [23] Kim et. al. (2021) Smoothing splines on Riemannian manifolds, with applications to 3D shape space, p.116. We will also use the results of parallel transport/ geodesic/ exponential map and inverse exponential map from this research paper

[12] See Do Carmo (2016) [9]: Differential Geometry of Curves & Surfaces, p.249

2.6 Exponential map and inverse exponential map

The exponential map and the inverse exponential map (some textbooks call the inverse exponential map as logarithm map) can be treated as the inverse operation of each other. Suppose the contact point to the manifold and the tangent space be $p \in M$.



(a) Exponential map (b) Inverse exponential map

Figure 6: Exponential map and inverse exponential map

An exponential map is used to map a vector v *from the tangent space* T_pM *to the manifold* M *(say* x*).*

$$x = exp_p(v) \tag{2.2}$$

In the opposite, the inverse exponential map is mapping a point on the manifold $x \in M$ *to the tangent space* T_pM *(say* v*).*

$$v = exp_p^{-1}(x) \tag{2.3}$$

See also Kim et. al. (2021) p.116; there are close form solutions for the exponential map and the inverse exponential map for sphere S^2.

$$exp_p(v) = pcos(||v||) + \frac{v}{||v||}sin(||v||) \tag{2.4}$$

$$exp_p^{-1}(x) = \frac{x - < p, x > p}{||x - < p, x > p||}cos^{-1}(< p, x >) \tag{2.5}$$

2.7 Non-unit speed case

Recall from Section 2.4, 2.5, those rely on the unit speed geodesic. If we take time information into account, the unit speed geodesic may not necessarily hold for the trajectory of a particle

on the sphere. In elementary Physics, we have learned that the speed is equivalent to the distance traveled per unit time [13]*. However, for the time series data that is observed in every fixed interval (say observed at* $t = 0, 1, 2, ...$*), and the speed must be 1, then the distance traveled is always 1, which may not make sense in real-life situations. So, instead of having unit-speed geodesic, we should have constant-speed geodesic to fit the situation.*

A. Constant speed geodesic

Suppose a particle is moving from p_i *to* p_j *(in direction* v*, which is the original unit speed), which takes time T. Then we can calculate the geodesic distance by using the inner product.*

Let the included angle between p_i *and* p_j *be* θ*, and on the unit sphere,* $||p_i|| = ||p_j|| = 1$*, then*

$$< p_i, p_j > = ||p_i||||p_j||cos(\theta)$$
$$\theta = cos^{-1}(< p_i, p_j >)$$

And so, using arc-length s *on the circle with angle* θ *on the sphere with radius 1, we have*

$$s = 2\pi(1) * \theta/2\pi$$
$$= \theta$$

Hence, the constant speed geodesic can be re-scaled to

$$\gamma(t) = p_i cos(t\frac{s}{T}) + vsin(t\frac{s}{T}) \tag{2.6}$$

Note that by using elementary calculus, we can get

$$\gamma'(t) = \frac{s}{T}[-p_i sin(t\frac{s}{T}) + vcos(t\frac{s}{T}] \tag{2.7}$$

By taking the Euclidean norm on both side, we can get $|\gamma'(t)| = |\frac{s}{T}|$*, which is a constant.*

[13] speed v = d / t

B. Parallel Transport

Recall from section 2.4, the parallel transport for sphere based on unit speed geodesic is

$$w(t) = w - < w, v > (v - \gamma'(t))$$

So for constant speed case, by re-scaling the velocity v *to* $v\frac{s}{T}$*, and the equation 2.7, the parallel transport changes to*

$$w(t) = w - < w, v > (v - \gamma'(t))\frac{s}{T} \tag{2.8}$$

3 Methodologies review

This section introduces required notations and assumptions for section 4 and 5.

3.1 Unrolling, unwrapping, and wrapping

Unrolling, unwrapping, and wrapping method is a useful technique to find a smooth curve given a set of discrete points on the manifold. Suppose there is a particle moving on the manifold, and we observe the particle's position at discrete time. The method of unrolling, unwrapping, and wrapping provides an iterative approach to fit a smooth curve onto the manifold.

The approach has been discussed extensively by Jupp and Kent [20]. A similar diagram for demonstrating the unrolling, unwrapping, and wrapping can be found in Kim et. al. (2020) Figure 2.

To demonstrate the process using the unrolling, unwrapping and wrapping. For example, suppose given that there is a curve (see Figure 7 colored with red wet ink) on the sphere.

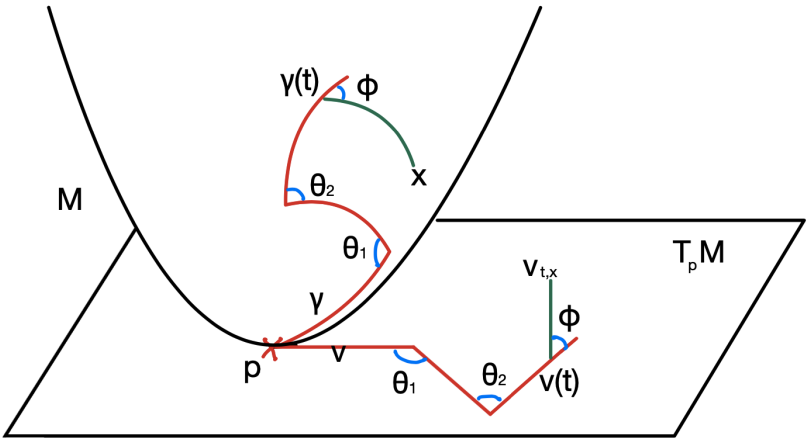


Figure 7: Example of unrolling, unwrapping, and wrapping

- *The unrolling stage is to roll a ball onto the floor (in an honest way, not sliding and twisting) such that when rolling the ball, the ball is always touching the floor. So, after completely rolling the ball along the curve, an ink trace will be on the floor.*
- *The wrapping stage is to make points on the sphere (may or may not be on the curve), and project onto the floor, such that distance to the curve and angle to the curve are*

preserved.

- *The wrapping stage projects points on the floor back to the sphere (a reverse process of unwrapping).*

The overall procedures for unrolling, unwrapping, and wrapping rely heavily on using parallel transport, geodesic, exponential map, and inverse exponential map. To process each method mathematically, first define the notations.

Suppose that a particle is moving on the manifold along a specific (unknown true) curve $\gamma(t)$*, where* $t \in \mathbb{R} \cup \{0\}$*. Now we observed the moving particle on the manifold at discrete consecutive times* $t_0 = 0, t_1, ..., t_n = T$ *for some* $n \in \mathbb{Z}$ *and* $t_1 - t_0 = t_2 - t_1 = ... = t_n - t_{n-1} = \Delta t$*. so we have a time series of position of the particle* $\gamma(0), \gamma(t_1), ..., \gamma(t_n)$*.*

3.1.1 Unrolling

To perform unrolling, first, we need to have an initial curve for unrolling (also called a base path or base curve).

In the initial step, we used the piece-wise geodesic curve to join consecutive points on the manifold. For instance, we define $g_i(t)$ *is the geodesic curve connecting* $\gamma(t_{i-1})$ *and* $\gamma(t_i)$*, and define* $g(t)$ *be the base curve in such a way that*

$$g(t) = g_i(t)\mathbb{1}\{t \in [t_{i-1}, t_i]\}\forall i = 1, 2, ..., n$$

That is the initial base curve for the unrolling stage.

After that, for the actual unrolling procedure, we use the concept of parallel transport of a vector and the inverse exponential map to map the point on the manifold to the tangent space.

First, denote the starting point is at $p = g(t_0) = g(0)$*, and denote the tangent space to the manifold* M *at* p *be* T_pM*, and also denote the contact point at the tangent space be the original.*

$$v(t_0) = 0 \in T_{g(t_0)}M$$

For the point at $t = t_1$: $g(t_1)$, we can directly perform the inverse exponential map $exp^{-1}_{g(t_0)}(g(t_1))$ to map this point onto the tangent space.

$$v(t_1) = exp^{-1}_{g(t_0)}(g(t_1))$$

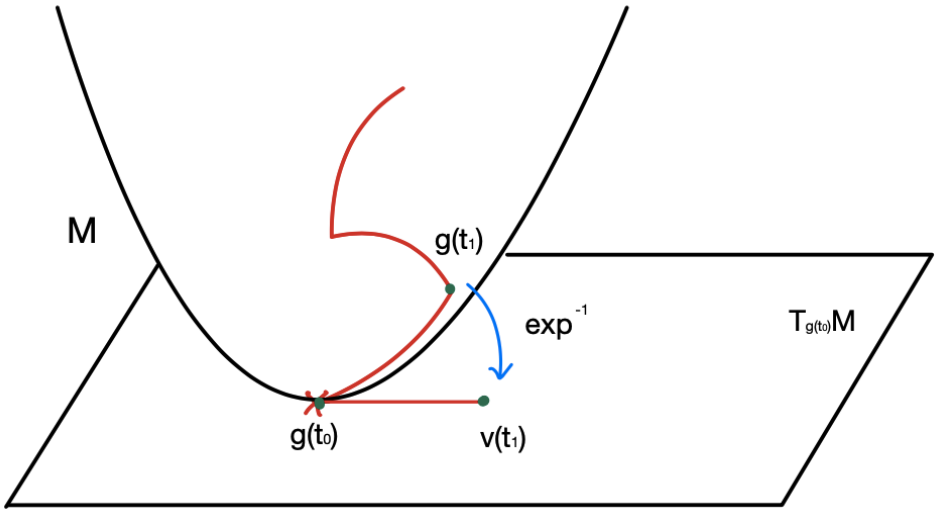


Figure 8: Unrolling for g up to the $g(t_1)$

For the point at $t = t_2$: $g(t_2)$, we have 3 steps to unroll the curve between $g(t_1)$ to $g(t_2)$ to the tangent place (demonstrate in Figure 9):

1. *Determine the tangent space to the manifold M at $g(t_1)$, $T_{g(t_1)}M$, and get the tangent vector on $T_{g(t_1)}M$ by inverse exponential map of $g(t_2)$, that is $exp^{-1}_{g(t_1)}(g(t_2))$*
2. *Perform parallel transport $P^{g(t_0)}_{g(t_1)}$, to transport the tangent vector found in (1), from $g(t_1)$ to $g(t_0)$, so the vector is now in the tangent space (pointing out from the contact point $g(t_0)$*
3. *Performs a Euclidean vector translation to translate the vector to pointing out from $v(t_1)$*

$$v(t_2) = v(t_1) + P^{g(t_0)}_{g(t_1)}(exp^{-1}_{g(t_1)}(g(t_2))$$

To proceed the same procedure, in general, we end up with the relationship:

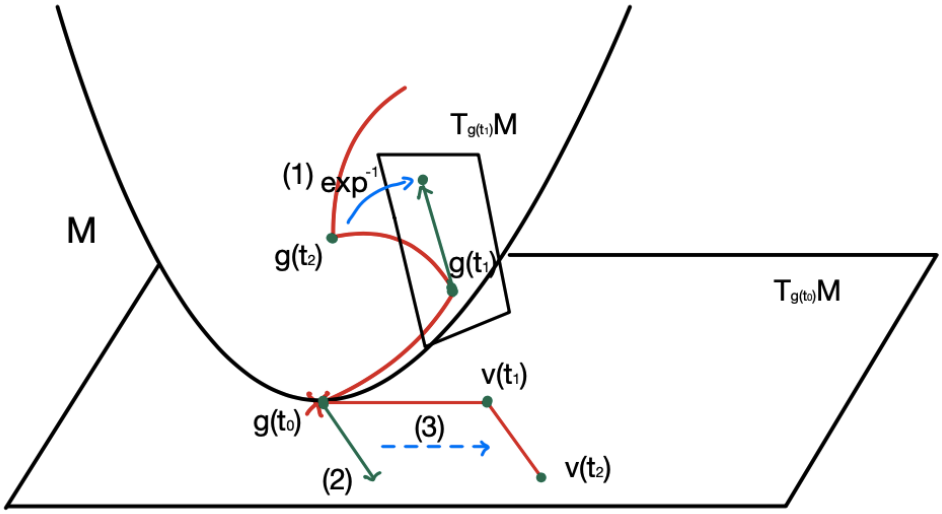


Figure 9: Unrolling for g up to the $g(t_2)$

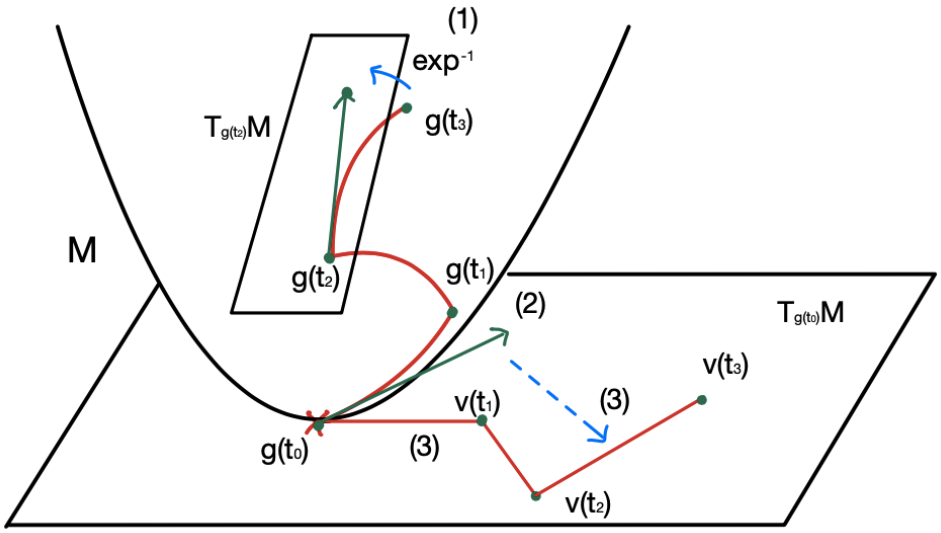


Figure 10: Unrolling for g up to the $g(t_3)$

$$v(t_i) = \begin{cases} 0 & for\ i = 0 \\ v(t_{i-1}) + P^{g(t_0)}_{g(t_1)} \circ P^{g(t_1)}_{g(t_2)} \circ ... \circ P^{g(t_{i-2})}_{g(t_{i-1})}(exp^{-1}_{g(t_{i-1})}(g(t_i)) & \forall i \in 1, 2, ..., n \end{cases} \tag{3.1}$$

So from the tangent space, we have the points $v(0) = 0, v(t_1), v(t_2), ..., v(t_n) \in \mathbb{R}^d$*, where* $d = dim(M)$*. After we obtained these discrete points on the tangent space, we can easily perform linear interpolation to join successive points to construct a continuous function of*

$v(t)$ *by*

$$v(t) = \begin{cases} v(t_i) & t = t_i \\ v(t_{i-1}) + \frac{t-t_{i-1}}{t_i - t_{i-1}}(v(t_i) - v(t_{i-1})) & t_{i-1} < t < t_i \end{cases} \tag{3.2}$$

3.1.2 Unwrapping

Suppose there is a point x *on the manifold at time* t *where* $t_{i-1} < t < t_i$*, the way to make this point* x *onto the tangent space* $T_{g(t_0)}M$ *are similar to unrolling process, but considering an extra parallel transport from* $g(t)$ *to* $g(t_{i-1})$*, and then perform the same argument as unrolling process. For instance, suppose at* t *where* $t_{i-1} < t < t_i$*,*

1. *Determine the tangent space to the manifold* M *at* $g(t)$*,* $T_{g(t)}M$*, and get the tangent vector on* $T_{g(t)}M$ *by inverse exponential map of* $g(t_i)$*, that is* $exp^{-1}_{g(t)}(g(t_i))$*.*
2. *Perform parallel transport* $P^{g(t_{i-1})}_{g(t)}$*, to transport the tangent vector found in (1).*
3. *Perform parallel transport from* $g(t_{i-1})$ *to* $g(t_{i-2})$*, then from* $g(t_{i-2})$ *to* $g(t_{i-3})$*, ..., until transport to* $g(t_0)$*, so the vector is now in the tangent space* $T_{g(t_0))}M$ *(pointing out from the contact point* $g(t_0)$*.*
4. *Using the equation 3.2 to get the position of* $v(t)$*, perform a Euclidean vector translation to translate the vector to pointing out from* $v(t)$*.*

Figure 11 demonstrate the unwrapping of $x \in M$ *at* t*, where* $t_2 < t < t_3$*.*

Mathematically, let we call this vector on the tangent space be $v_{t,x}$

$$v_{t,x} = \begin{cases} exp^{-1}_{g(t_0)}(x) & t = t_0 = 0 \\ v(t) + P^{g(t_0)}_{g(t_1)} \circ P^{g(t_1)}_{g(t_2)} \circ ... \circ P^{g(t_{i-1})}_{g(t)}(exp^{-1}_{g(t)}(x)) & t_{i-1} < t < t_i \end{cases} \tag{3.3}$$

In the special case of $t = t_i$*, then we denote that the parallel transport* $P^{g(t)}_{g(t)}$ *is an identity map, so it takes no effect on transporting vector* $P^{g(t)}_{g(t)}(x) = x$*. It means that there is no need to perform extra parallel transportation (the right-most transport in equation 3.3).*

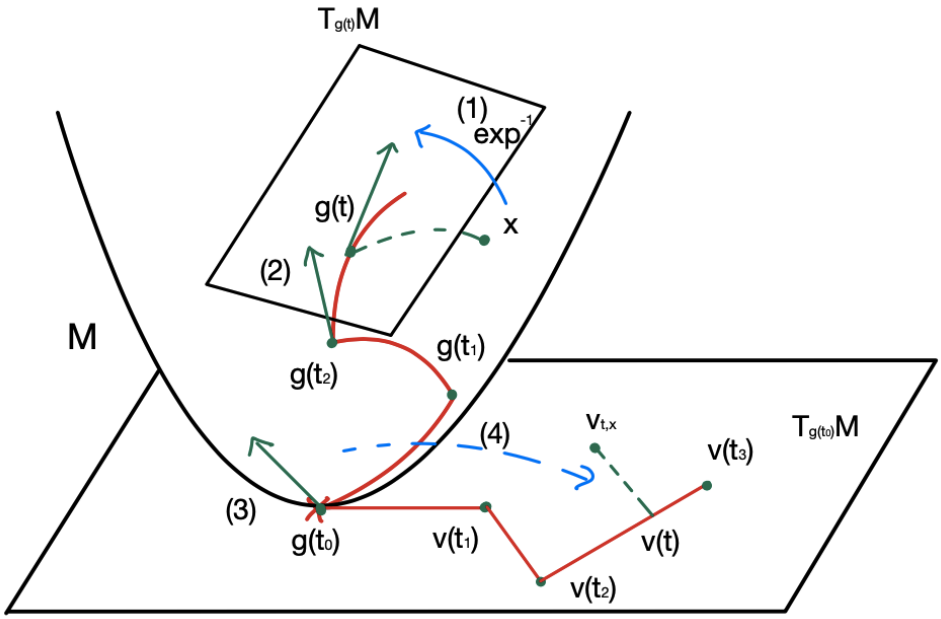


Figure 11: Unwrapping point on M at time t

3.1.3 Wrapping

Wrapping is the reverse process of unwrapping that maps back a vector on the tangent space to the manifold M. Suppose there is a tangent vector $v_{t,x} \in T_{g(0)}M$ at time $t \in [0, T]$, then simply like the change of subject of 3.3, that:

$$x = \begin{cases} exp_{g(t_0)}(v_{t,x}) & t = t_0 = 0 \\ P^{g(t)}_{g(t_{i-1})} \circ P^{g(t_{i-1})}_{g(t_{i-2})} ... \circ P^{g(2)}_{g(1)} \circ P^{g(1)}_{g(0)}(v_{t,x} - v(t)) & t_{i-1} < t < t_i \end{cases} \tag{3.4}$$

then the point $x \in M$ is the wrapped point on the manifold

For instance, suppose at t where $t_{i-1} < t < t_i$,

1. *Using the equation 3.2 to get the position of $v(t)$, then calculate the vector pointing from $v(t)$ to $v_{t,x}$ (i.e. $v_{t,x} - v(t)$). After that, perform a Euclidean vector translation to translate the vector to pointing out from $v(0)$ (i.e., the contact point $g(t_0)$).*
2. *Perform parallel transport the vector in step(1), from $g(t_0)$ to $g(t_1)$, $g(t_1)$ to $g(t_2)$, , ..., until transport to $g(t_{i-1})$.*
3. *Perform parallel transport $P^{g(t)}_{g(t_{i-1})}$, to transport the tangent vector in step(2).*
4. *Determines the tangent space to the manifold M at $g(t)$, $T_{g(t)}M$, and use exponential*

map of to get the point on the manifold M $exp^{-1}_{g(t)}(v_{t,x} - v(t))$.

Figure 12 demonstrate the unwrapping of $x \in M$ *at* t*, where* $t_2 < t < t_3$.

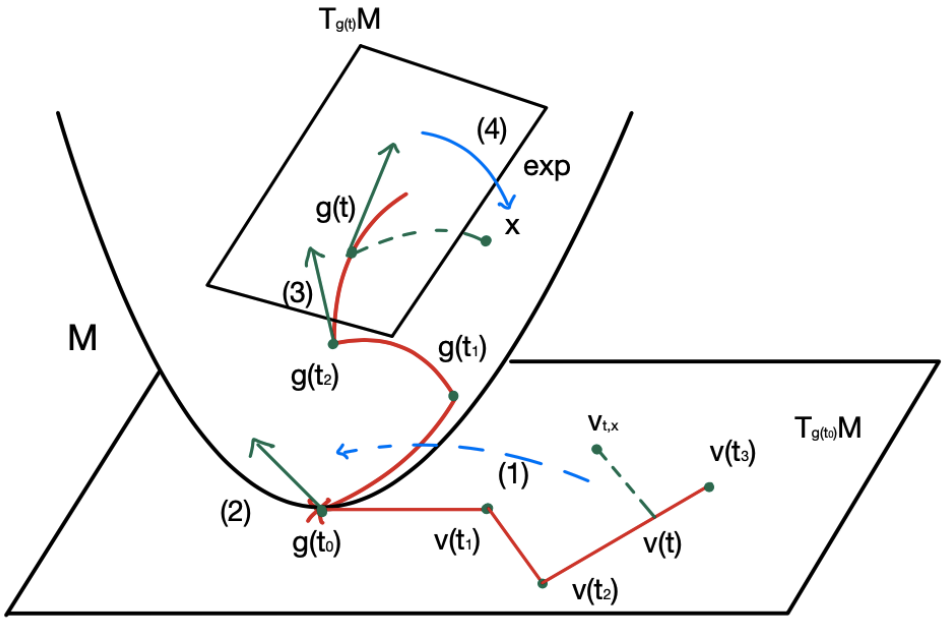


Figure 12: Wrapping point from tangent space to the manifold M at time t

3.2 Cubic Smoothing Spline

The Cubic Smoothing Spline method is a regression method to fit a set of data under smoothness regularization. This method will be used in section 5.1.1.

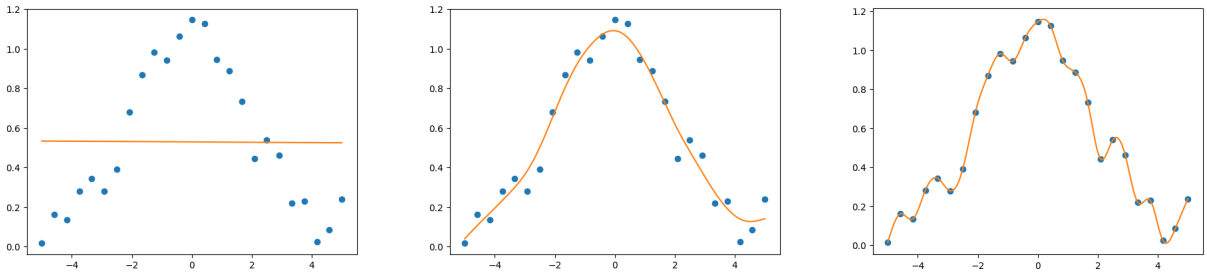
Let $\{x_i, y_i\}_{i=1}^n$ *be n pairs of data where* $x_i, y_i \in \mathbb{R}$*, assumed that the true model is*

$$y_i = f(x_i) + \epsilon_i \ \forall i \in 1, 2, ..., n$$

where $f(.)$ *is an smooth function, and the errors are mutually independent with mean 0 and constant variance* σ^2 *(also an unknown).*

The method to estimate this function $f(.)$ *is to find a function* f *that minimizes the error sum of square, with an additional penalization term.*

$$\hat{f}(x) = \underset{f}{argmin} \ [\lambda \sum_{i=1}^{N} (y_i - f(x_i)) + (1 - \lambda) \int f''(x)^2 dx] \tag{3.5}$$



(a) smoothness=0 (b) smoothness=0.8 (c) smoothness=1

Figure 13: Cubic Smoothing Spline with various smoothness parameter

The first term of this equation is the ordinary least square, while the second term controls the curvature of the function. The parameter λ is used to control the smoothness of the resulting function $\hat{f}(x)$.

- *If $\lambda = 0$, then the first term of Equation 3.5 is ignored, the minimum will be a zero function trivially.*
- *If $\lambda = 1$, then the second term for controlling the curvature term is ignored, and so it is the same as interpolating every point.*

See Figure 13 for various values of smoothness parameter [14]*. In general, we expect to have the smoothness parameter λ between 0 and 1 that balances both the smoothness of the curve and the importance of the datasets.*

3.3 Gaussian Processes Regression (GPR)

This section will provide a brief introduction to the Gaussian Processes Regression.

Gaussian Processes Regression [15] *is a non-parametric regression method that is a popular method in statistics and machine learning applications. GPR has been widely used because it provides a powerful probabilistic structure, not only is able to make predictions but also provides uncertainty measures over the predictions.*

[14] Similar code can be found on *csaps official website: https://csaps.readthedocs.io/en/latest/*

[15] See [14]: Bayesian Data Analysis, 3rd edition, p.501; and [40], Jie Wang (2020) An intuitive Tutorial to Gaussian Processes Regression

3.3.1 Gaussian Processes

Gaussian Processes rely on the Multivariate Normal Distribution (MVN distribution). Recall that a random vector $\mathbf{x} = (x_1, x_2, ..., x_d)^T$ *is called a d-dimensional MVN distribution if and only if* $\mathbf{a}^T\mathbf{x}$ *is a (univariate) normal distribution* $\forall \mathbf{a} \in \mathbb{R}^d$*. If the random vector* $\mathbf{x}$ *is a MVN with*

- *Mean vector* $\mu = \mathbf{E}(\mathbf{x}) \in \mathbb{R}^d$
- *Covariance matrix* $\mathbf{\Sigma} = \mathbf{E}[(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T] \in \mathbb{R}^{dxd}$

Then we write

$$\mathbf{x} \sim N_d(\mu, \mathbf{\Sigma}) \tag{3.6}$$

For Gaussian processes (GP)[16]*, it generalized to model random function. The Gaussian process prior on* μ *defines as a random function, having mean function* $\mathbf{m}$ *and covariance function (Kernel)* K*, denoted as*

$$\mu \sim GP(\mathbf{m}, K) \tag{3.7}$$

For which when drawing at specified points $x_1, x_2, ..., x_d$*, the random vector* $(\mu(x_1), ..., \mu(x_d))^T$ *has a d-dimensional MVN, with mean vector* $(m(x_1), ..., m(x_d))^T$ *and covariance matrix* $K(x_i, x_j)$ $\forall i, j = 1, 2, ..., d$*.*

$$(\mu(x_1), ..., \mu(x_d))^T \sim N_d((m(x_1), ..., m(x_d))^T, K(x_1, ..., x_d)) \tag{3.8}$$

There are a few commonly used covariance function (Kernel). For example,

1. *Radial basis function (RBF) kernel*

   RBF is a popular kernel that is used to approximate the covariance structure of the data

   $$K(x_i, x_j) = \tau^2 exp(-\frac{|x_i - x_j|^2}{l^2}) \tag{3.9}$$

   Where τ *controls the magnitude and* l *controls the smoothness, and* $|x_i - x_j|^2$ *is the square of Euclidean distance between* x_i *and* x_j*.*

[16] See [14]: Bayesian Data Analysis, 3rd edition, p.501

2. *White Noise kernel*

   When combined with another kernel (like RBF kernel), the white noise kernel can estimate the global noise level of data [17].

$$K(x_i, x_j) = \begin{cases} noise\_level & x_i == x_j \\ 0 & otherwise \end{cases} \tag{3.10}$$

There are other kernels that are popular in practice, like Matérn kernel and Exp-Sine-Squared kernel [18].

In this dissertation, we will use the combination of the Radial basic function kernel and White Noise Kernel as our kernel.

3.3.2 Gaussian Processes Regression

Suppose given the observation $D = \{(\mathbf{x_i}, y_i)\}_{i=1}^n$, *where* $\mathbf{x_i} \in \mathbb{R}^d$ *and* $y_i \in \mathbb{R}$, *and assumed the data is in the form*

$$y_i = f(\mathbf{x_i}) + \epsilon_i \tag{3.11}$$

Where $f(.)$ *is an unknown random function The Gaussian Processes Regression finding the distribution of the random function* $f(.)$ *given the data* $\{(\mathbf{x}, y_i)\}$.

Indeed, it can be shown that, if the prior is a GP, given the observations, then the posterior distribution of $f|D$ *is also a GP* [19].

For the implementation, we will make use of the Scikit-learn package to perform GPR.

Example of GPR

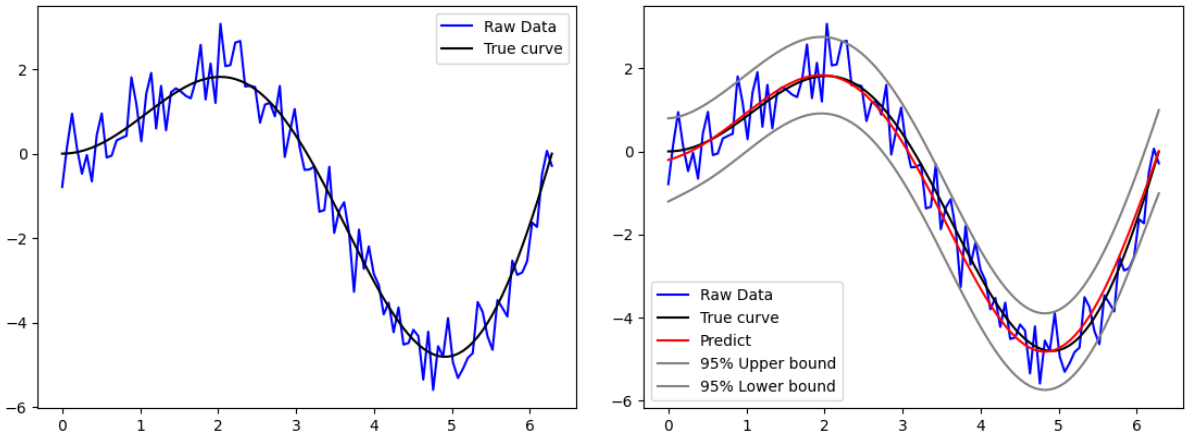
Suppose that the given data is in the relationship

$$y = xsin(x) + \epsilon$$

Where $x \in [0, 2\pi], \epsilon \sim N(0, 0.5)$.

[17] See Scikit-learn documentation section 1.7.1: https://scikit-learn.org/stable/modules/gaussian_process.html
[18] see also Scikit-learn documentation section 1.7.5
[19] See Wang (2020) [40]: An intuitive tutorial to Gaussian Processes Regression, p.10

We wish to estimate the function $f(x) = xsin(x)$. After fitting the data and using the kernels (RBF kernel + WhiteNoise kernel), we can get the mean prediction and the uncertainties of the model.



(a) Raw data (b) Fitted using GPR

Figure 14: Gaussian Processes Regression example

3.3.3 Use GPR to forecast future value

Once we have trained the GP, we can then try to predict the future value.

Example

Continue to the previous example, and predict the value after 2π.

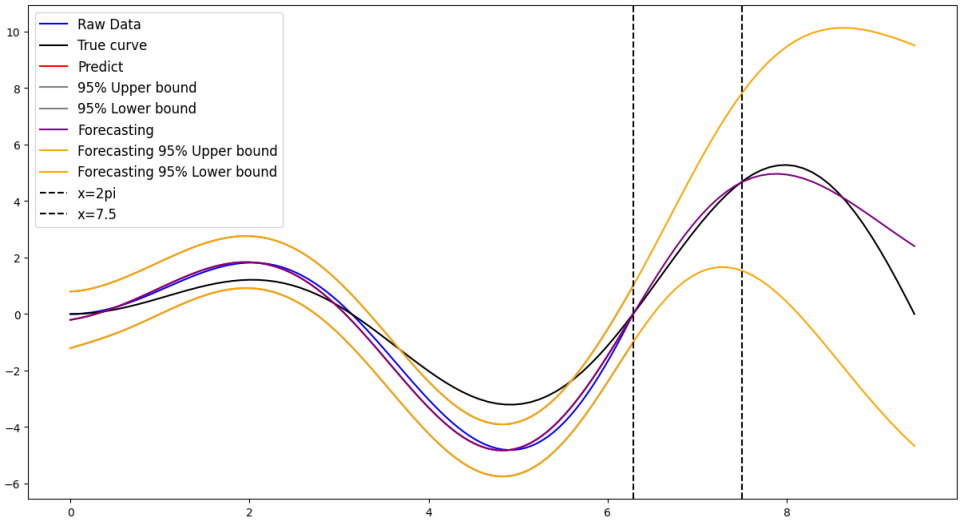


Figure 15: Example of GPR with forecasting

We can see that the new predicted function works well from $x \in [0, 2\pi]$, and this is also quite well around $[2\pi, 7.5]$. But after that, the fitted mean and the true curve are not closed. Also, the uncertainty bound is very wide.

We need to note that, as discussed in Planas, Oune and Bostanabad (2020) [20] *stated that the predictive power of GPs significantly deteriorates in extrapolation. So, in general, even though GP is able to make predictions of future value, it is impossible to make reliable long-term forecasting.*

In this dissertation, we will perform prediction (extrapolation) up to 5 time steps ahead. Also, we will discuss the behavior of the predictive ability.

3.4 Centroid of data on the manifold

This part will be used when dealing with samples of trajectories for unrolling. This is because, for curve fitting for those sample time series trajectories, we need to have a unified base path for unrolling. So, we need to determine an initial base path using a suitable criterion.

Suppose that there are n data points $x_1, x_2, ..., x_n \in M$. There are two common methods to deal with the centroid of data $\{x_i\}_{i=1}^n$ on the sphere.

- *Karcher Mean (or Fréchet Mean)*
- *Approximation using the tangent space*

3.4.1 Karcher Mean (Fréchet Mean)

The Karcher mean m is the centroid of these data points on the manifold M, which defines as

$$m = \underset{p \in M}{argmin} \sum_{i=1}^{N} d^2(p, x_i)$$

Where $d(.,.)$ is the metric defined on M (e.g., Euclidean space uses Euclidean distance; Riemannian Manifold uses geodesic distance). The general approach here depends on the shape of the manifold M and the metric.

[20] [32] Planas, Oune and Bostanabad (2020): Extrapolation With Gaussian Random Processes and Evolutionary Programming

This relies on a numerical optimization algorithm (e.g., Steepest gradient descent). However, the gradient may be hard to derive explicitly, sometimes we may need to use an existing package to calculate, e.g., using the package geomstats[21]

3.4.2 Approximation using the tangent space

Although Karcher mean is a more popular way of defining the mean data on the manifold. However, here we provide another way that can give an approximation to this centroid based on the tangent space, exponential map, and inverse exponential map.

1. *pick a point* $x \in M$*, and set the tangent space at a point* x *to* M *be* T_xM
2. *using inverse exponential map to map all points* x_i *to* T_xM *using* $v_i = exp_x^{-1}(x_i)$
3. *we can then calculate the Euclidean mean for each coordinate* $\bar{v} = (\frac{1}{n}\sum_{i=1}^{n} v_i^{(1)}, ..., \frac{1}{n}\sum_{i=1}^{n} v_i^{(d)})$
4. *using the exponential map to project back* $\bar{v}$ *will be the centroid of these data points* $m = exp_x(\bar{v})$

Note that although an exponential map only defines locally around the point of contact. However, because the sphere has its convenience structure, the point x *can be any point on* M *unless the point for the exponential map is at the other end (e.g., North Pole vs. South Pole). So unless some data point* x_i *is exactly at the South Pole, or otherwise, the above method can be conveniently used choose the North Pole (*$x = (0, 0, 1)$*).*

3.4.3 Comparison between them

Although Karcher Mean gives the most popular mean definition, the tangent space method can provide a simple and intuitive way to approximate (See Figure 16, Table 4 and 2 for the comparisons) it as well. Also, we can re-use all of our written Python code without relying on another package. So, for the subsequent section for centroid (mean) calculation, we will use the tangent space method.

[21] Geomstats for Frechet mean: https://geomstats.github.io/notebooks/06_practical_methods__riemannian_frechet_mean_and_tangent_pca.html

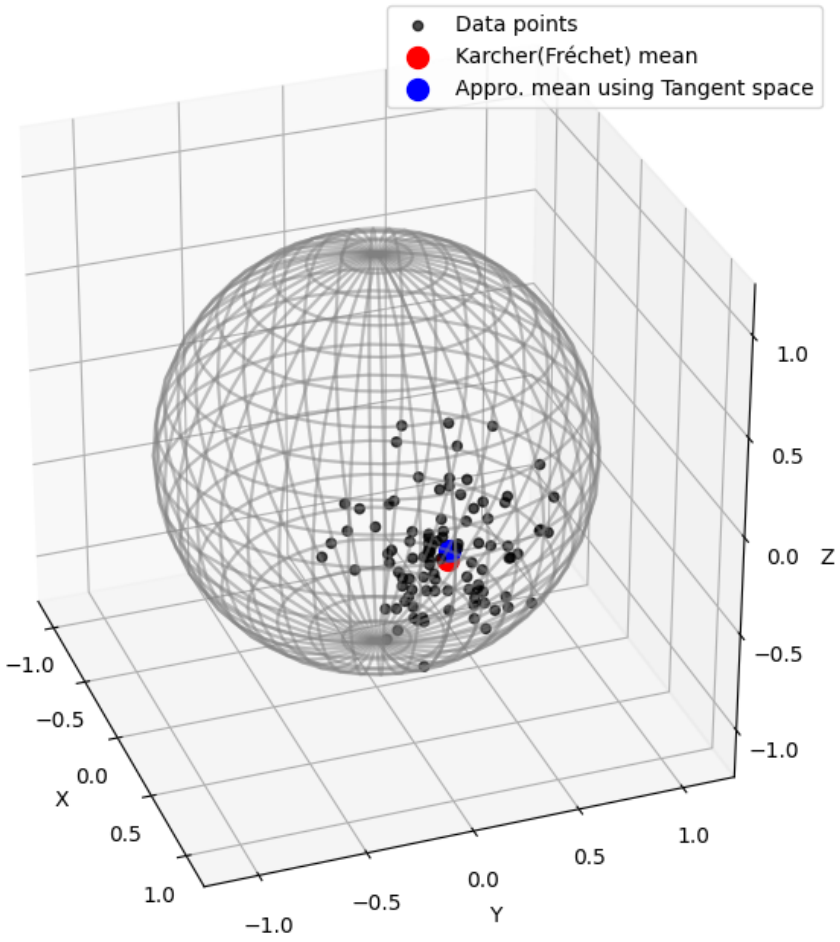


Figure 16: 3D scatter plot, with mean using both methods

Method	x	y	z
Karcher Mean	0.97997464	-0.09176615	0.17671636
Approximation using the Tangent Space	0.97147476	-0.08196176	0.22252878

Table 1: Comparison of centroid calculation method based on 10 spherical data

Method	x	y	z
Karcher Mean	0.99997314	-0.00713869	0.00166018
Approximate using Tangent Space	0.99843699	-0.00728171	0.05541257

Table 2: Comparison of centroid calculation method based on 1000 spherical data

However, for more general usage, especially for manifolds other than sphere usage, using Karcher Mean would be a better choice because the use of approximation using tangent space may lead to incorrect results, especially when the data may be scattered a lot.

3.5 Mean squared error

3.5.1 Known true turve

Similar to many regression problems, sometimes we may check the mean squared error of the curve fit versus the true value.

Suppose in ordinary linear regression, we have the relationship $y_i = \beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p} + \epsilon_i$, $\forall i = 1, 2, ..., n$, *with some suitable assumptions. After we fit the data, then we have a fitted value of the response* $\hat{y}_i$, $\forall i$ *in* $1, 2, ..., n$. *Then we can define the Mean Squared Error to assess the average error of the estimate apart from the true value by*

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Here, we used the squared Euclidean distance to assess the error $(y_i - \hat{y}_i)^2$.

However, if our response is a time series of trajectory, each position on M, *then we cannot simply use Euclidean distance to calculate the error. This is because, for some specific manifolds, like a surface of Swiss roll, the position between 2 endpoints may be very close if we view the manifold in Euclidean space. However, to travel from 1 end to another end, the total distance travel is needed to across the whole Swiss roll. So, in general, if the points indeed reside on some specific surface, we need to use another metric to determine the distance between 2 points, like geodesic distance.*

$$MSE_g = \frac{1}{n}\sum_{i=1}^{n}(d(y_i, \hat{y}_i))^2$$

where $d(a, b)$ *is the geodesic distance between* a *and* $b \in M$.

For the sphere case, we defined the shortest path connecting a *and* b *would be the great circle. So, the geodesic distance between these 2 points is indeed the arc length connecting* a *and* b. *For the distance between 2 time series trajectories on the sphere, commonly, we can define either the largest squared geodesic distance among all points, or the total squared point-wise geodesic distance.*

$$MSE_g = \frac{1}{n}\sum_{j=1}^{N} \underset{1\le i\le n}{max}(d(c_j(t_i), \gamma(t_i))^2 \tag{3.12}$$

or

$$MSE_g = \frac{1}{n}\sum_{j=1}^{N}\sum_{i=0}^{n}(d(c_j(t_i), \gamma(t_i))^2 \tag{3.13}$$

In the section for curve fitting, we will use Equation 3.13 to calculate the mean squared error between the fitted trajectory and the true curve at observed time point $\{t_i\}_{i=0}^{n}$.

3.5.2 Unknown true curve

In reality, the true curve should always be a (fixed) unknown quantity. So, instead of fitting every sample for training, we should reserve a portion of samples for testing and use the testing samples to evaluate the MSE_g.

Let $\{c_j\}_{j=1}^{N}$ *be all sample trajectories. Let the first* m *samples for training and the last* $N-m$ *samples for testing* [22]*. Because of using the first* m *samples for training, we will obtain a model to estimate the sample mean curve (either using centroid-connected curve in Section 3.4; or curve fitting method to get the fitted mean curve in Section 5.1), say the sample mean curve* $\bar{c}$*, then the* MSE_g *formula becomes*

$$MSE_g = \frac{1}{N-m}\sum_{j=N-m+1}^{N}\sum_{i=0}^{n}(d(c_j(t_i), \bar{c}(t_i))^2 \tag{3.14}$$

In later sections, we will assume that the true curve is known (so use Equation 3.13. And for later sections, we will simply ignore the subscript g*, and use the name* MSE *for short.*

[22] Assuming all trajectories are random samples. If not, the split should depend on random sampling

4 Problem formulation

This section contains

- *The definitions, notations, assumptions, and data preparations for Section 5.*
- *Outline the methods will be used throughout Section 5, using the Section 3.*

4.1 Outlines and assumptions

In this section, we will focus on the data on the sphere S^2 instead of a more general Manifold M.

Suppose that there are N sample time series trajectories on the sphere S^2. Let c_j be the sample trajectory j for $j = 1, 2, ..., N$ on the sphere S^2, which is observed the position at discrete time t_i for $i = 0, 1, 2, ..., n$, where $t_0 = 0 < t_1 < t_2 < ... < t_n = T$, with $t_i - t_{i-1} = \Delta t$ for $1 \leq i \leq n$

With the assumptions:

1. *Each trajectory is starting at the North pole, i.e. $c_j(t_0) = c_j(0) = (0, 0, 1)\ \forall j$*
2. *Each sample trajectory is independent.*

See Figure 17 for 2 sample trajectories, each starting at the North pole, and each point is observed at time $t = 0, 1, 2, ..., 30$, and connected successive points using piece-wise geodesic. (The curve definition and sample generation method will be discussed in 4.2.1).

Generally, Section 5 heavily relies on the use of unrolling, unwrapping, and wrapping method to unroll a sample trajectory curve from the sphere S^2 to the tangent plane with a contact point at the starting point of the curve, which is a horizontal Euclidean plane. For short, name this tangent plane to the sphere S^2 at the starting point be TS^2 (omit the contact point).

4.2 Data preparation

In this subsection, we will introduce the true curve formula for our simulation and statistical analysis throughout Section 5.

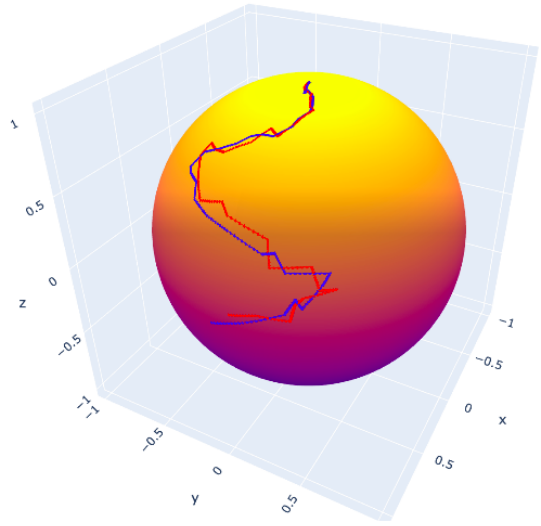


Figure 17: 2 sample trajectories (connected with piece-wise geodesic)

4.2.1 True curve

Note that the data is a time series. Our approach is to first define the data in $(u, v) \in \mathbb{R}^2$ *parameterization first, and then convert it into* $(x, y, z) \in S^2$.

The actual construction using (u, v) *parameterization is as follows (displayed in Figure 18). The* (u, v) *relationship is a pure si1nusoidal curve using the equation is* $v = \frac{\pi}{6} sin(6u)$. *The actual trajectory on the sphere is displayed in Figure 19.*

Time t	u	v
0	0	0
5	$\pi/12$	$\pi/6$
10	$2\pi/12$	0
15	$3\pi/12$	2π - $\pi/6$
20	$4\pi/12$	0
25	$5\pi/12$	$\pi/6$
30	$6\pi/12$	0

Table 3: Time-series of the true curve trajectory on u-v plane

4.2.2 Sample time series generation with random errors

Recall that this dissertation would like to make an inference if there are N sample trajectories collected, so we need to have a way to generate samples from the given true time series trajectory according to Table 3.

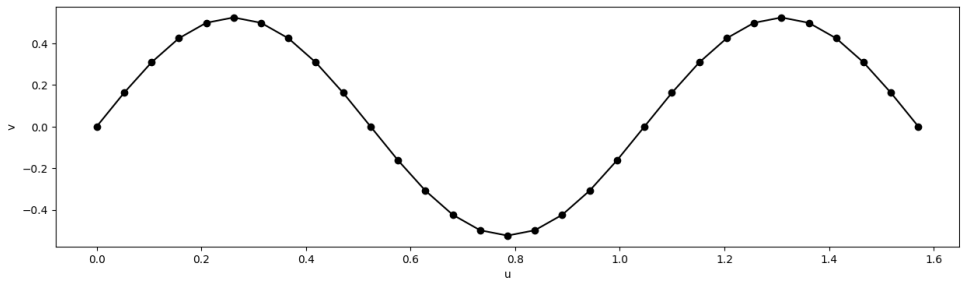


Figure 18: True curve on u-v plane

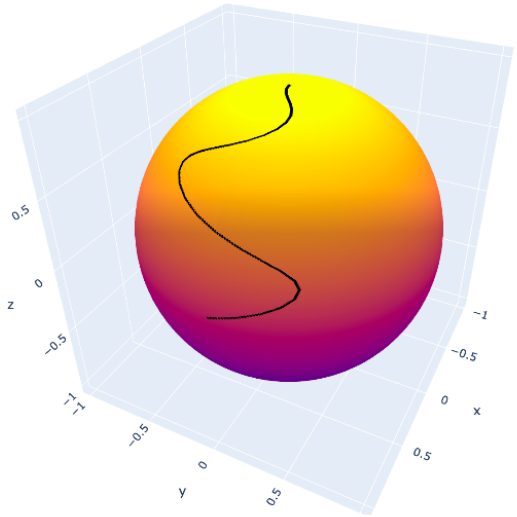


Figure 19: True curve on the sphere

There are different methods to generate points on the sphere with mean position p and with variability. For example, the use of von Mises–Fisher distribution[23]*.*

For our simple case, we can simply make use of (u, v) parameterization, and add variation onto the (u, v) for each observed time t (except when $t = 0$), say, $(u_{new}, v_{new}) = (u + Z_1, v + Z_2)$, where Z_1, Z_2 are two independent and identically distributed normal variables with mean 0 and pre-determined standard deviation σ, $Z_i \sim N(0, \sigma^2)$

We should note that this approach will not have the same effect as using von Mises-Fisher distribution, but it is adequate for our use case.

The following are 2 sample time series trajectories generated from the true curve, with common-sd $= 0.05$.

[23]von Mises–Fisher distribution is a commonly used distribution in Directional Statistics. For sphere, this can generate points around a mean direction μ with concentration, see [**?**]

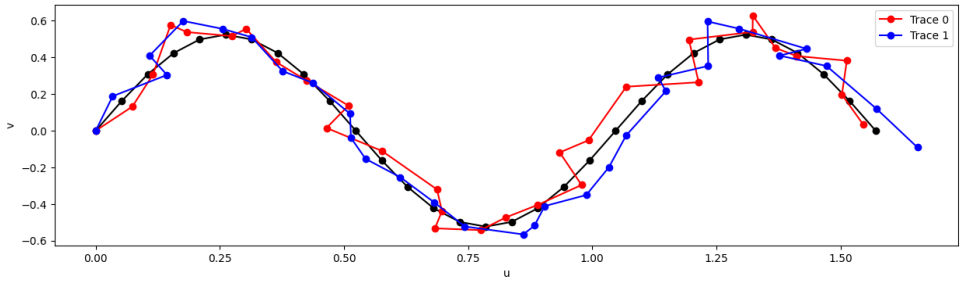


Figure 20: 2 sample trajectories parameterized on u-v plane (with true curve parameterization)

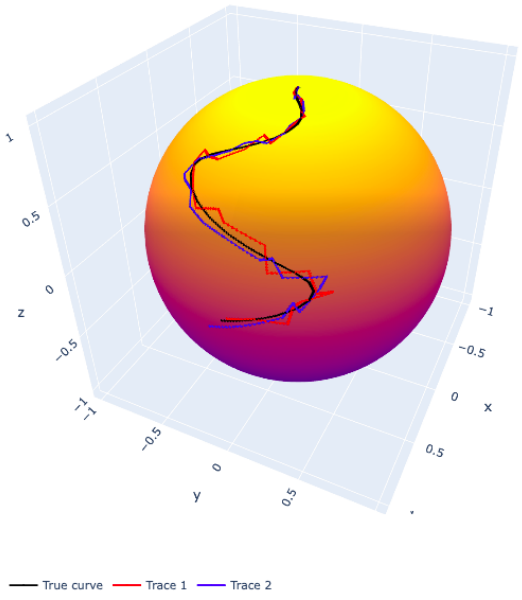


Figure 21: 2 sample trajectories on the sphere (with true curve)

5 Statistical analysis

Following the previous sections, we can now perform statistical analysis using the same notations. Assume:

1. *The true curve γ is unknown fixed, but we collected a sample of size 10 of time series trajectories $\{c_j\}_{j=1}^{10}$, each observed at time $\{t_i\}_{i=0}^{30}$, and is starting at the North pole $(0, 0, 1)$.*
2. *Sample trajectories are mutually independent, starting at the North pole, and assumed coming from the same true curve.*
3. *The sample curves generation4.2.2 under the same common-sd $= 0.05$ for all time points and for all samples.*

In subsection 5.1, we will carry out curve fitting strategy using Cubic Smoothing Spline and Gaussian Processes Regression (GPR) For both fitting methods, we will perform curve fitting for the case of 1 sample trajectory and 10 sample trajectories, respectively. The reason is that, first, it may be of interest that the effect of sample size may affect the curve fitting accuracy. Second, for previous studies like Jupp and Kent (1987) [20] and Kim et. al. (2021) [23], their data fitting example is focused on 1 sample time series. We would like to extend this to multiple samples. A comparison of both fitting methods will be discussed.

To discuss the accuracy, we will use the MSE mentioned in the section 3.13 and compare both MSE based on the fitted function to the true curve.

In section 5.2, we will focus on the use of GPR to perform prediction (up to 5 time steps ahead) on the tangent plane TS^2, and wrap back to the sphere. Also, we can make use of GP capability to simulate points on the tangent place (and also wrap back to the sphere) to see the possible future position of the mean trajectory.

5.1 Curve fitting and modeling

For the curve fitting method, the procedure is:

1. *Find the initial base path, first calculate the centroid using 3.4 of all samples at each time*

point, then connect all such centroid using piece-wise geodesic (for 1 sample trajectory case, this centroid-connected curve equivalent to the trajectory).

2. *Unroll the base path.*
3. *Unwrap all points of each sample curves c_j onto the tangent plane to obtain vectors. Note that the dimension of the tangent plane is 2, so each vector $v(t_i)$ can be written as $(x(t_i), y(t_i))$ explicitly.*
4. *Consider the x-coordinate and y-coordinate of the tangent vector. We can perform fitting on both x against t, and y against t, independently. After fitting, we will perform an adjustment on the fitted $\hat{x}(t)$ and $\hat{y}(t)$ to make $\hat{x}(0) = \hat{y}(0) = 0$.* [24]
5. *Reconstruct each component in step 4 into a single vector for each time point. Then, perform estimation on inter-time (take the number of inter-time points G = 1). For example, for $t_0 < t_1 < ... < t_n$, we construct 1 inter-time point for successive times, e.g., our current observed times are 0, 1, ..., 30; then we construct to another times, 0, 0.5, 1, 1.5, ..., 29, 29.5, 30.*
6. *Wrap back all vectors as well as the inter-time vectors to the sphere, it will be our next new base path for unrolling.*
7. *Iteratively, perform step 2-6 until the geodesic distance between the previous path and the new path are less than a predefined tolerance (say, $\epsilon < 0.001$).*

5.1.1 Cubic Smoothing Spline

Using the Cubic Smoothing Spline technique 3.2 with the iteration procedure 5.1. Recall that Cubic Smoothing Spline has its own smoothness parameter that we need to balance between points interpolation and smoothness of the function. There are different methods to address the best smoothness parameter (See Kim et. al. (2021) [23], section 3.1, p.118). They used the leave-one-out cross-validation method to find the optimal smoothness parameter that minimizes the mean squared error of the observed tangent vector and the fitted unrolling

[24] This is because we strictly required the initial tangent vector to be a zero-vector to the sphere at the point of contact with the North Pole. The method used is to perform translation of $\hat{x}_{fit}(x)$ vertically by the amount $\hat{x}_{fit}(0)$, to make $\hat{x}_{adj}(t) = \hat{x}_{fit}(t) - \hat{x}_{fit}(0)$ for all t. A similar operation will also be performed for $\hat{y}_{fit}(t)$. We should note that $\hat{x}_{fit}(0)$ would not be exactly zero like many ordinary regression problems. However, we should note that the differences are very small. So, performing a simple translation would not have a large impact.

(if i^{th} observation excluded).

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n} ||v_i - \widehat{f^{\dagger}_{\lambda,-i}}(t_i)||^2 \tag{5.1}$$

In practice, it is recommended to use the above method to get the best smoothness parameter. From our study, we will simply use a fixed smooth parameter to perform our fitting. We will use the smoothness $= 0.85$.

A. Fitting based on 1 sample

Based on the configuration:

- *1 sample trajectory*
- *smoothness = 0.85*
- *inter-time interpolation G = 1*
- *convergence criterion* $\epsilon = 10^{-3}$

The Figures 22, 23 and 24 are the fittings on the tangent plane TS^2.

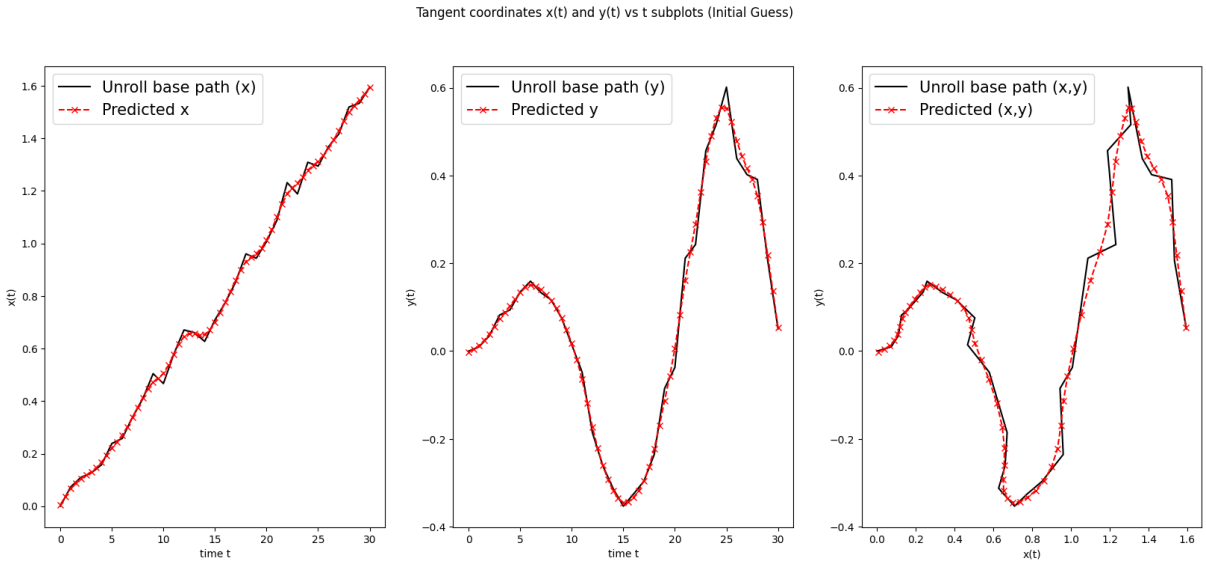


Figure 22: Cubic Smoothing Spline, based on 1 sample, initial path

The mean squared error for this fitting is 0.001011 (up to 4 significant figures).

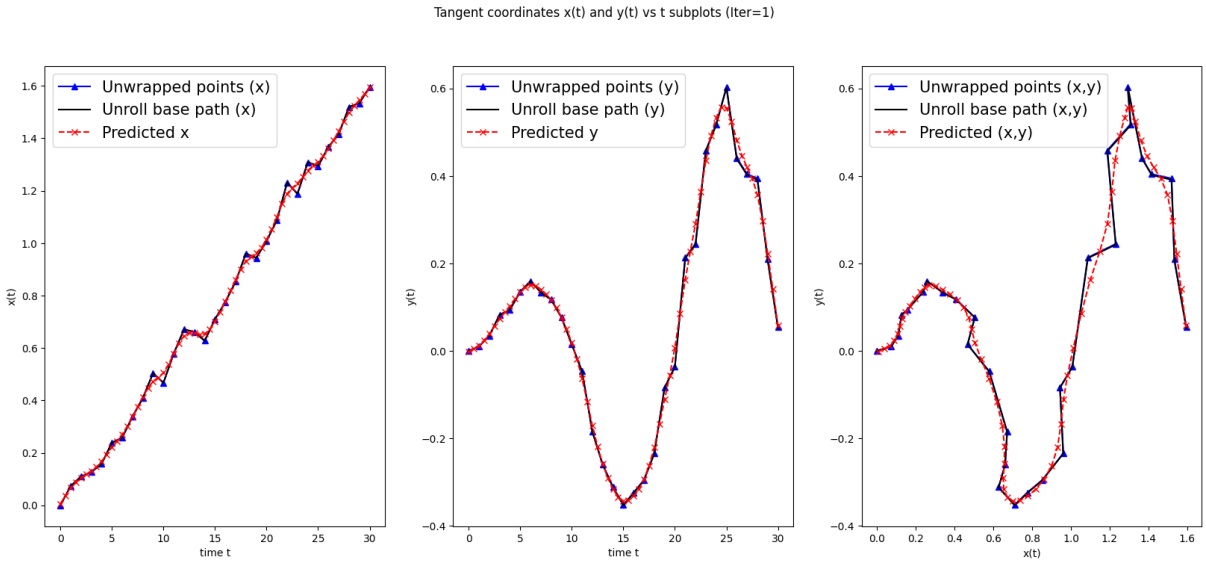


Figure 23: Cubic Smoothing Spline, based on 1 sample, final path

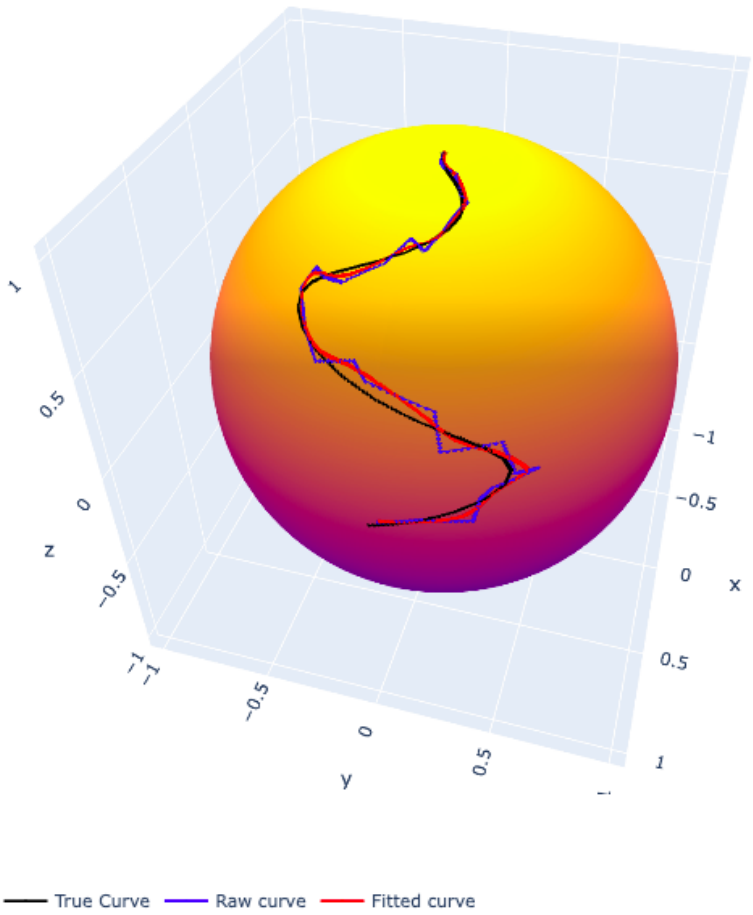


Figure 24: Cubic Smoothing Spline, based on 1 sample, fitted on the sphere

B. Fitting based on 10 samples

Based on the configuration:

- *10 sample trajectories*
- *smoothness = 0.85*
- *inter-time interpolation G = 1*
- *convergence criterion* $\epsilon = 10^{-3}$

Recall that for there are N samples, we need to have an initial base path for unrolling. This base path will be determined using method 3.4.2. For further iterations, we will focus on the fitting of this centroid-connected curve (unlike GPR for N samples).

The following are the fittings on the tangent plane TS^2.

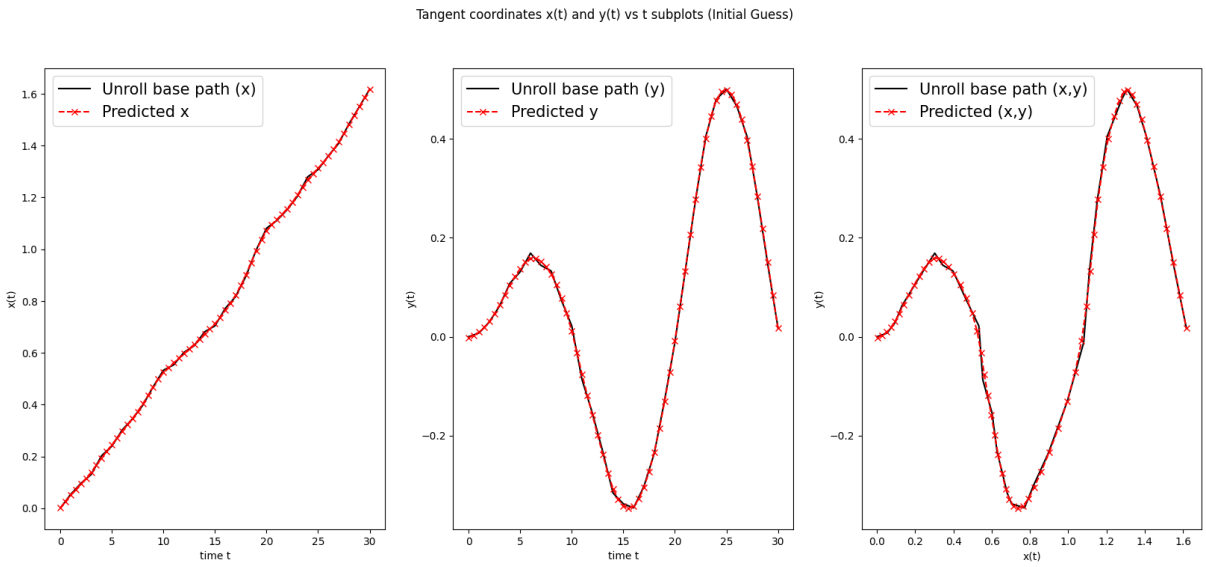


Figure 25: Cubic Smoothing Spline, based on 10 samples, initial path=centroid-connected-path

The mean squared error for this fitting is 0.0002094 (up to 4 sig. fig.).

5.1.2 Gaussian Processes Regression (GPR)

Using the GPR (Section 3.3) with the iteration procedure 5.1. Recall that using GPR, we need to configure the covariance matrix (in GPR literature, this is also known as Kernel).

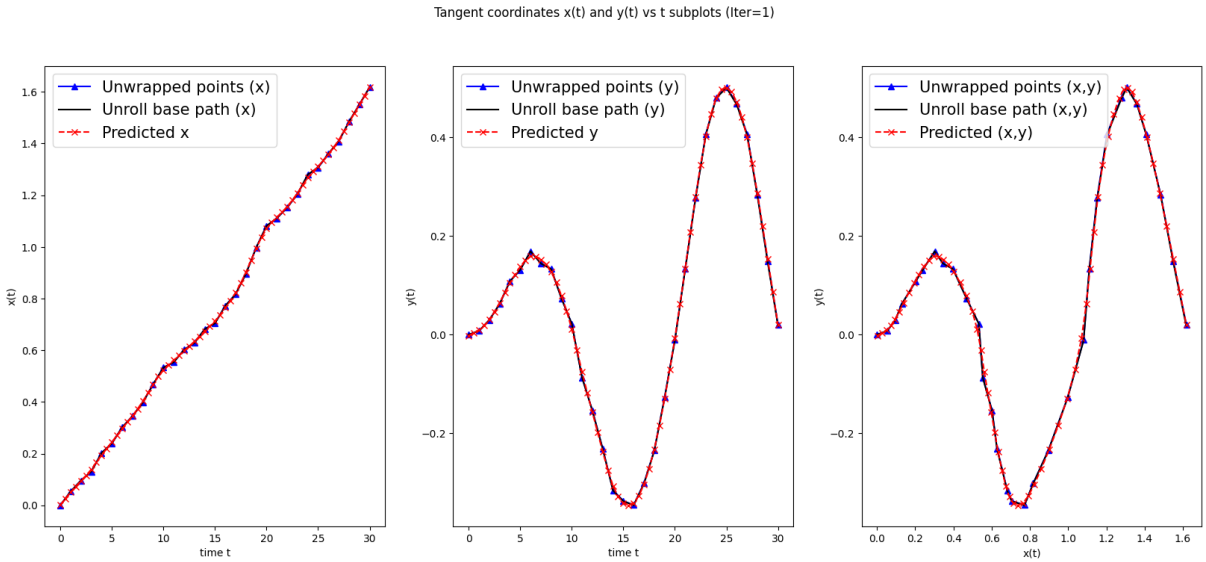


Figure 26: Cubic Smoothing Spline, based on 10 samples, final path

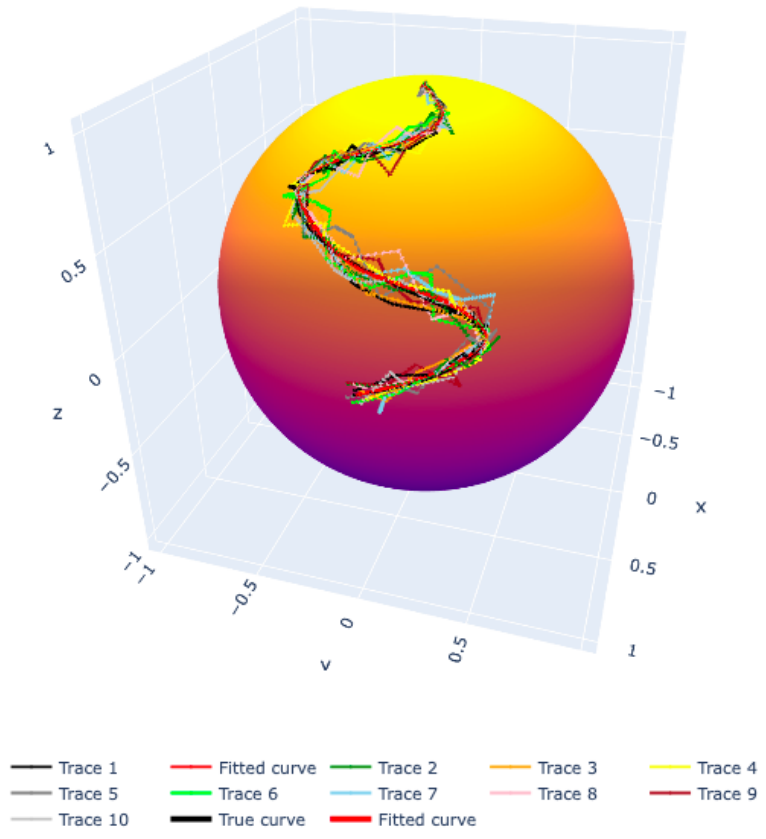


Figure 27: Cubic Smoothing Spline, based on 10 samples, fitted on the sphere

Common kernels are the Radial basis function (RBF) kernel and Matérn kernel, see Scikit-

learn documentation [25].

A. Fitting based on 1 sample

Based on the configuration:

- *1 sample trajectory*
- *using RBF + White Noise kernel*
- *inter-time interpolation $G = 1$*
- *convergence criterion $\epsilon = 10^{-3}$*

The followings are the fitting on the tangent plane TS^2

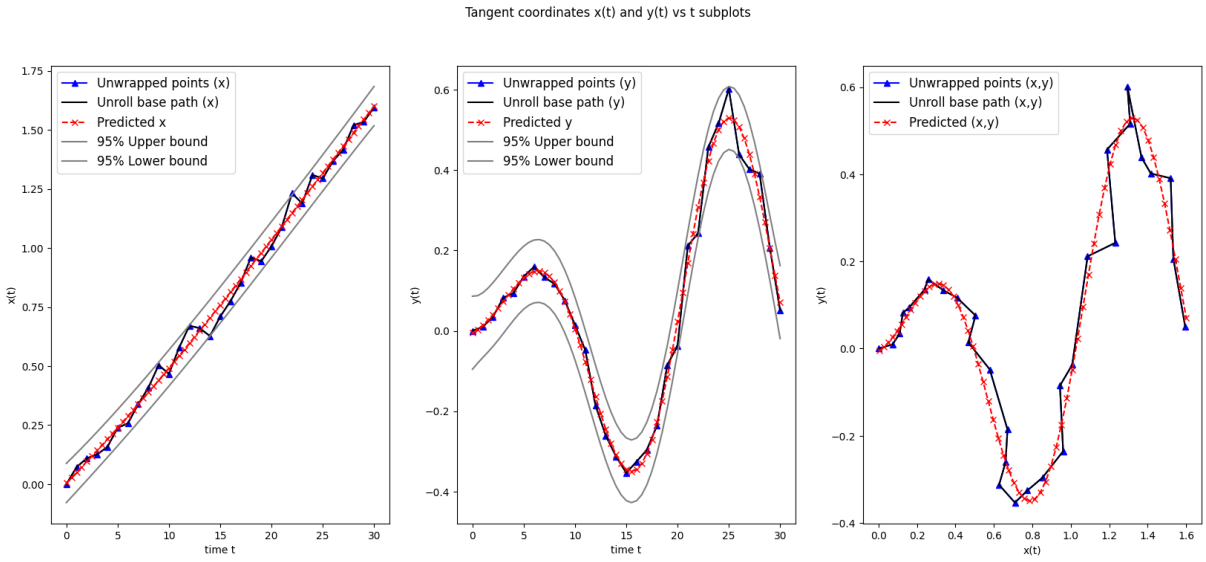


Figure 28: GPR, based on 1 sample, initial path

The mean squared error for this fitting is 0.0009274 (up to 4 sig. fig.).

B. Fitting based on 10 samples

The general approach for 1 sample and 10 samples are the same. Those of which required the use of fitting GPR on the unwrapped points on the tangent plane.

Based on the configuration:

- *10 sample trajectories*

[25] See Scikit-learn documentation: https://scikit-learn.org/stable/modules/gaussian_process.html

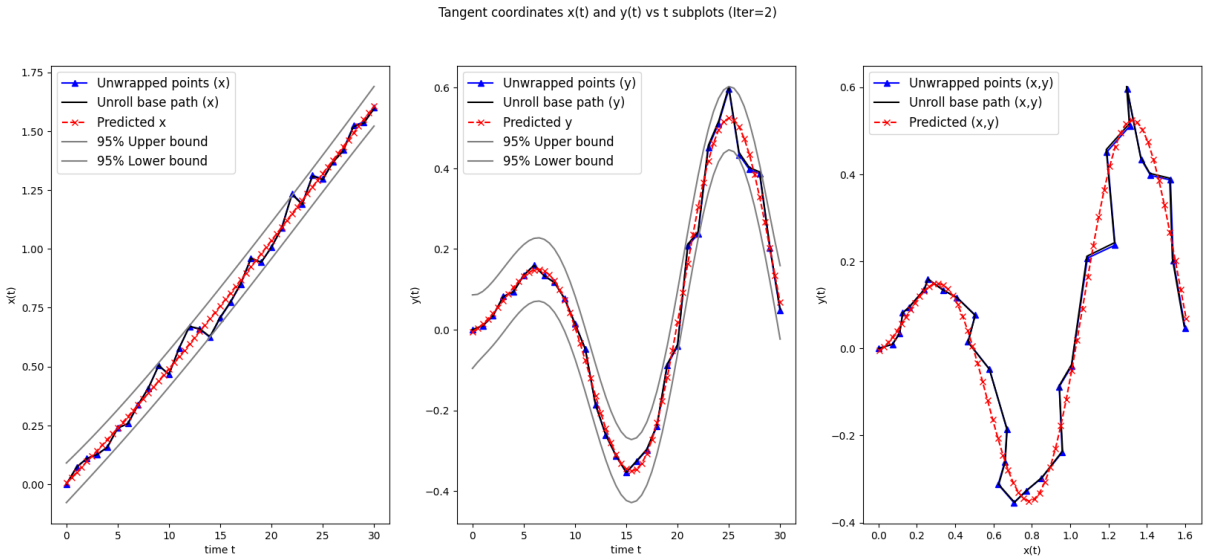


Figure 29: GPR, based on 1 sample, final path

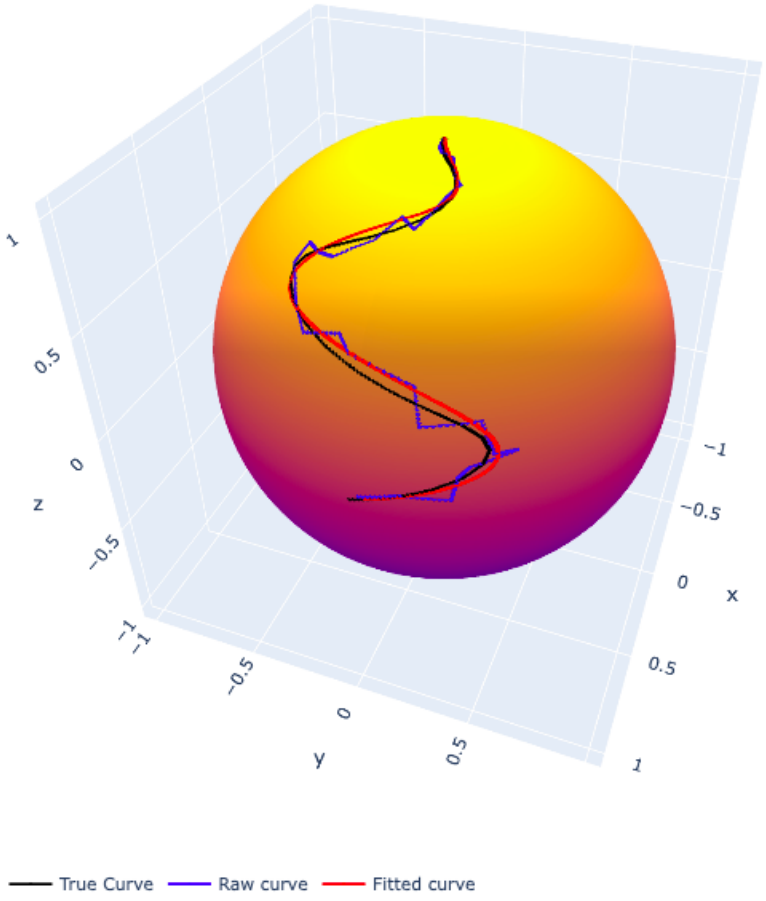


Figure 30: GPR, based on 1 sample, fitted on the sphere

- *using RBF + White Noise kernel*
- *inter-time interpolation $G = 1$*
- *convergence criterion $\epsilon = 10^{-3}$*

The followings are the fitting on the tangent plane TS^2.

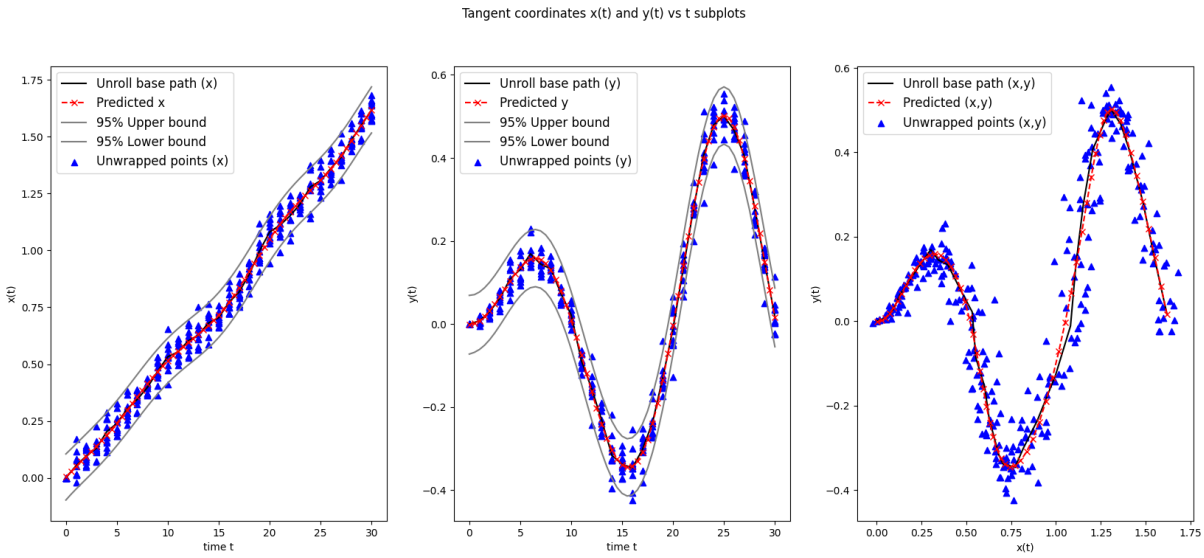


Figure 31: GPR, based on 10 samples, initial path

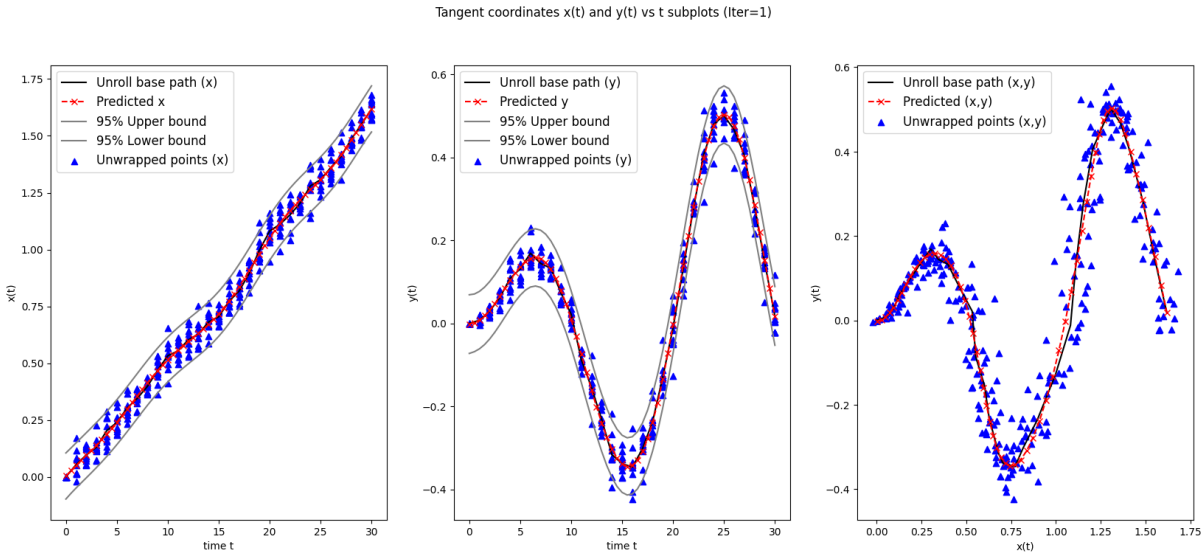


Figure 32: GPR, based on 10 samples, final path

The mean squared error for this fitting is 0.0002127 (up to 4 sig. fig.).

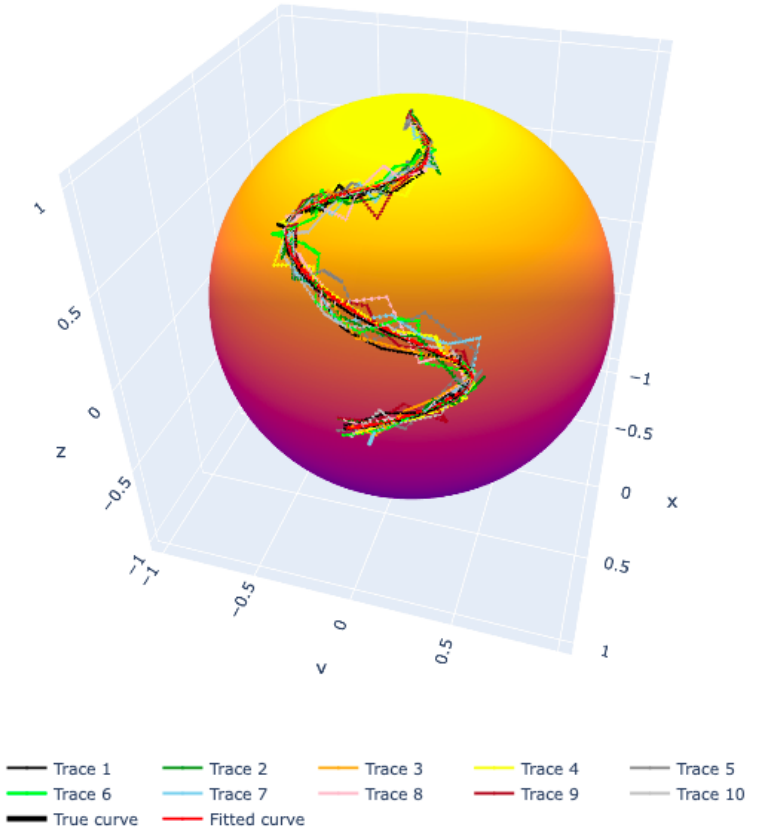


Figure 33: GPR, based on 10 samples, fitted on the sphere

5.1.3 Comparison

Based on the above findings, we can see that both methods have similar mean squared errors. Also, they show that the mean squared error decreased as the number of samples increased. The following will conduct the same simulation study again, with different sample sizes.

From the plot, we have the following findings:

1. *We can see that the use of GPR has a smaller MSE than Cubic Smoothing Spline in general.*
2. *The MSE is decreasing as the sample size increases. However, as the sample size is at least 10, the MSE becomes stable, and the drop is not significant. So, for computational efficiency, using sample size $= 10$ is a good choice, as getting more and more samples does not improve the MSE a lot.*
3. *Fitting GPR is slower than fitting Cubic Smoothing Spline for any sample size (because the GPR fitting algorithm requires the unwrapping of every point for every sample*

Sample Size	Cubic Smoothing Spline	Gaussian Processes Regression
1	0.002204	0.001465
3	0.000711	0.000721
5	0.000382	0.000376
7	0.000175	0.000108
10	0.000152	0.000135
15	0.000133	0.000109
20	0.000097	0.000087
25	0.000086	0.000088
30	0.000053	0.000045
35	0.000053	0.000036
40	0.000040	0.000026
45	0.000043	0.000048
50	0.000041	0.000023

Table 4: Comparison of both methods, under different sample size

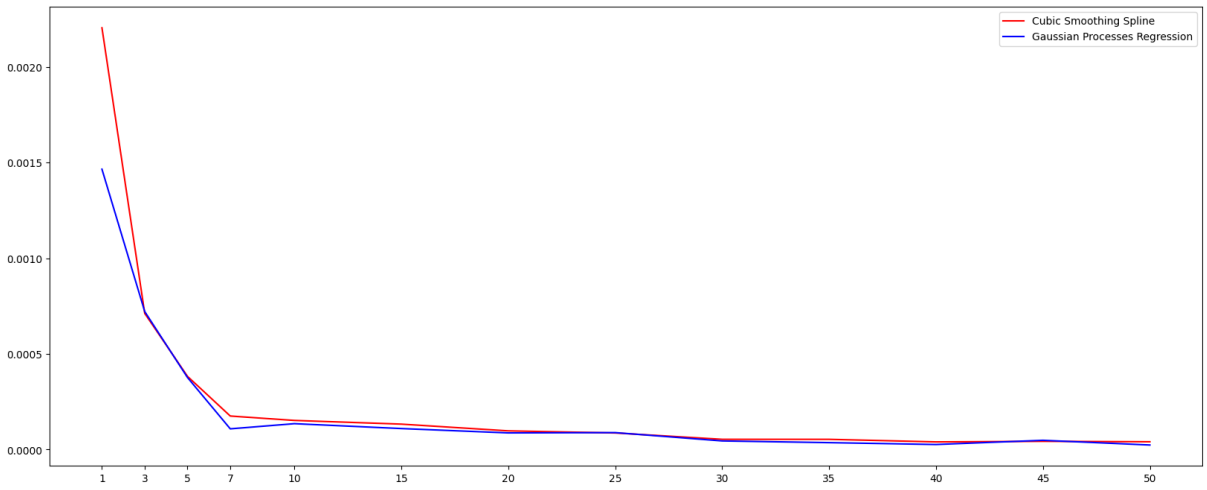


Figure 34: Plot the comparison of both methods, under different sample size

trajectory to the tangent space.

In general, we know that fitting GPR requires more computational power, and the accuracy (in terms of MSE) is not significantly better than fitting the Cubic Smoothing Spline. However, in terms of the probabilistic structure of the GP model and the recent advance in computational power (e.g., the use of GPU). For a reasonable sample size (say $N \leq 20$), for curve fitting and forecasting, using GPR seemed to be better over the Cubic Smoothing Spline.

5.2 Future point predictions on the sphere

In this section, we will use the GPR to perform predictions on the tangent plane TS^2 and wrap it back to the sphere.

- *Predict mean k-steps ahead position on the sphere, for $k = 1, ..., 5$*
- *possible k-steps ahead position on the sphere.*

5.2.1 Simulation using the trained GPR

Unlike the Cubic Smoothing Spline, which is mainly used for curve fitting, the Gaussian Processes model is a probabilistic model that is also capable of simulating samples from the trained model, as well as making future predictions.

In this section, following the previous findings 5.1.3, we will use 10 sample trajectories to train a GPR model and perform predictions.

A. On the tangent plane

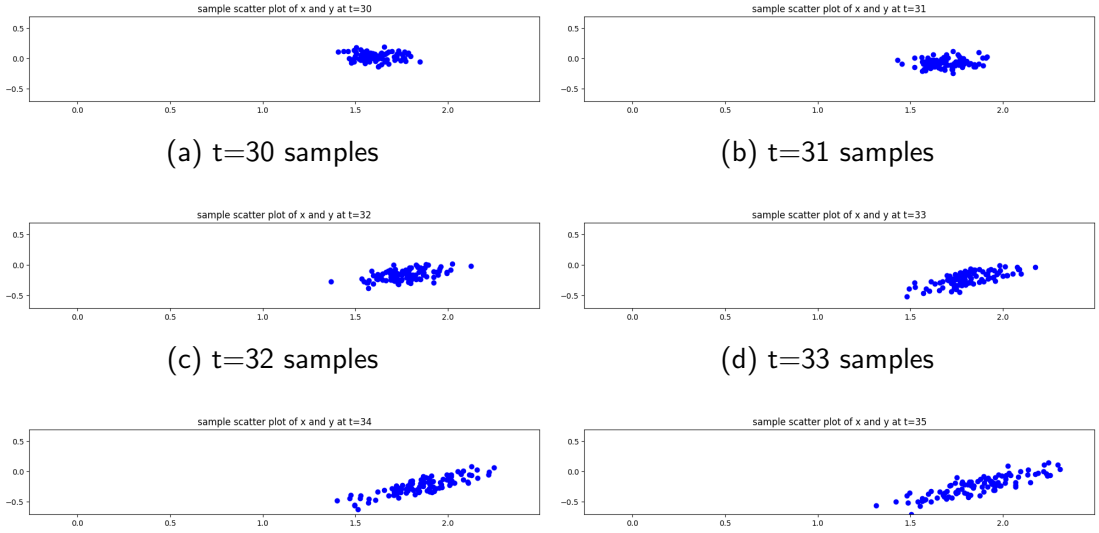
Using this trained model, we can find the predicted values of both x and y coordinates of the tangent vectors. Denote the k-steps ahead prediction of x and y, given observed up to time t_n be $\hat{x}(t_n + k\Delta t)$ and $\hat{y}(t_n + k\Delta t)$, where now our $t_n = 30, \Delta t = 1$, and $k = 1, 2, ..., 5$.

k	$\hat{x}(30+k)$	$\hat{y}(30+k)$
1	1.693772	-0.084179
2	1.765013	-0.169616
3	1.823753	-0.222683
4	1.862338	-0.244491
5	1.874229	-0.240351

Table 5: k-steps ahead predictions

The followings are the scatter plots of $(\hat{x}(30 + k), \hat{y}(30 + k)$ for those 100 samples, for $k = 1, ..., 5$ (for reference, also put the scatter plot at t=30).

From Figure 36, we can see that, for time $t \leq 30$, all samples (gray curves) are very close to the fitted mean (blue curves). For $t \geq 31$, we can see that the predicted values spread more (red curves).



(a) t=30 samples (b) t=31 samples

(c) t=32 samples (d) t=33 samples

(e) t=34 samples (f) t=35 samples

Figure 35: Scatter plot of samples at t=30, 31, ..., 35

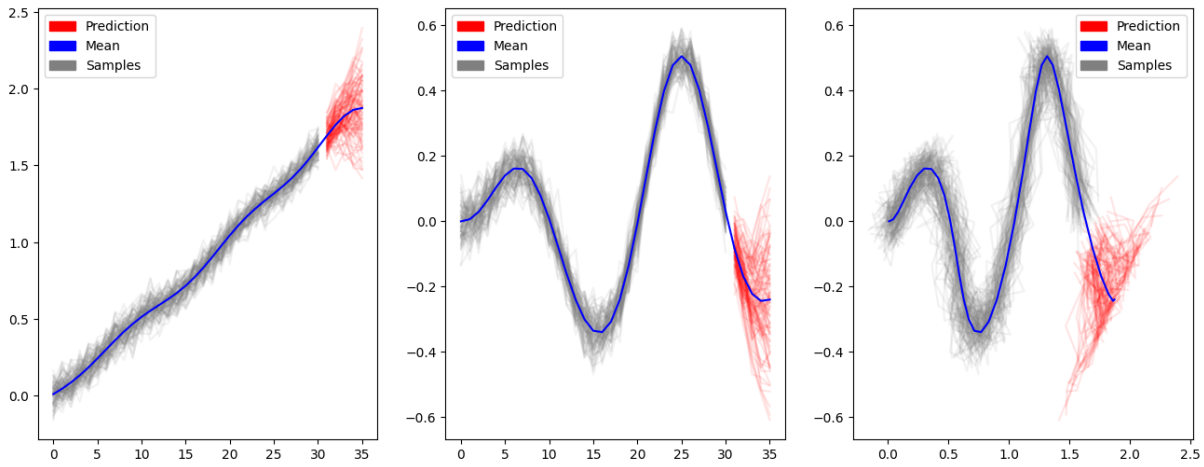


Figure 36: Predictions on the tangent plane

The same evidence can be seen in Figure 35. At $t = 30$ (last observed time), the scatter plot is not scattered a lot. As the step ahead increases, the scatter plot variability increases. This behavior makes sense because the uncertainty of future values will be accumulated, and so the overall variability of future values will increase.

B. Rolling and wrapping back to the sphere

From the trained model, because we have the mean predictions for (x, y)*, and we have our predictions up to* $t = 35$*, assuming the rolling*[26] *and unrolling is bijective, then we perform the same rolling procedure, to make our fitted predictions mean function back to the sphere. Hence, we can wrap back all future predicted values to the sphere. Figure 37 shows the prediction of mean curve on the sphere.*

Figure 39 shows the predictions of on the sphere from $t = 30$ *to* 35 *on the sphere ; and the Figure 40 shows the individual predictions rom* $t = 30$ *to* $t = 35$*.*

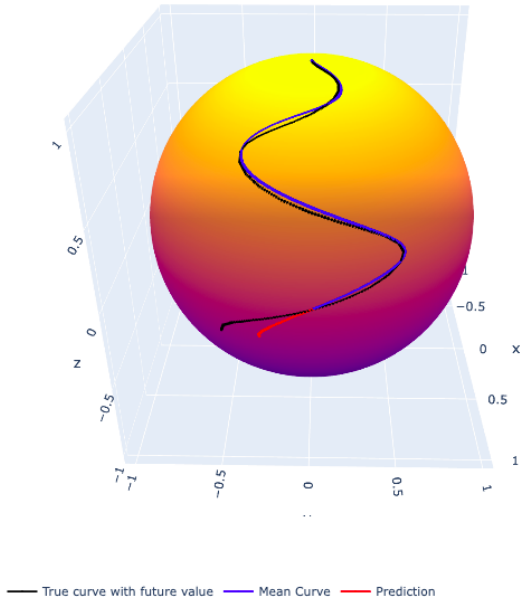


Figure 37: Prediction of mean curve on the sphere

The geodesic distance between the true value and the predicted value at time t *(up to 4 sig. fig.) is*

t	error using geodesic distance
30	0.02405
31	0.06367
32	0.1173
33	0.1717
34	0.2166
35	0.2464

Table 6: k-steps ahead prediction error (using geodesic distance)

[26] We refer the terms *rolling be the reverse process of unrolling.*

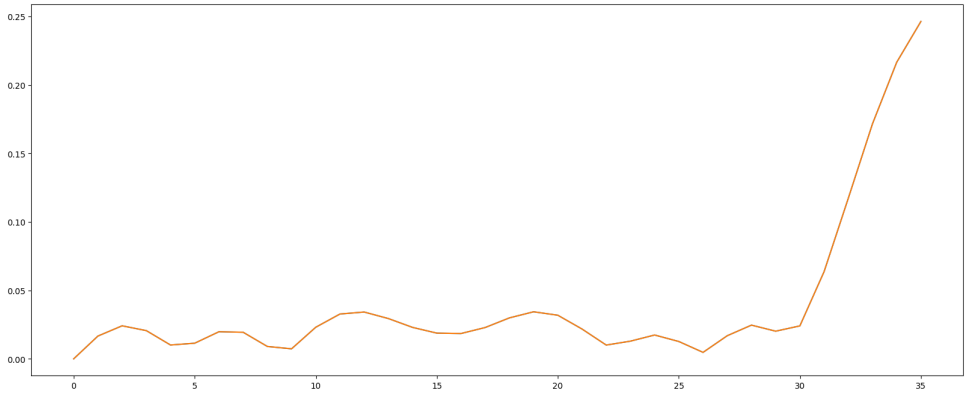


Figure 38: Geodesic distance between true curve and fitted curve at t

We can see that the error increases as t *increases after* $t > 31$*. At* $t = 35$*, the error is about 4 times than at* $t = 31$*.*

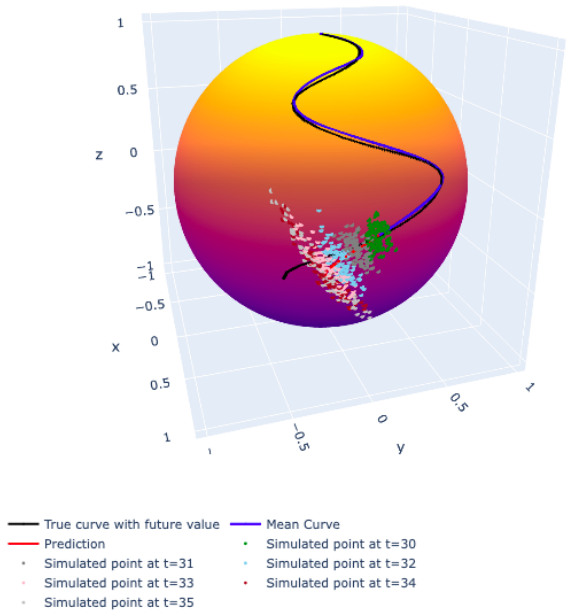
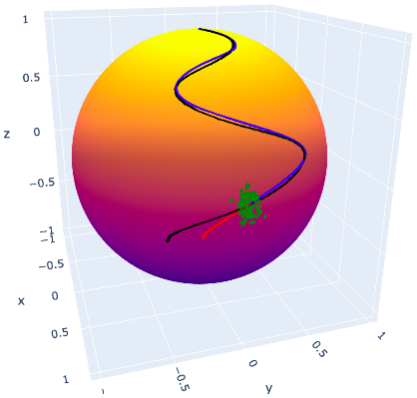
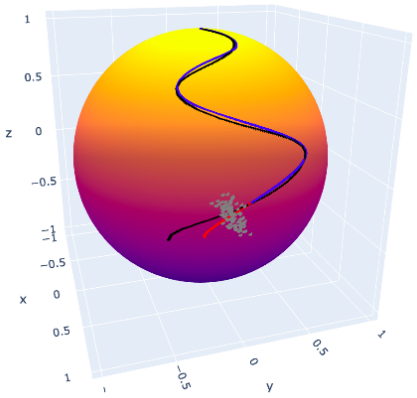


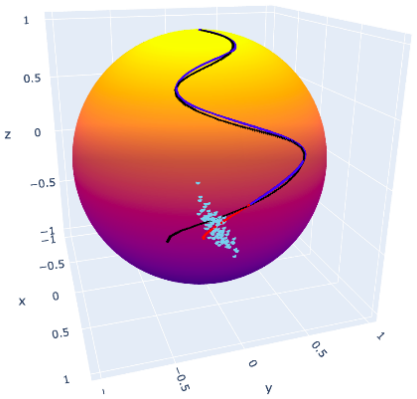
Figure 39: Scatter plot of samples from t=30 to 35 on the sphere



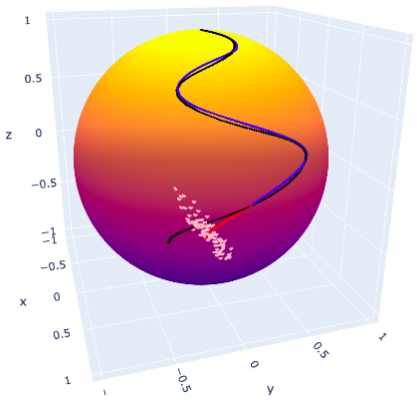
t=30 predictions



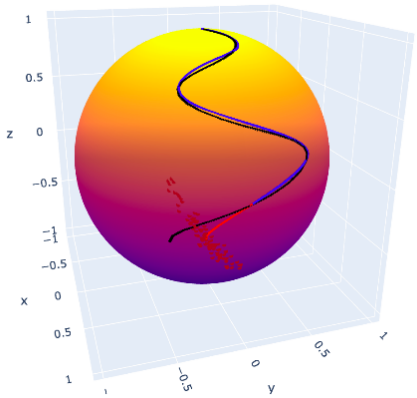
t=31 predictions



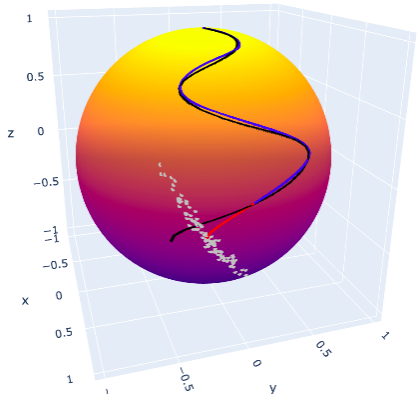
t=32 predictions



t=33 predictions



t=34 predictions



t=35 predictions

Figure 40: Individual scatter plot of samples from t=30 to 35 on the sphere

6 Discussions

6.1 To use curve fitting in common practice

In section 5.1, we have performed curve fitting based on Cubic Smoothing Spline and GPR and perform forecasting based on GPR. However, we have fitted the data using a fixed smoothness parameter for Cubic Smoothing Spline and using the pre-determined kernels for GPR.

In practice, we shall fit in a more general approach.

- *For Cubic Smoothing Spline, we should use the leave-one-out cross-validation method to find the best smoothness parameter that minimizes the MSE.*
- *For GPR, we should try more kernels for fitting, say Matérn kernel and Exp-Sine-Squared kernel* [27]*.*

6.2 Correlated structure of the output using GPR

From our modeling of the output on the tangent place, we perform independent GP on both $x(t)$ and $y(t)$ components. Some may doubt that the modeling should also consider the inter-dependence between $x(t)$ and $y(t)$. For more advanced usage, there is a method called Multi-task GP that can directly model $x(t)$ and $y(t)$ and its correlation [28]*.*

6.3 Choice of kernels

In section 3.3.3, we tried to use the trained GPR to forecast k-steps ahead prediction. And the results are reasonable for $k = 1, 2$. However, the subsequent predictions are not performing well. As we mentioned above, we should try different kernels to assess the prediction ability and compare which one may be closer to the true mean curve. Intuitively, the Exp-Sine-Squared kernel may be a good choice because our true curve definition 4.2.1 exhibits a periodic behavior. But in general, this may require a trial-and-error to see which kernels work best.

[27] Exp-Sine-Squared kernel allows modeling periodic functions, see section 1.7.5.7 *https://scikit-learn.org/stable/modules/gaussian_process.html*

[28] See https://docs.gpytorch.ai/en/latest/examples/03_Multitask_Exact_GPs/Multitask_GP_Regression.html

6.4 Unknown true curve

We should also note that, for curve fitting strategy and forecasting, we used the True Curve to compare with the model to calculate MSE. However, in reality, we should never know the true curve. So, it is better to use the estimated mean with testing curves (as discussed in 3.5.2)

6.5 Sample size determination

A common practice in machine learning is to perform a 80-20 split, i.e., using 80% of samples for training and the remaining 20% of samples for testing the error. However, this proportion may depend on the underlying project.

In our study, due to the computational time vs. MSE accuracy trade-off we discussed before, we think 50-50 split should also be a good choice as we do not have too many samples to train in reality (as training in a large batch of samples does not improve a lot). So, reserving more samples for testing may be a good choice.

For example, if we would like to use 50-50 split, and have 10 samples for training, and 10 samples for testing, then we need to have 20 samples in total.

6.6 Limitations and assumptions of the rolling and wrapping back

Recall from methodology section 3.3.3, we discussed that the problem when performing extrapolation would be poor for a distant future time. From the scatter plot above, we can see that 1 to 2 steps ahead prediction seemed to make sense. However, for 3 steps and forward, the prediction seemed not too reliable (See the scatter plot in Figure 36 and 40)

In addition, we performed rolling back the mean function of the tangent plane to the sphere and wrapped all the predictions to the sphere. If we needed to apply the same technique to manifolds other than sphere, we need to have some important assumptions:

Assumption I: unrolling and rolling is bijective

This assumption means, for our usual unrolling, if we know the curve γ of the manifold M, then we can unroll under 3.10 and 3.2 relationship. In the reverse operation, we assumed that, given the tangent vectors on the tangent space, we would get a unique curve on the manifold.

In words, it means the curve on the manifold, and the curve on the tangent space is one-to-one.

Assumption II: the predictions on the manifold are well-defined

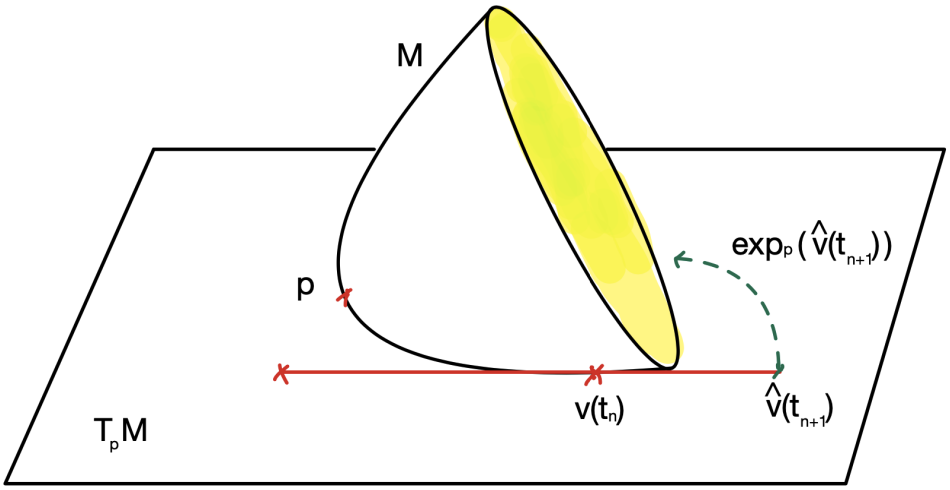


Figure 41: Potential issue in hemisphere

This assumption means although we can predict future points on the tangent space using the GPR, we need to assume that the points in reverse operation are well-defined.

i.e., Suppose at time t, the predicted tangent vector is $\hat{v} \in T_pM, \exists x \in M$ such that $wrapping(\hat{v}) = x$

This assumption is cruel because, for some manifold (e.g., hemisphere with base, see Figure 41), the usual unrolling of the curve on the surface of a hemisphere is allowed. However, if we perform prediction on the tangent space and roll and wrap back the data, the parallel transport is not guaranteed for predicted values (as the curve may be across the region of the diameter, that is not differentiable)

7 Conclusions

In conclusion, for the sphere case, we show that we can apply the Cubic Smoothing Spline and Gaussian Processes Regression method to fit single sample trajectory and multiple sample trajectories on the tangent plane.

We also show that, based on suitable assumptions and limitations listed in Section 6.6, the forecasting of the future positions on the sphere is accurate, up to 2-time steps ahead, but the prediction becomes worse after 2-time steps.

We need to point out that, as listed in the discussions section 6, some assumptions and limitations must be fulfilled before applying the forecasting method. Furthermore, this section also lists some general and practical approaches to conducting our experiments.

A Differential Geometry Review Example

A.1 Parallel transport on the sphere for section 2.4

This section provides an example of parallel transporting a vector on the sphere along a curve that used the definition defined in 2.4.

Consider a special case of the surface of sphere S^2 with radius 1 and centered at the origin. So the equation of the surface of this sphere is $x^2 + y^2 + z^2 = 1$. Let p_0 be a point on S^2, with local parameterization U using (u, v), and a map $f : U \to S^2$ with

$$f(u, v) = (sinucosv, sinusinv, cosu)$$

Let γ be a curve on S^2 and the tangent vector field along γ is w and

$$f_u = (cosucosv, cosusinv, -sinu)$$
$$f_v = (-sinusinv, sinucosv, 0)$$

Let

$$\begin{aligned} E(u,v) =< f_u, f_u > &= |f_u|^2 \\ &= (cosucosv)^2 + (cosusinv)^2 + (-sinu)^2 = cos^2u + sin^2u \\ &= 1 \\ F(u,v) =< f_u, f_v > & \\ &= (-sinusinv)(cosucosv) + (sinusinv)(cosucosv) + (-sinu)(0) \\ &= 0 \\ G(u,v) =< f_v, f_v > &= |f_v|^2 \\ &= (-sinusinv)^2 + (sinucosv)^2 + (0)^2 \\ &= sin^2u \end{aligned}$$

To find the Christoffel Symbols, first find $< f_{uu}, f_u >, < f_{uu}, f_v >, < f_{uv}, f_u >$,

$< f_{uv}, f_v >, < f_{vv}, f_u > and < f_{vv}, f_v >:$

$$f_{uu} = (-sinucosv, -sinusinv, -cosu)$$
$$f_{vv} = (-sinucosv, -sinusinv, 0)$$
$$f_{uv} = (-cosusinv, cosucosv, 0)$$

Then

$$\begin{aligned}
< f_{uu}, f_u > &= (-sinucosv)(cosucosv) + (-sinusinv)(cosusinv) + (-cosu)(-sinu) \\
&= 0 \\
< f_{uu}, f_v > &= (-sinucosv)(-sinusinv) + (-sinusinv)(sinucosv) + (-cosu)(0) \\
&= 0 \\
< f_{uv}, f_u > &= (-cosusinv)(cosucosv) + (cosucosv)(cosusinv) + (0)(-sinu) \\
&= 0 \\
< f_{uv}, f_v > &= (-cosusinv)(-sinusinv) + (cosucosv)(sinucosv) + (0)(0) \\
&= sinu\ cosu \\
< f_{vv}, f_u > &= (-sinucosv)(cosucosv) + (-sinusinv)(cosusinv) + (0)(-sinu) \\
&= -sinu\ cosu \\
< f_{vv}, f_v > &= (-sinucosv)(-sinusinv) + (-sinusinv)(sinucosv) + (0)(0) \\
&= 0
\end{aligned}$$

and so the Christoffel Symbol can be determined

$$< f_{uu}, f_u > = \Gamma^1_{11} < f_u, f_u > + \Gamma^2_{11} < f_v, f_u >= \Gamma^1_{11}(1) + \Gamma^2_{11}(0)$$

$$\Gamma^1_{11} = 0$$

$$< f_{uu}, f_v > = \Gamma^1_{11} < f_u, f_v > + \Gamma^2_{11} < f_v, f_v >= \Gamma^1_{11}(0) + \Gamma^2_{11} sin^2 u$$

$$\Gamma^2_{11} = 0$$

$$< f_{vv}, f_u > = \Gamma^1_{22} < f_u, f_u > + \Gamma^2_{22} < f_v, f_u >= \Gamma^1_{22}(1) + \Gamma^2_{22}(0)$$

$$\Gamma^1_{22} = -sinu\ cosu$$

$$< f_{vv}, f_v > = \Gamma^1_{22} < f_u, f_v > + \Gamma^2_{22} < f_v, f_v >= \Gamma^1_{22}(0) + \Gamma^2_{22}(sin^2 u)$$

$$\Gamma^2_{22} = 0$$

$$< f_{uv}, f_u >=< f_{vu}, f_u > = \Gamma^1_{12} < f_u, f_u > + \Gamma^2_{12} < f_v, f_u >= \Gamma^1_{12}(1) + \Gamma^2_{12}(0)$$

$$\Gamma^1_{12} = 0$$

$$< f_{uv}, f_v >=< f_{vu}, f_v > = \Gamma^1_{12} < f_u, f_v > + \Gamma^2_{12} < f_v, f_v >= \Gamma^1_{12}(0) + \Gamma^2_{12}(sin^2 u)$$

$$\Gamma^2_{12} = 1/tanu$$

Now consider the curve and the vector field along the curve

$$\gamma(t) = f(u(t), v(t))$$

$$w(t) = a(t) f_u + b(t) f_v$$

From the result of the covariant derivative, substitute all the results above (for simplicity, omit the (t) *for each function)*

$$\begin{aligned}
\frac{Dw}{dt} &= (a' + \Gamma^1_{11} a u' + \Gamma^1_{12} a v' + \Gamma^1_{12} b u' + \Gamma^1_{22} b v') f_u \\
&+ (b' + \Gamma^2_{11} a u' + \Gamma^2_{12} a v' + \Gamma^2_{12} b u' + \Gamma^2_{22} b v' f_v \\
&= [(a' + (0)(au') + (0)(av') + (0)(bu') + (-sinucosu)(bv')] f_u \\
&+ [b' + (0)(au') + (1/tanu)(av') + (1/tanu)(bu') + (0)(bv')] f_v \\
&= [a' - (sinucosu) bv'] f_u + [b' + (av' + bu')/tanu] f_v
\end{aligned}$$

For the covariant derivative $\frac{Dw}{dt} = 0$, since f_u and f_v are linearly independent (definition of basis vectors), so we need the coefficient of both f_u and f_v to be zero. This leads to a system of differential equations.

$$\begin{cases} a' - (sinu\ cosu)bv' & = 0 \\ b' + (av' + bu')/tanu & = 0 \end{cases} \tag{A.1}$$

For simplicity, consider the case where $u = u_0 \neq 0$ is a constant, then we can $u'(t) = 0$ and $v'(t) = 1$ (assuming unit speed). So, the above system of ODE can be further simplified as

$$\begin{cases} a' - (sinu_0\ cosu_0)b(1) & = 0 \\ b' + [a(1) + b(0)]/tanu_0 & = 0 \end{cases} \tag{A.2}$$

Then the first ODE of A.2 gives

$$a'(t) - (sinu_0 cosu_0)\ b(t)\ (1) = 0$$
$$a'(t) = (sinu_0 cosu_0)\ b(t)$$

and the second one of A.2

$$b'(t) + [a(t)\ (1) + b(t)\ (0)]/tanu_0 = 0$$
$$b'(t) = (-1/tanu_0)\ a(t)$$

differentiate both sides with respect to t, and substitute $a'(t)$

$$b''(t) = (-1/tanu_0)\ a'(t)$$
$$b''(t) = (-1/tanu_0)(sinu_0\ cosu_0)\ b(t)$$
$$b''(t) + cos^2 u_0\ b(t) = 0$$

Since this is a second order linear differential equation with constant coefficients, it has a general solution (involves solving the auxiliary equation $\lambda^2 + cos^2 u_0 = 0$, which has complex

roots $\lambda = \pm(cosu_0)i$*)*

$$b(t) = A\ cos[(cosu_0)t] + B\ sin[(cosu_0)t]$$

where A, B *are constants.*

We can try to solve A *and* B *by considering initial conditions and we assume* $u = u_0$ *fixed, then* $w(0)$ *is moving directed toward* f_v*, and so* $a(0) = 0$ *and* $b(0) = 1$

$$b(0) = Acos(0) + Bsin(0)$$
$$A = 1$$

To solve B*, since* $b(t) = cos[(cosu_0)t] + Bsin[(cosu_0)t]$*, differentiate it gives* $b'(t) = -cosu_0\ sin[(cosu_0)t] + (Bcosu_0)\ cos[(cosu_0)t]$ *hence substitute back to the ODE involving* $b'(t)$ *and* $a(t)$*:*

$$b'(t) = (-1/tanu_0)\ a(t)$$
$$-cosu_0\ sin[(cosu_0)t] + (Bcosu_0)\ cos[(cosu_0)t] = (-1/tanu_0)\ a(t)$$
$$a(t) = sinu_0\ sin[(cosu_0)t] - Bsinu_0\ cos[(cosu_0)t]$$

and at $t = 0, a(0) = 0$*, so*

$$a(0) = sinu_0\ sin(0) - Bsinu_0\ cos(0)$$
$$0 = -Bsinu_0$$
$$B = 0$$

So, we have

$$\begin{cases} a(t) = sinu_0\ sin[(cosu_0)t]. \\ b(t) = cos[(cosu_0)t] \end{cases} \tag{A.3}$$

Hence, the solution in this special case is

$$\begin{aligned} w(t) &= a(t)f_u + b(t)f_v \\ &= sinu_0 \ sin[(cosu_0)t]f_u + cos[(cosu_0)t]f_v \end{aligned}$$

This represents the parallel transport of initial vector $w(0)$ *moving along the curve* $\gamma(t) = f(u(t), v(t)) = f(u_0, t)$.

Note that we put a special initial case and assumption (fixed $u = u_0$*). However, the overall approach is also applicable generally, but the solution may not have an explicit solution. And we may need to use numerical methods to solve the system of differential equations IVP problem.*

B Programming Code

We used Python programming language to conduct our statistical analysis and simulations. The followings are the Google Colab link that allows to run the code interactively.

The main notebook, that contains the following sections:

https://colab.research.google.com/drive/1nOPCpyRmpBNrqUMqxDVnoIQjT-0ZAWSn?usp=share_link

1. *Basic functions*
2. *Unrolling, unwrapping, and wrapping functions*
3. *Graph plotting function*
4. *Curve fitting method*
5. *Forecasting*

And for examples or extra materials:

https://colab.research.google.com/drive/1UnIWANU8mxNNFV1jiEWLe4s5ty48l1hh?usp=sharing

C References

[1] *Banchoff, T. F., & Lovett, S. (2022). Differential geometry of curves and surfaces. CRC Press.*

[2] *Bär, C. (2010). Elementary differential geometry. Cambridge University Press.*

[3] *Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.*

[4] *Brownlee, J. (2016). Machine learning mastery with Python: understand your data, create accurate models, and work projects end-to-end. Machine Learning Mastery.*

[5] *Bu-Qing, S., & Ding-Yuan, L. (2014). Computational geometry: curve and surface modeling. Elsevier.*

[6] *Chihara, L. M., & Hesterberg, T. C. (2022). Mathematical statistics with resampling and R. John Wiley & Sons.*

[7] *Cox, T. F. (2005). An introduction to multivariate data analysis. (No Title).*

[8] *Davison, A. C., & Hinkley, D. V. (1997). Bootstrap methods and their application (No. 1). Cambridge university press.*

[9] *Do Carmo, M. P. (2016). Differential geometry of curves and surfaces: revised and updated second edition. Courier Dover Publications.*

[10] *Dryden, I. L., & Mardia, K. V. (2016). Statistical shape analysis: with applications in R (Vol. 995). John Wiley & Sons.*

[11] *Fisher, N. I., Lewis, T., & Embleton, B. J. (1993). Statistical analysis of spherical data. Cambridge university press.*

[12] *Gaetan, C., & Guyon, X. (2010). Spatial statistics and modeling (Vol. 90). New York: Springer.*

[13] *Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., & Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. Advances in neural information processing systems, 31.*

[14] *Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). third. Bayesian data analysis.*

[15] *Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2*

[16] *Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.*

[17] *Hida, T., & Hitsuda, M. (1993). Gaussian processes (Vol. 120). American Mathematical Soc..*

[18] *Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.*

[19] *Inc., P. T. (2015). Collaborative data science. Montreal, QC: Plotly Technologies Inc. Retrieved from https://plot.ly*

[20] *Jupp, P. E., & Kent, J. T. (1987). Fitting smooth paths to spherical data. Journal of the Royal Statistical Society Series C: Applied Statistics, 36(1), 34-46.*

[21] *Kay, D. C. (2011). Tensor calculus. (No Title).*

[22] *Kendall, D. G., Barden, D., Carne, T. K., & Le, H. (2009). Shape and shape theory. John Wiley & Sons.*

[23] *Kim, K. R., Dryden, I. L., Le, H., & Severn, K. E. (2021). Smoothing splines on Riemannian manifolds, with applications to 3D shape space. Journal of the Royal Statistical Society Series B: Statistical Methodology, 83(1), 108-132.*

[24] Kitagawa, T., & Rowley, J. (2022). von Mises-Fisher distributions and their statistical divergence. arXiv preprint arXiv:2202.05192.

[25] Lancaster, P., & Salkauskas, K. (1986). Curve and surface fitting. An introduction. London: Academic Press.

[26] Louis, M., Bône, A., Charlier, B., Durrleman, S., & Alzheimer's Disease Neuroimaging Initiative. (2017). Parallel transport in shape analysis: a scalable numerical scheme. In Geometric Science of Information: Third International Conference, GSI 2017, Paris, France, November 7-9, 2017, Proceedings 3 (pp. 29-37). Springer International Publishing.

[27] Marsden, J. E., & Tromba, A. (2003). Vector calculus. Macmillan.

[28] Morgan, F. (1998). Riemannian geometry: A beginners guide. AK Peters/CRC Press.

[29] Nina Miolane, Saiteja Utpala, Nicolas Guigui, Luís F. Pereira, Alice Le Brigant, Hzaatiti, Yann Cabanes, Johan Mathe, Niklas Koep, elodiemaignant, ythanwerdas, xpennec, tgeral68, Christian, Tra My Nguyen, Olivier Peltre, pchauchat, Jules-Deschamps, John Harvey, ... Marius. (2023). geomstats/geomstats: Geomstats v2.6.0 (2.6.0). Zenodo. https://doi.org/10.5281/zenodo.8089259

[30] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[31] Pinsky, M., & Karlin, S. (2010). An introduction to stochastic modeling. Academic press.

[32] Planas, R., Oune, N., & Bostanabad, R. (2020, August). Extrapolation with gaussian random processes and evolutionary programming. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 84003, p. V11AT11A004). American Society of Mechanical Engineers.

[33] Pressley, A. N. (2010). Elementary differential geometry. Springer Science & Business Media.

[34] Rasmussen, C. E., & Williams, C. (2006). Gaussian processes for machine learning the mit press. Cambridge, MA, 32, 68.

[35] Serfozo, R. (2009). Basics of applied stochastic processes. Springer Science & Business Media.

[36] Simonoff, J. S. (2012). Smoothing methods in statistics. Springer Science & Business Media.

[37] Smith, R. C. (2013). Uncertainty quantification: theory, implementation, and applications (Vol. 12). Siam.

[38] Stoyan, D., & Stoyan, H. (1994). Fractals, random shapes and point fields: methods of geometrical statistics. (No Title).

[39] Sutherland, W. A. (2009). Introduction to metric and topological spaces. Oxford University Press.

[40] Wang, J. (2020). An intuitive tutorial to Gaussian processes regression. arXiv preprint arXiv:2009.10862.