

# Flow Visualization on Curved Manifolds

BACHELORARBEIT

zur Erlangung des akademischen Grades

**Bachelor of Science**

im Rahmen des Studiums

**Medizinische Informatik**

eingereicht von

**Jakob Troidl**

Matrikelnummer 01525071

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. M. Eduard Gröller

Mitwirkung: Dipl.-Ing. Dr.techn. Peter Rautek

Wien, 28. Mai 2019

---

Jakob Troidl

---

M. Eduard Gröller



# Flow Visualization on Curved Manifolds

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Science**

in

**Medical Informatics**

by

**Jakob Troidl**

Registration Number 01525071

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. M. Eduard Gröller

Assistance: Dipl.-Ing. Dr.techn. Peter Rautek

Vienna, 28<sup>th</sup> May, 2019

---

Jakob Troidl

---

M. Eduard Gröller



# Erklärung zur Verfassung der Arbeit

Jakob Troidl

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 28. Mai 2019

---

Jakob Troidl



# Danksagung

Ich möchte vor allem meinem Betreuer Peter Rautek für seinen wertvollen und hilfreichen Rat bei der Implementierung und beim Schreiben dieser Arbeit danken. Außerdem möchte ich dem gesamten Team der High Performance Visualization Group der King Abdullah University of Science & Technology dafür danken, dass ich meine Bachelorarbeit im Rahmen eines echten Forschungsprojekts schreiben konnte. Deweiteren gilt mein Dank Eduard Gröller sowohl für seine Betreuung als auch für die Möglichkeit, diese Arbeit im Ausland schreiben zu können. Zu guter Letzt möchte ich meinen Freunden und Kollegen, aber insbesondere meinen Eltern und Großeltern danken, die mir mein Studium durch ihre Unterstützung ermöglicht haben.



# Acknowledgements

Foremost I want to thank my supervisor Peter Rautek for his valuable and helpful input during implementation and writing. Furthermore, I want to thank the whole team of the High-Performance Visualization Group at King Abdullah University of Science & Technology for allowing me to write my bachelor thesis as part of a real research project. I want to extend my gratitude to Eduard Gröller for his supervision and for giving me the possibility to write this thesis abroad. I want to thank all my friends and colleagues who supported me during my studies especially my parents and grandparents who gave me the opportunity to accomplish my studies through their continuous support.



# Kurzfassung

Klimaforscher simulieren Wind- und Meeresströmungen, um diese besser zu verstehen. Dafür benötigen sie Visualisierungen, um ihre Ergebnisse zu validieren und weiter zu verbessern. In dieser Arbeit stellen wir ein Rahmenwerk vor, das instationäre 2D-Vektorfelder auf gekrümmten Oberflächen visualisiert. Unser Rahmenwerk berechnet die Visualisierungen nicht in 3D, sondern intrinsisch in 2D. Als Beispiel zeigen wir Visualisierungen von 2D Windströmungen auf der Kugel. Wir verwenden Methoden der differentiellen Geometrie, um Line Integral Convolution und Path Lines auf gekrümmten Oberflächen zu berechnen. Während Line Integral Convolution einen Überblick über einen Zeitschritt eines instationären Vektorfeldes bietet, geben Path Lines einen detaillierteren Einblick. Wir animieren die Line Integral Convolution Bilder und Path Lines, um die Richtung der Strömung zu zeigen. Außerdem beleuchten wir die Path Lines, um die räumliche Wahrnehmung zu verbessern. Unsere Visualisierungen werden in Echtzeit berechnet und können interaktiv verwendet werden.



# Abstract

Climate researchers often use simulations to generate 2D vector fields of wind or ocean currents. They need visualization tools to validate and further improve their research. In this work, we present a framework that is capable of visualizing unsteady 2D flow fields on curved surfaces. An important property of our framework is that it works intrinsically in 2D, instead of in 3D ambient space. Our primary example is the visualization of 2D geophysical flow on the 2-sphere. We build on methods from differential geometry to compute line integral convolution and path lines intrinsically on curved surfaces. While line integral convolution provides an overview of one time step of an unsteady flow field, path lines give us a more detailed insight into an unsteady flow field. We animate the line integral convolution images and the path lines to show the direction of the flow. Furthermore, we illuminate the path lines to improve spatial perception. Our visualizations are all computed in real time and can be used interactively.



# Contents

<b>Kurzfassung</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Contents</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Goal of the Thesis . . . . .	1
1.3 Structure of the Work . . . . .	2
<b>2 Related Work</b>	<b>3</b>
2.1 Line Integral Convolution . . . . .	3
2.2 Illuminated Path Lines . . . . .	4
<b>3 Mathematical Background</b>	<b>5</b>
3.1 Definition of a Manifold . . . . .	5
3.2 Charts of a Manifold . . . . .	6
3.3 Tangent Vector Spaces . . . . .	7
3.4 Chart Induced Basis . . . . .	9
3.5 Change of Components . . . . .	10
3.6 The Metric Tensor . . . . .	11
<b>4 Flow on the 2-Sphere</b>	<b>13</b>
4.1 The 2-Sphere as a Manifold . . . . .	13
4.2 The Metric Tensor of the 2-Sphere . . . . .	13
<b>5 Visualization of Geophysical Flow on the 2-Sphere</b>	<b>17</b>
5.1 2D Geophysical Flow Data . . . . .	17
5.2 Euler Integration . . . . .	18
5.3 Line Integral Convolution . . . . .	19
5.4 Illuminated Path Lines . . . . .	20
<b>6 Conclusion</b>	<b>27</b>
	xv

6.1	Limitations . . . . .	27
6.2	Future Work . . . . .	27
	<b>List of Figures</b>	<b>29</b>
	<b>Bibliography</b>	<b>31</b>

# Introduction

Climate change is a significant challenge for our generation. Scientists around the world are researching to understand why climate behaves as it does. They use supercomputers to build precise models of the climate. These simulations produce large scale data sets of wind flows or ocean currents. Having access to interactive visualization systems that can handle these large data sets is essential for climate researchers to gain a better understanding of their research. In this work, we present a framework that is capable of visualizing big flow data sets on curved surfaces interactively. We focus on the visualization of 2D geophysical data on a sphere. We perform all computations intrinsically, which means that we only take the surface of a given shape into account and not the surrounding 3D space. In this project, we chose to perform all computations intrinsically in 2D because climate simulations are often computed in 2D as well and output 2D intrinsic vector fields with longitude, latitude coordinates.

## 1.1 Motivation

Flow data often comes from simulations that compute 2D vector fields. We show algorithms that visualize this data on curved manifolds by taking only 2D vectors into account. A lot of papers that have studied flow visualization on curved manifolds use 3D vector fields instead. This leads to algorithms where integral curves have to be projected to the 2D surface. While it is mathematically simpler to work in 3D ambient space of a 2D surface embedding, we show that our purely intrinsic approach is very efficient.

## 1.2 Goal of the Thesis

The goal of this work is to describe two very important flow visualization methods from a purely intrinsic point of view. We show how to compute line integral convolution and illuminated path lines on curved manifolds. We explicitly use 2D vector fields for

computing these visualizations, which has a positive impact on computation time and memory usage. Further, the intrinsic computations lend themselves to a very efficient massively parallel implementation on modern graphics hardware (GPUs).

### 1.3 Structure of the Work

After an overview of the state of the art of flow visualization on curved surfaces, we present an introduction to the underlying mathematical concepts. We make the reader familiar with the idea of manifolds and what to keep in mind when dealing with vector fields on manifolds. After that, we discuss methodology, implementation and results for line integral convolution and illuminated path lines. In all our examples, the important case of the sphere and 2D geophysical flow are the primary example.

## Related Work

This chapter is divided into related work regarding line integral convolution (LIC) and illuminated path lines. We present several different approaches and compare them to our work.

### 2.1 Line Integral Convolution

When dealing with flow on curved surfaces, we have to distinguish between intrinsic and extrinsic computation approaches. Most of the published work about LIC takes an extrinsic approach. For example, Laramée et al. [LJH03] visualize 3D unsteady flow on arbitrarily curved surfaces. Their method is called image based flow visualization, which generates output images similar to LIC. Indeed the computation of the textures differs from the LIC algorithm. They associate 3D flow data with triangulated surface meshes by mapping a 2D texture to the surface. Our work is able to handle parameterized surfaces and we use an intrinsic approach suitable for climate simulations that produce 2D vector fields. Van Wijk [Wij03] presents a similar image based approach.

Mao et al. [MKFI97] introduce another extrinsic and view dependent technique to compute LIC in image space with an arbitrarily complex topology. They intersect rays, starting from each pixel, with the surface and evaluate the convolution integral only at the visible intersections.

In their work *LIC on Surfaces* Stalling and Hege [SH97] present both an intrinsic and an extrinsic approach. In contrast to our intrinsic method they assume that the manifold can be parameterized by one 2D Euclidian coordinate system. Our method can deal with manifolds that do not satisfy this condition. For example, the sphere cannot be represented by one coordinate system which we explain later in more detail. Moreover, Stalling and Hege adapt the kernel size of the LIC filter to avoid distortions in the resulting texture. Since we have a metric tensor for our surface, we do not need to do this because the metric allows us to take steps of constant sizes on the manifold. The

extrinsic approach of Stalling and Hege is to compute a texture segment for each triangle of an arbitrary surface. The major challenge here is to create a continuous texture over the whole triangulated surface.

## 2.2 Illuminated Path Lines

Integral curves such as stream lines, path lines, streak lines and time lines are helpful tools to visualize vector fields. We are focusing on illuminated path lines in an unsteady flow field. Several papers proposed techniques to place and illuminate curves in an optimal manner. Jobard and Lefer [JL97] present an algorithm that distributes stream lines evenly in 2D space. Since they are working with steady flow fields they only apply their algorithm for stream lines and not for other integral curves. Turk and Banks [TB96] introduce an image guided approach to place stream lines. They improve the stream line placement by minimizing an energy function that takes a low-pass filtered version of the current image and compares it with the desired visual density. In contrast to these approaches, we seed a small number of path lines close to each other and interactively compute the path lines starting around the mouse cursor. Line illumination plays an essential role when it comes to spatial perception. However, lines do not have normals, which can be used to compute illumination. Zoeckler et al. [ZSH96] are using the tangent vectors of the curves to formulate rules for the ambient, diffuse and specular illumination components. They apply conventional texture mapping techniques available in regular graphics hardware to map colours to the integral lines. In our work, we use their approach to illuminate path lines. Mallo et al. [MPSS05] present simplified expressions of the Blinn-Phong-Model, which approximate the lines as infinitely small tubes.

All these methods compute integral curves in Euclidian space. Since we are using methods from differential geometry, we compute the integral curves inside the charts. We can, at least locally, compute the path lines in  $\mathbb{R}^2$ .

# Mathematical Background

In this chapter, we describe the underlying mathematical concepts from differential geometry. We will try to introduce them as general as possible using the language of differential geometry. Later, we specialize them for our purpose of flow visualization on the sphere. The advantage of this approach is that the mathematical concepts can be used unchanged for more advanced cases. Our description and notation are based on the first two chapters of the work of B. Schutz [Sch80] on *Geometrical methods of mathematical physics* and lecture notes of *Geometric Anatomy of Theoretical Physics* by Frederic Schuller [Rea19] [Sch19]. The reader should be equipped with a basic knowledge of Topology and Linear Algebra.

## 3.1 Definition of a Manifold

In order to understand the definition of a manifold, the reader should be familiar with topological spaces and open sets. Our geometric objects on which we want to visualize flow (e.g., the 2-sphere) make use of the mathematical concept of a manifold. A set  $M$  is called a *manifold* if there is an open neighbourhood  $U$  for each point  $p$  in  $M$  such that a continuous and bijective mapping to an open set of  $\mathbb{R}^n$  exists.  $M$  and  $\mathbb{R}^n$  are considered to be topological spaces. The dimension of the manifold is defined by  $n$ . Let us have a closer look at the terms *continuous* and *bijective*. A map  $f : M \rightarrow N$  between two topological spaces  $M$  and  $N$  is called *continuous* if and only if the inverse image of every open set of  $N$  is open in  $M$  as well. In other words,  $f$  has to map every open set in  $M$  to an open set in  $N$ , as well as  $f^{-1}$  has to map every open set in  $N$  to an open set in  $M$ .  $f$  is called *bijective* if every element in  $M$  maps exactly to one element in  $N$ . If  $f^{-1}$  is continuous too,  $f$  is called a *homeomorphic* function. This definition allows us to view  $M$  locally like  $\mathbb{R}^n$  because of  $f : M \rightarrow \mathbb{R}^n$ . We can map every point  $p$  in a local region  $U$  of the manifold  $M$  to an  $n$ -tuple of real numbers  $(x^1(p), \dots, x^n(p)) \in \mathbb{R}^n$ , called the

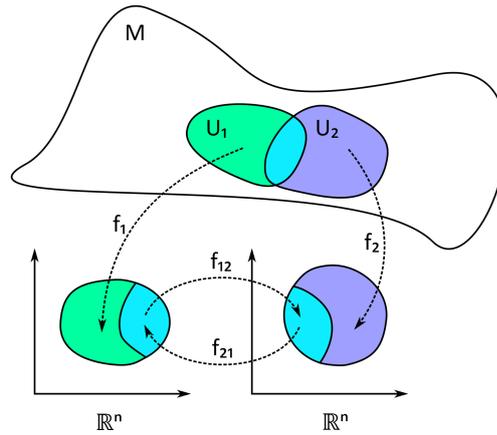


Figure 3.1: A manifold  $M$  with its co-ordinate maps  $f_1$  and  $f_2$ .  $U_1$  and  $U_2$  are subsets of  $M$ .  $f_{12}$  and  $f_{21}$  are chart transition maps. **Source:** [Jon19]

*co-ordinates* of  $p$  under the map  $f$ . A common way to write these coordinates is to place the indices as superscripts.

## 3.2 Charts of a Manifold

One open neighbourhood  $U$  does not necessarily include all points of  $M$ . Typically there is no single open neighbourhood that contains all points of  $M$ . For instance the 2-sphere needs at least 2 maps to cover it.

### 3.2.1 Definition of a Chart and an Atlas

We need to define open neighbourhoods on  $M$ , such that every point  $p$  on  $M$  is covered by at least one open neighbourhood. A *chart* is defined by an open neighbourhood  $U_i$  and its map  $f_i$ . A set of charts that covers the whole manifold  $M$  is called an *atlas*  $\mathcal{A}$ . A more formal definition is given as a collection of charts  $\mathcal{A} := \{U_i, f_i \mid i \in \mathcal{A}\}$  such that

$$\bigcup_{i \in \mathcal{A}} U_i = M.$$

Note that  $\mathcal{A}$  is a set of indices for all charts in  $\mathcal{A}$ . In Figure 3.1 two subsets of  $M$  are shown.  $U_1$  and  $U_2$  are overlapping charts that do not cover the whole manifold. Therefore, they are not an atlas but could be part of one. Both charts have a mapping  $f_i$  to a subspace of  $\mathbb{R}^n$ . adjacent charts of an atlas must have overlaps which directly follows from the definition of a chart as an open set. Figure 3.1 shows the overlap between  $U_1 \cap U_2$  in turquoise.

### 3.2.2 Chart Transition Maps

All points that are inside the intersection between two charts can be represented by two different mappings. Therefore they can be expressed with two different pairs of coordinates. For example consider a point  $p \in U_1 \cap U_2$  in Figure 3.1.  $f_1$  maps  $p$  to a point  $x = (x^1, x^2, \dots, x^n)$  in  $\mathbb{R}^n$ .  $f_2$  maps  $p$  to another point  $y = (y^1, y^2, \dots, y^n)$  in  $\mathbb{R}^n$ . A chart transition map describes the relationship between  $x$  and  $y$ . To construct a chart transition map from  $x$  to  $y$  we first take  $f_1^{-1}(x)$  which gives us a unique point  $s$  on  $M$ .  $s$  is unique on  $M$  because  $f_1$  is a homeomorphic function. Now  $f_2(s)$  takes us from  $M$  to  $y$ . We can formalize the chart transition map  $f_{12}$  as  $f_{12} = f_2(f_1^{-1}(x))$ . Another notation for  $f_2$  after  $f_1^{-1}$  is  $f_2 \circ f_1^{-1}$ , which is a function from  $\mathbb{R}^n \rightarrow \mathbb{R}^n$ .

## 3.3 Tangent Vector Spaces

We first introduce curves and smooth functions on manifolds before having a closer look at tangent vector spaces on manifolds. This definition of vectors is used in differential geometry and might be different from what many readers associate with vectors.

### 3.3.1 Smooth Functions on Manifolds

A function  $f$  on a manifold  $M$  assigns a real number to every point on  $M$ . For one input point on  $M$   $f$  assigns a real number as the output. The more formal definition of a smooth function on  $M$  is given as an infinite dimensional vector space over  $\mathbb{R}$  over the underlying set of all smooth functions  $\mathcal{C}^\infty(M)$ :

$$\mathcal{C}^\infty(M) := \{f : M \rightarrow \mathbb{R} \mid f \text{ is smooth}\}.$$

If  $f$  is infinite differentiable, it is called smooth. Further, we define addition for two smooth functions  $f$  and  $g$  and point wise scalar multiplication with any  $\lambda \in \mathbb{R}$

$$(f + g)(p) := f(p) + g(p)$$

$$(\lambda f)(p) := \lambda f(p)$$

### 3.3.2 Smooth Curves on Manifolds

We define a smooth curve as a mapping  $\gamma : \mathbb{R} \rightarrow M$  from an open set in  $\mathbb{R}$  onto the manifold  $M$ . This definition does not yet say anything about the direction of the curve on  $M$ . Figure 3.2 shows an image of a curve on a manifold.

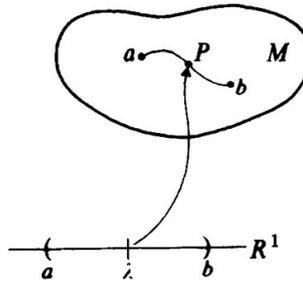


Figure 3.2: A curve on a manifold.  $a$  and  $b$  are the boundaries of the open set in  $\mathbb{R}$ .

**Source:** [Sch80]

### 3.3.3 A Vector in Linear Algebra

Vectors are often described as a direction in Euclidian space with a certain length. This is only one kind of vector. Generally speaking, a vector is an element of a vector space. To construct a vector space, vector addition and scalar multiplication must be defined. These two rules together, with eight other axioms, form a vector space. These axioms are the associativity and commutativity of addition, the existence of an identity and inverse element for the addition, the associativity of scalar multiplication, the distributivity of scalar and vector sums and the existence of an identity element in scalar multiplication. The most important concept to remember is that a vector is an abstract element of a vector space. A vector space can be anything that follows the axioms above, for example, functions in Euclidian space or diagonal matrices.

### 3.3.4 Tangent Vectors

With a good understanding of curves, functions and vector spaces we are well prepared to have a closer look at tangent vectors on manifolds. Intuitively speaking the velocity at which a function changes along a curve  $\gamma$  at  $p$  is the *tangent vector*  $X_{\gamma,p}$ . Often the tangent vector is depicted as a geometrical arrow with a direction and a length. Such depictions are only well defined in the two- and three dimensional Euclidian space and not on manifolds in general. That is why we define the tangent vector  $X_{\gamma,p}$  of the curve  $\gamma$  at point  $p$  as a directional derivative of a linear map

$$X_{\gamma,p} : \mathcal{C}^\infty(M) \xrightarrow{\sim} \mathbb{R}.$$

Here, we consider  $\mathbb{R}$  to be a 1-dimensional vector space over the real numbers. The linear map  $X_{\gamma,p}$  is defined as taking a function  $f \in \mathcal{C}^\infty(M)$  and mapping it to a real number by taking the derivative along the curve  $\gamma$  at  $p$ :

$$f \mapsto (f \circ \gamma)'(\lambda_0).$$

$f \circ \gamma$  is the function composition  $f$  after  $\gamma$  and equivalent to  $f'(\gamma(\lambda_0))$ . Since  $f \circ \gamma$  is a function from  $\mathbb{R} \rightarrow \mathbb{R}$  we can take the derivative like in one-dimensional calculus. This mapping  $X_{\gamma,p}$  is characteristic for a curve  $\gamma$  at a point  $p$  that is given by the parameter of the curve  $\lambda$ . Hence it is called a tangent vector of  $\gamma$  at  $p \in M$ . Note that we are not limited to a certain function  $f$ . In other words, the tangent vector is one way to describe the derivative of every smooth function on  $M$ .

### 3.3.5 Tangent Spaces

Having the concept of a tangent vector defined at  $p$  we can think of the corresponding vector space at  $p$ . The tangent vector space  $T_pM$  is the set of all tangent vectors at  $p$  on  $M$

$$T_pM := \{X_{\gamma,p} \mid \gamma \text{ is a smooth curve through } p\}.$$

To make  $T_pM$  a vector space, we need to define the operations  $\oplus$  (vector addition) and  $\odot$  (scalar multiplication) which have to fulfill several conditions. Vector addition and scalar multiplication are defined as follows

$$\oplus : T_pM \times T_pM \rightarrow T_pM$$

$$(X_{\gamma,p} \oplus X_{\delta,p})(f) := X_{\gamma,p}(f) + X_{\delta,p}(f)$$

$$\odot : \mathbb{R} \times T_pM \rightarrow T_pM$$

$$(\lambda \odot X_{\gamma,p})(f) := \lambda \cdot X_{\gamma,p}(f)$$

where  $+$  and  $\cdot$  denote the usual operations on the vector space  $\mathbb{R}$ .

## 3.4 Chart Induced Basis

Every vector in  $T_pM$  can be represented as a linear combination of some basis vectors. We may use specific basis vectors that are solely defined by the choice of a chart. May  $(U, x)$  be a chart on  $M$  where  $U$  is a subset of  $M$  and  $x$  the co-ordinate map of the chart. The co-ordinate maps  $x^a : U \rightarrow x(U)$  are smooth functions on  $U$ . Note that  $x^a$  maps  $M \rightarrow \mathbb{R}$ . For example, a two-dimensional sphere  $S^2$  would have two coordinate maps  $x^1$  and  $x^2$ . The number of basis vectors is equal to the number of dimensions of the manifold. We define the basis vectors of  $T_pM$  as a set

$$\left\{ \left( \frac{\partial}{\partial x^a} \right)_p \mid 1 \leq a \leq \dim M \right\}$$

where  $\left( \frac{\partial}{\partial x^a} \right)_p \in T_p M$ . Hence it is a set of tangent vectors  $X_{\gamma_{(a),p}}$  that is dependent on a chart  $U$  a point  $p \in U$  and the functions  $x^a$ . We define the basis vectors by using the special chart induced curves  $\gamma_{(a)}$

$$\gamma_{(a)}(t) := x^{-1} \circ (0, \dots, 0, t, 0, \dots, 0)_a.$$

Note that  $(0, \dots, 0, t, 0, \dots, 0)_a$  is an element of  $\mathbb{R}^d$  that has  $t$  at the  $a$ -th position. Let us put the definition of chart induced curves  $\gamma_{(a)}$  in the context of the other definitions. Previously we defined a curve as a map  $\mathbb{R} \rightarrow M$ . This is exactly what the curves  $\gamma_{(a)}$  are doing here.  $\gamma_{(a)}$  takes  $t \in \mathbb{R}$  as an input parameter and outputs a point on the manifold by inverting the chart map  $x$ . Using the inverse chart map that operates on  $\mathbb{R}^d \rightarrow M$  is valid because it is a homeomorphic function. This definition is possible because we are dealing with finite dimensional manifolds. All basis vectors of a tangent space  $T_p M$  at  $p$  are now defined by

$$X_{\gamma_{(a),p}} : (x^a \circ \gamma_a)'(t) \rightarrow \mathbb{R}.$$

Every basis vector is only valid in the vector space  $T_p M$  at a specific point  $p$  covered by the chart  $U$ .

### 3.5 Change of Components

Now we have a tool to describe every vector  $X_{\gamma,p} \in T_p M$  as a linear combination of the basis vectors of  $T_p M$  in a chart dependent manner. We now need to find a rule that tells us how to change the components of a vector, when the basis vectors are changed. Particularly we need to change the components of a tangent vector located at a point in an overlapping chart region. Let  $\delta_x^j$  be the components of a tangent vector with respect to a chart  $(U, x)$  induced basis. We want to compute the change of these components with respect to a new chart  $(V, y)$  induced basis. We need the objects  $\left( \frac{\partial y^i}{\partial x^j} \right)_p$ :

$$\left( \frac{\partial y^i}{\partial x^j} \right)_p = \partial_i (y^i \circ x^{-1})(x(p))$$

These are the component wise partial derivatives of the chart transition maps from  $U$  to  $V$ . We define our new components  $\delta_y^i$  with respect to the chart  $(V, y)$  induced basis:

$$\delta_y^i = \sum_{j=1}^{\dim M} \delta_x^j \left( \frac{\partial y^i}{\partial x^j} \right)_p.$$

This is the sum over all components with respect to the basis of  $U$  multiplied with the partial derivatives of the corresponding components of the chart transition map from  $U$  to  $V$ . The change of components allows us to speak about one vector with respect to different basis vectors in different charts.

### 3.6 The Metric Tensor

The primary motivation for mentioning metric tensors in this work is that they endow a manifold with a tool for measuring lengths. Consider a 2-sphere embedded in 3D Euclidian space. We are using the metric tensor to compute the length of a vector. On every point  $p$  on our manifold  $M$  we can define a metric tensor  $g_p$ . The metric tensor is basically a function  $g_p(X_p, Y_p)$  that takes two tangent vectors  $X_p$  and  $Y_p$  of the tangent space  $T_pM$  as an input and outputs a real number.  $g_p$  has to be bilinear, symmetric and nondegenerate [Car92]. In our case we have a second order metric tensor field  $g$  defined by an inner product  $\langle v, w \rangle := g(v, w)$  to each  $T_pM$ , which distributes smoothly over  $M$ . Note that  $v$  and  $w$  are both elements of a tangent space  $T_pM$ . The length of one vector  $w$  is then given by  $\|w\| := \sqrt{\langle w, w \rangle}$ .



# Flow on the 2-Sphere

## 4.1 The 2-Sphere as a Manifold

In this work, we focus on 2-dimensional manifolds. A common example for a 2D manifold is the 2-sphere. The 2-sphere is the surface of the sphere without its interior. The 2-sphere does not have a volume. The sphere is an example for showing the need of several charts. We cannot find one chart that covers the whole sphere. The biggest chart we can construct is an open set that contains the whole sphere except one point (e.g. the south pole). If we included this point, there would be no  $f$  that is homeomorphic to  $\mathbb{R}^2$ . Hence we need at least two charts to cover the whole sphere. In our framework, we choose to use six charts to reduce distortions. Four charts are arranged symmetrically around the equator, and one chart overlays each pole. Figure 4.1 shows our arrangement of charts around a 3-dimensional embedding of the sphere.

## 4.2 The Metric Tensor of the 2-Sphere

In the arrangement of our six charts that are the orthographic projection of the sphere, each chart  $U \subset \mathbb{R}^2$  is defined inside a disk of radius  $r$  in the  $(u, v)$  plane. All  $(u, v)$ -coordinates in  $U$  have to satisfy  $U := \{(u, v) : u^2 + v^2 \leq r^2\}$ . We use an *inclusion map*  $\iota$  for each chart from  $\mathbb{R}^2$  (intrinsic view) to  $\mathbb{R}^3$  (extrinsic view) to define the metric:

$$\iota_{(x,y)} : U \subset \mathbb{R}^2 \hookrightarrow \mathbb{R}^3$$

$$(u, v) \mapsto (u, v, w).$$

Note that  $w$  above is defined as  $w := \sqrt{r^2 - u^2 - v^2}$ . To avoid severe distortions in a chart we use it only if  $u^2 \leq w^2$  and  $v^2 \leq w^2$ . This rule defines the optimal region which

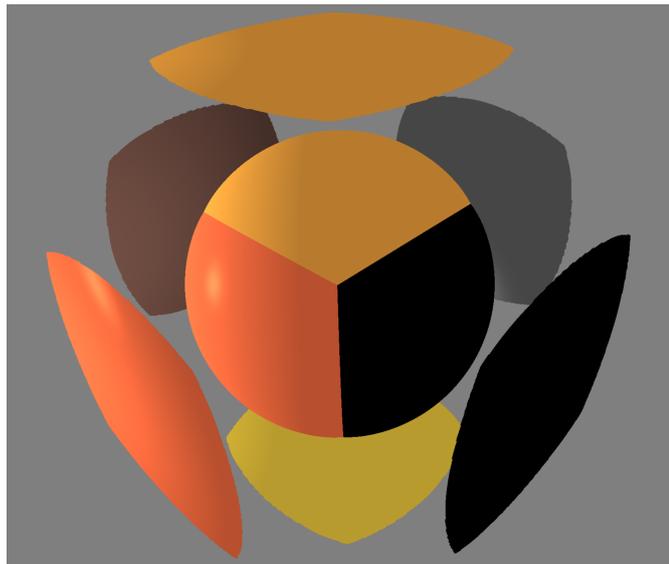


Figure 4.1: The different regions of the sphere covered by charts. Six charts align symmetrically around the sphere as planes in  $\mathbb{R}^2$ . All adjacent charts are overlapping.

is covered by a chart. Figure 4.1 shows these charts and their projections on the 2-sphere. The basis vectors  $\{\frac{\partial}{\partial x^i}\} = \partial_i = e_i, i \in \{1, 2\}$  in each chart are intrinsically defined by the components  $(1, 0)$  and  $(0, 1)$ . Figure 4.2 shows the two chart induced curves  $\gamma_{(a)}$  that are used to define a chart induced basis at a certain point on the manifold, on one chart as well as on the 2-sphere. From an extrinsic perspective  $e_1$  and  $e_2$  map to the partial derivative of the corresponding  $\iota$ , i.e.,

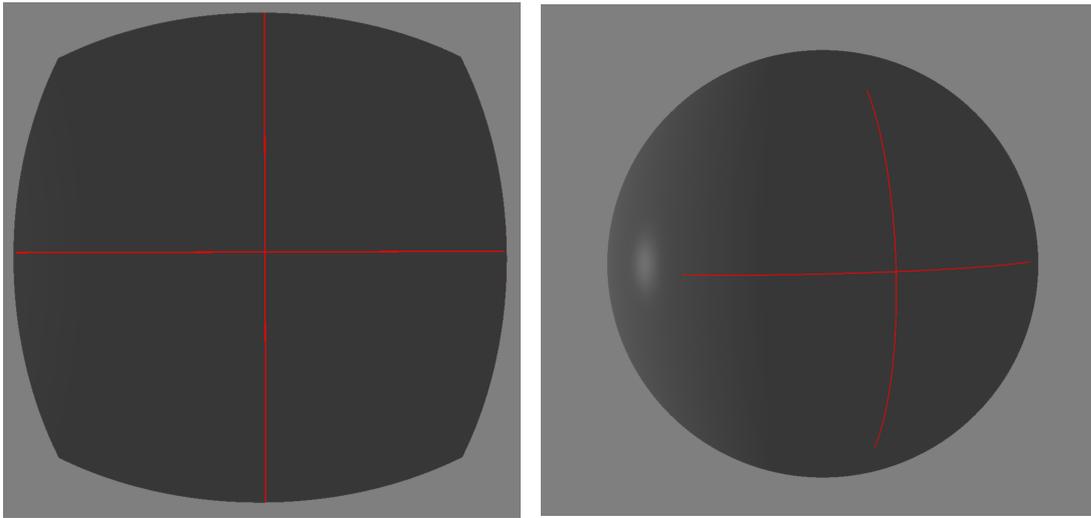
$$\tilde{e}_1|_{(u,v)} = \begin{pmatrix} 1 \\ 0 \\ -u/w \end{pmatrix}, \tilde{e}_2|_{(u,v)} = \begin{pmatrix} 0 \\ 1 \\ -v/w \end{pmatrix}.$$

The components of  $\tilde{e}_1$  and  $\tilde{e}_2$  are given in a Cartesian coordinate system in  $\mathbb{R}^3$ . Finally we can introduce the components  $g_{ij}$  of our intrinsic metric tensor  $g$  like

$$g_{ij}|_{u,v} = \frac{1}{w^2} \begin{bmatrix} \tilde{e}_1 \cdot \tilde{e}_1 & \tilde{e}_1 \cdot \tilde{e}_2 \\ \tilde{e}_2 \cdot \tilde{e}_1 & \tilde{e}_2 \cdot \tilde{e}_2 \end{bmatrix},$$

where  $\cdot$  is the dot product. This expression further simplifies to

$$g_{ij}|_{u,v} = \frac{1}{w^2} \begin{bmatrix} r^2 - v^2 & uv \\ uv & r^2 - u^2 \end{bmatrix}.$$



(a) Two curves on a chart

(b) Two curves projected on the sphere

Figure 4.2: Each curve is used to define one basis vector. The tangent vector space of these basis vectors is located on the point where both curves intersect.

In the case of the sphere  $g(w, w)$  gives the squared length for a vector  $w := (u, v)$  as:

$$g(w, w) = \begin{pmatrix} u & v \end{pmatrix} g_{ij} \begin{pmatrix} u \\ v \end{pmatrix}.$$

The metric tensor is defined on each chart. Even though we derive the metric tensor in  $\mathbb{R}^3$ , we nevertheless, use the metric entirely intrinsically.





# Visualization of Geophysical Flow on the 2-Sphere

This chapter describes the flow visualization techniques we have implemented in our framework. After a short description of our data set, we show the technique of Eulerian integration to compute integral curves of a vector field. LIC as well as path lines make use of this integration technique. Our description is based on the work *Numerical Algorithms* of Justin Solomon [Sol15]. After that, we present our methodology and results for line integral convolution and illuminated path lines.

## 5.1 2D Geophysical Flow Data

The data set we are using is provided by the Deutsches Klima Rechenzentrum (DKRZ). The data is given in the NetCDF-Format and includes wind data for the whole surface of the earth. We have a resolution of approximately 5 kilometres. The data set contains 96 time steps, one time step every 15 minutes which is the wind data of one day. It has a size of approximately 17 GB. The vectors are given in  $uv$  components on an ICON grid on the sphere. All components are defined for one chart and its basis vectors. Using one chart is possible because the 2-sphere is discretized. We transform this chart into our arrangement of six charts (Figure 4.1). We define two basis vectors for each chart and change the components of the vectors with respect to the new basis vectors. To compute all vectors with the correct length on a given position in a chart is only possible because we know the distortion of the metric tensor. More information about the data set and its representation can be found in the documentation of the ICON Grid of DKRZ [LL19].

## 5.2 Euler Integration

We use numerical methods to compute the path lines and the stream lines employed for LIC. Intuitively speaking we want to trace the path that a massless particle would take. If the vector field is steady over time the resulting line is called a stream line, if not it is called a path line. Euler integration is the simplest method to compute integral curves for a vector field. From a given starting position  $s_0$  on the manifold  $M$  we move one step  $dt$  in the direction of the vector  $V(s_0)$ . It is also valid to move in the backward direction  $-V(s_0)$ . One integration step takes us to a new position  $s_1$  on  $M$  from where we can repeat that process. Since  $s_1$  is not necessarily at a position where we have stored our discretized vector field, we compute the vector  $V(s_1)$  through interpolation of the surrounding vectors. We can formalize one forward integration step as:

$$s_{i+1} = s_i + dtV(s_i).$$

The method gets more accurate the smaller we chose  $dt$ . However, there is always an error since we cannot compute curves with an infinitely small  $dt$ . This inaccuracy is visualized in Figure 5.1. We always need to find a tradeoff between accuracy and performance. The smaller we chose  $dt$  the more expensive the computation gets. To compute curves, a starting position  $s_0$  and one or more termination conditions for the integration must be defined. We define these later for each visualization method. This description is valid for computations in Euclidian space. Since we are dealing with curved manifolds, we can only use Eulerian integration inside the charts. If the integration leaves a chart, we use the chart transition maps to continue the integration in another chart.

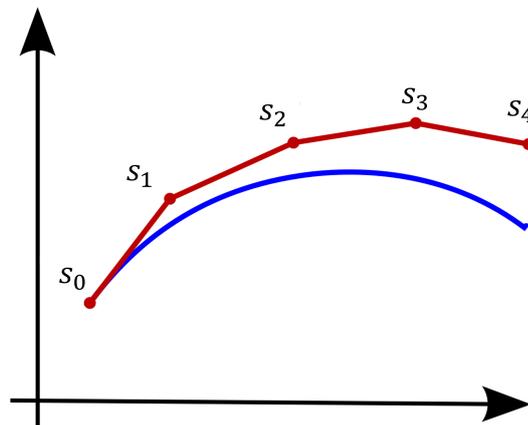


Figure 5.1: The red curve shows the Euler approximation of the blue curve. A smaller distance between the sample points  $s_i$  results in a more precise representation of the blue curve. **Source:** [Tak19]

## 5.3 Line Integral Convolution

### 5.3.1 Methodology

Line integral convolution [CL93] is well suited to provide an overview of one time step of a vector field. LIC is a texture based visualization technique which smudges a noise texture in the direction of the flow. Neighbouring texels along the flow have high coherence while normal to the flow direction there is no coherence. More generally speaking, we convolve a 2D noise image  $N$  along the stream lines of the flow. Since we want to display the LIC texture on the 3D embedding of a manifold, we compute the LIC texture per fragment in the chart and map it back to the curved surface. We compute a stream line with  $n$  steps for each fragment by making  $n/2$  forward and  $n/2$  backward Euler steps. We terminate the integration in a chart if we have reached the maximum number of integration steps or the integration comes to a chart boundary. In that case, we continue the integration in another chart. We obtain the integration starting position in the new chart by applying the respective chart transition map. To visualize the direction of the flow, we shift a time-dependent periodic filter function  $h(u, t)$  along the stream line. We weight each texel along the stream line with the corresponding value of the filter function. Through shifting the filter function over time, the LIC texture is animated (see Video [Tro19]). One has to be aware that the animation only shows the direction of the flow and not how particles evolve over time. Since the LIC texture is computed using stream lines it only shows one time step of our unsteady flow field. Figure 5.2 shows two time steps of our periodic filter function. In summary, we obtain the grey value of a texel at position  $x$  in the chart at time  $t$  by

$$LIC(x, t) = \int_{-1}^1 N(s_x(u))h(u, t)du$$

where  $s_x(u)$  is a parameterization of the stream line going through  $x$ . At  $u = -1$  is the start point of the stream line and at  $u = 1$  is the end point of the stream line. The noise image  $N$  takes the 2D position  $s_x(u)$  of the stream line to output a random grey value. Note that  $h(u, t)$  is either 1 or 0. When applying this method to each fragment, we get a LIC texture for the visible 3D embedding of the manifold. Additionally, we colour-code the length of the flow vector located at each texel. We determine the length of the vector at each texel by using the metric and mapping it to an RGBA colour value. The LIC grey value was then weighted with this colour. Furthermore, we allow the user to step through all time steps of the vector field by linearly interpolating the flow data between the different time steps.

### 5.3.2 Implementation

We wrote a framework in C++ and OpenGL that implements a GPU based version of this LIC algorithm. All fragment colours are computed in parallel in the fragment shader that has access to the vector field textures and the noise texture. Computing

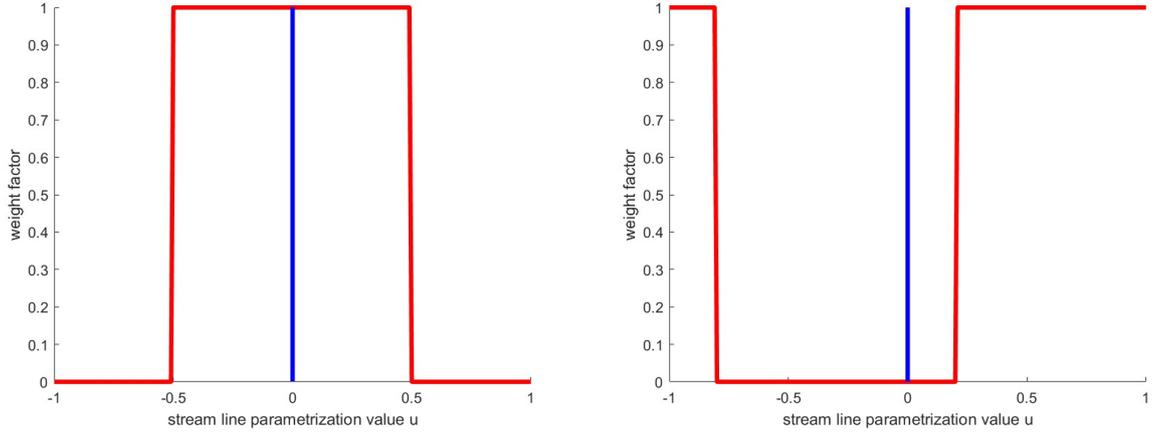


Figure 5.2: Two different time steps of our periodic filter function  $h(u, t)$ . Since we make  $n/2$  forward and  $n/2$  backward Euler steps, the blue vertical line represents the position of the texel on the stream line.

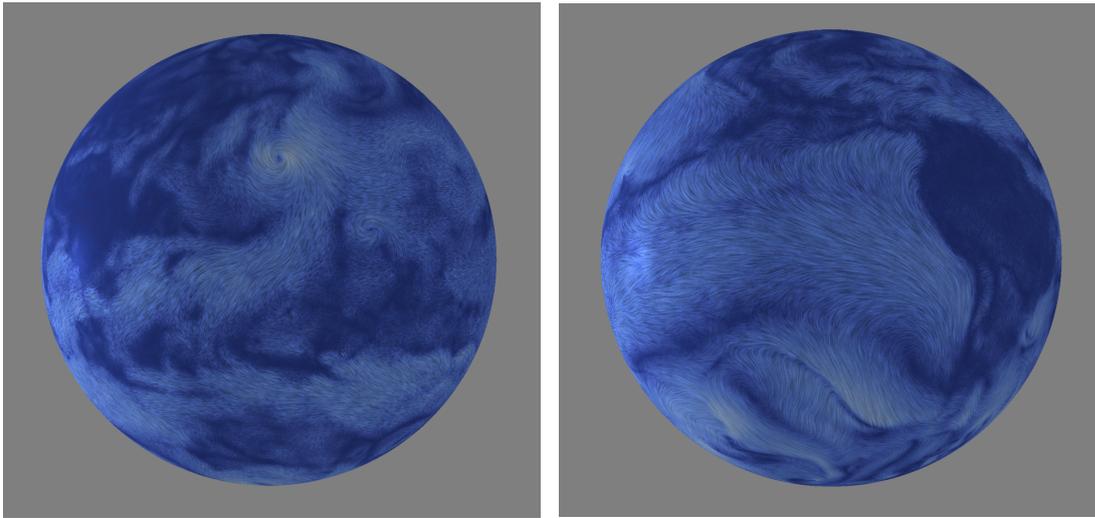
LIC on the GPU results in a massive performance speedup compared to CPU based implementations. Our visualization algorithm runs in real-time for 4k image resolutions using twelve 256x256 flow textures. 2 textures represent two discrete time steps for each of the 6 charts. We use 2 time steps per chart to be able to interpolate between them.

### 5.3.3 Results

Figure 5.3 shows the results of our LIC implementation. In Figure 5.3a we can observe a vortex structure close to Japan. A dark blue colour visualizes slow wind speed while white represents a high velocity. Close to the center of the vortex the wind velocities are relatively high while in the center of the vortex there is almost no wind activity. Especially Figure 5.3b shows that wind current arises more likely over the ocean than over land. Hence the shape of South America is visible in dark blue in Figure 5.3b. The accompanying video [Tro19] shows the animation of the LIC that visualizes the direction of the flow. In Figure 5.3c we can observe how wind currents change over time. Both pictures show the wind over the African continent. The physical time difference between these two pictures is around 12.5 hours.

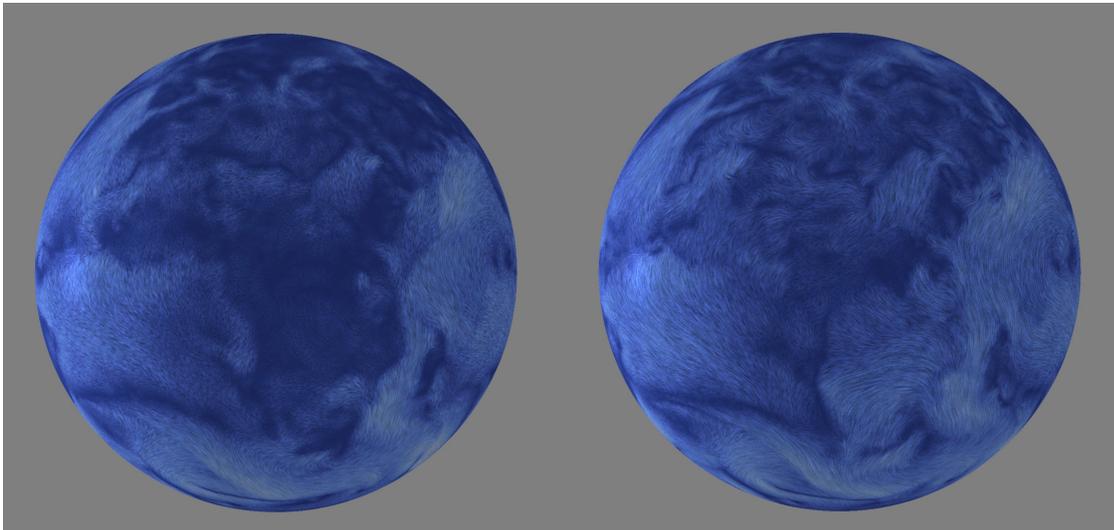
## 5.4 Illuminated Path Lines

While line integral convolution provides an overview of a snapshot of a vector field, path lines [Had19] let us gain a better understanding of how specific particles behave in a flow field. In this section, we explain how we compute and visualize path lines.



(a) A vortex structure close to Japan

(b) Wind currents at the South American coast



(c) Wind currents over the African continent at two different time steps

Figure 5.3: LIC visualization of the wind data set of the DKRZ

### 5.4.1 Methodology

Intuitively we can think of a path line as the trajectory that a massless particle takes through an unsteady flow field. We compute a single path line in a chart by making  $n$  forward Euler steps from a given starting position  $s_0$ . Unlike stream lines, with each integration step, we also advance the integration time. We always take a constant step forward in time. There are three possible reasons to terminate the integration of a path line in a chart:

- We have reached the end of the time domain
- We have reached the maximum number of integration steps
- We have reached a chart boundary. In this case, we restart the integration in another chart.

To improve the spatial perception of path lines, we illuminate them with an adapted Phong illumination algorithm proposed by Zoeckler et al. [ZSH96]. Since there are no normals on a line, we use the tangent vectors of the curve to compute the diffuse and the specular lighting component. We approximate the tangent  $T$  of each discrete point  $p_i$  on the path line as  $T_{s_i} = s_{i+1} - s_{i-1}$ . We compute the intensity  $I$  of each particular path line fragment as  $I = I_{\text{ambient}} + I_{\text{diffuse}} + I_{\text{specular}}$ , where  $I_{\text{ambient}}$  is a constant scalar value. The other components are defined as:

$$I_{\text{diffuse}} = k_d \sqrt{1 - (L \cdot T)^2},$$

$$I_{\text{specular}} = k_s (V \cdot R)^\alpha,$$

$$V \cdot R = (L \cdot T)(V \cdot T) - \sqrt{1 - (L \cdot T)^2} \sqrt{1 - (V \cdot T)^2},$$

where  $\cdot$  is the dot product. Note that  $L$  specifies the direction of the light,  $V$  the viewing direction and  $R$  the direction of the reflection of the incoming light.  $k_d$ ,  $k_s$  and  $\alpha$  are scalar weights.  $\alpha$  controls the size of the specular highlights. Moreover we developed three approaches to visualize the temporal meaning of path lines:

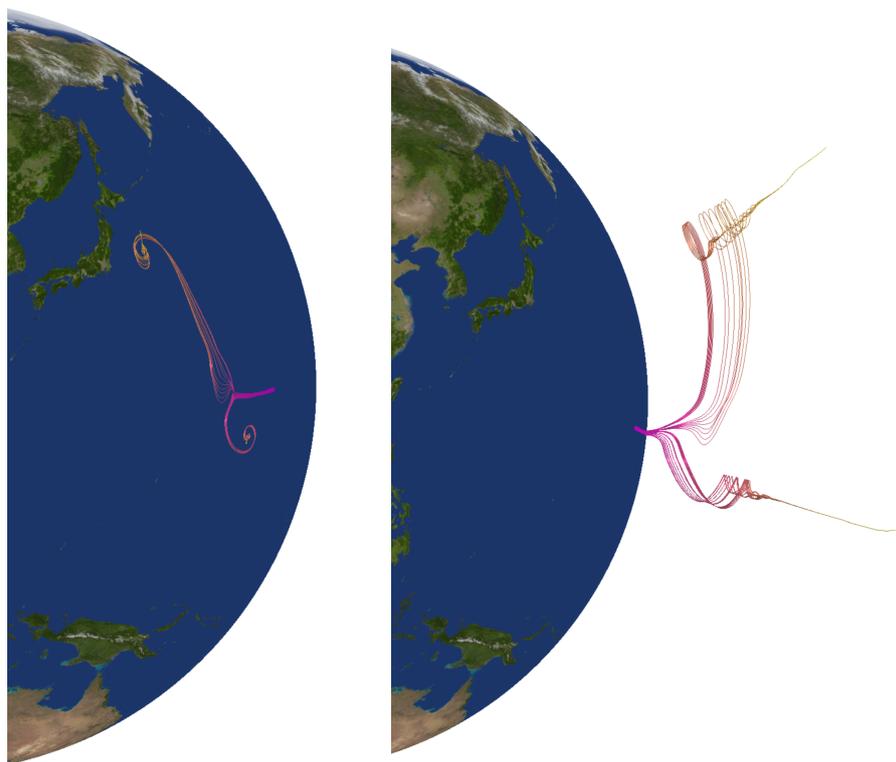
1. We colour coded the age of the particle on the path line. Pink areas in our results represent a young particle, while yellow areas show old particles.
2. By lifting the path lines from the surface of the sphere in the ambient 3D space, we add a time dimension. The higher the path line segment is lifted, the older is the particle at this segment. To visualize the correspondence between the surface of the sphere and the 3D path line, we draw the illustrative shadows of the path line on the sphere. The shadow is the projection of the 3D path line onto the sphere.
3. We animate the path lines by shifting a periodic opacity function along the path lines. We have implemented three animation modes. See our video [Tro19] for more details.

#### 5.4.2 Implementation

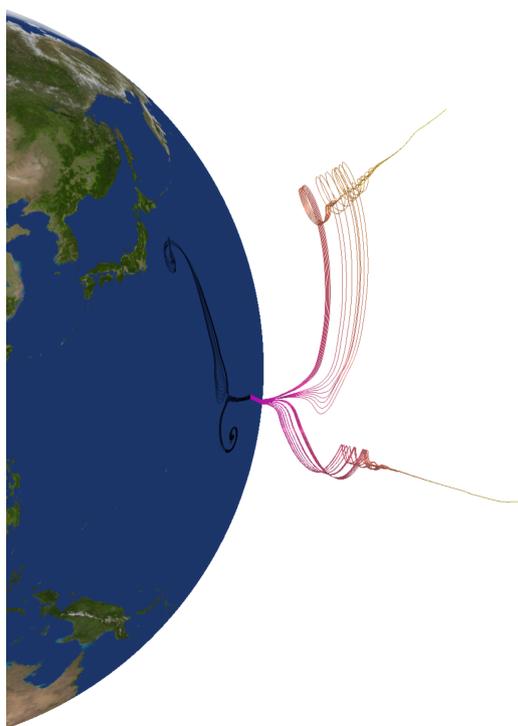
In contrast to the LIC algorithm, we compute the path lines in parallel on the CPU and perform the illumination in parallel on the GPU. Our implementation allows the user to interactively seed multiple path lines in a regular grid around the mouse cursor.

### 5.4.3 Results

Figure 5.4 compares three different visualization approaches of path lines. The result of the intrinsic path line integration is shown in Figure 5.4a. Since we perform all computations in 2D in the charts, the path lines lie on the surface of the sphere. By lifting them, we can observe features of the path lines we could not see before (shown in Figure 5.4b). For example, some path lines are swirling around some other path lines that are not moving. To better visualize the correspondence between the path lines in the 3D space and the surface of the sphere, shadows are drawn (shown in Figure 5.4c). Figure 5.5a is an example of how the illumination of path lines can improve spatial perception. Because of the specular highlights, we can estimate the curvature of the lines in 3D space. Figure 5.5b shows a close view of one of our animation modes (the particle like animation). Our video [Tro19] shows all animation modes in more detail. In Figure 5.6 we show our results for another data set. We colour coded the age of the particles on the path lines. In pink areas, the particles of the path line are very young since they were just seeded, while in yellow areas they are old.

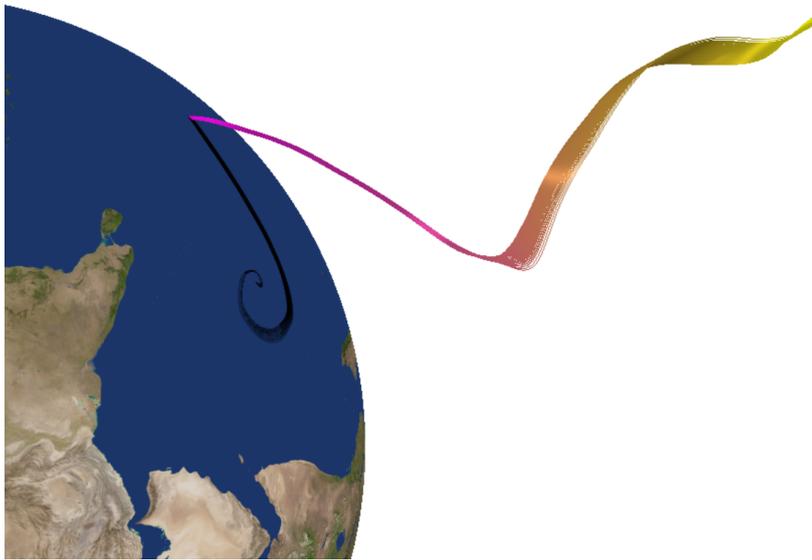


(a) Path lines on the surface of the sphere      (b) Lifted in 3D ambient space

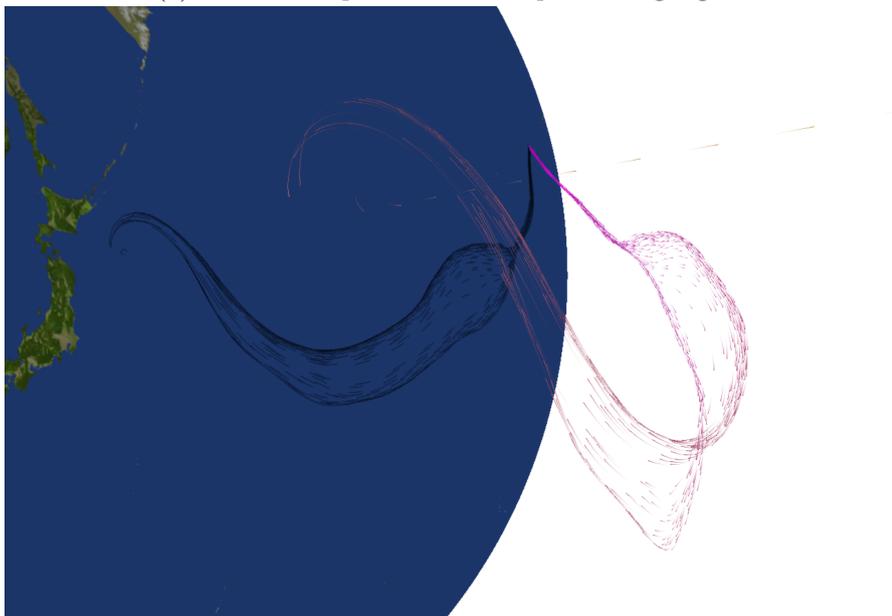


(c) With illustrative shadows

Figure 5.4: Comparison between visualizing path lines on the surface of the sphere and lifting them in the surrounding 3D space.



(a) Illuminated path lines with specular highlights



(b) Particle like animation of path lines

Figure 5.5: Examples for a specular highlight on a path line and one of our three animation modes

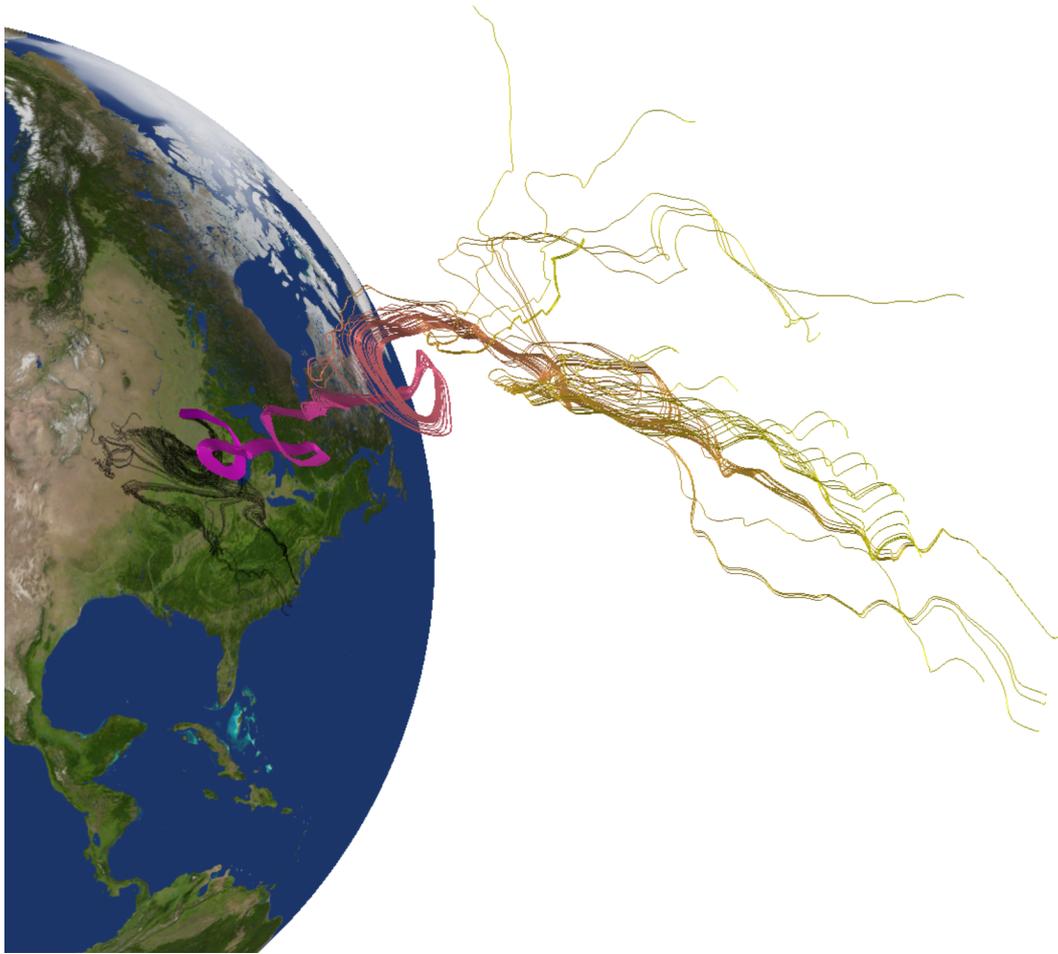


Figure 5.6: Path lines of wind data set provided by the Atmospheric and Climate Modeling research group at King Abdullah University of Science & Technology

# Conclusion

We presented an entirely intrinsic framework for interactive flow visualization on curved manifolds. After a detailed introduction to the relevant topics of differential geometry, we described two flow visualization approaches on curved surfaces. While line integral convolution provides an overview of a flow field, path lines let us gain more detailed insight into a flow data set. We used these algorithms to visualize data from climate simulations.

## 6.1 Limitations

There are two significant limitations of our intrinsic approach. Firstly, we are limited to surfaces that have an analytical description. Currently, we define the metric tensor for measuring lengths. For arbitrarily complex surfaces, we would need to convert the vectors to their 3D (extrinsic) representation. The second disadvantage is that we currently only visualize one layer of flow data. Our approach could be extended to visualize flow data on multiple layers of different altitude, like the wind at different heights, by altering the radius of the concentric sphere or by intersecting different layers of flow data.

## 6.2 Future Work

Future work could extend the intrinsic approach to more complex parametric surfaces than the sphere, e.g., the torus. All algorithms we describe could be adapted to other parametric surfaces or triangle meshes in general. Also, other visualization techniques like streak lines, time lines and streak surfaces could be extended to curved surfaces.



# List of Figures

3.1	A manifold $M$ with its co-ordinate maps $f_1$ and $f_2$ . $U_1$ and $U_2$ are subsets of $M$ . $f_{12}$ and $f_{21}$ are chart transition maps. <b>Source:</b> [Jon19] . . . . .	6
3.2	A curve on a manifold. $a$ and $b$ are the boundaries of the open set in $\mathbb{R}$ . <b>Source:</b> [Sch80] . . . . .	8
4.1	The different regions of the sphere covered by charts. Six charts align symmetrically around the sphere as planes in $\mathbb{R}^2$ . All adjacent charts are overlapping.	14
4.2	Each curve is used to define one basis vector. The tangent vector space of these basis vectors is located on the point where both curves intersect. . . . .	15
5.1	The red curve shows the Euler approximation of the blue curve. A smaller distance between the sample points $s_i$ results in a more precise representation of the blue curve. <b>Source:</b> [Tak19] . . . . .	18
5.2	Two different time steps of our periodic filter function $h(u, t)$ . Since we make $n/2$ forward and $n/2$ backward Euler steps, the blue vertical line represents the position of the texel on the stream line. . . . .	20
5.3	LIC visualization of the wind data set of the DKRZ . . . . .	21
5.4	Comparison between visualizing path lines on the surface of the sphere and lifting them in the surrounding 3D space. . . . .	24
5.5	Examples for a specular highlight on a path line and one of our three animation modes . . . . .	25
5.6	Path lines of wind data set provided by the Atmospheric and Climate Modeling research group at King Abdullah University of Science & Technology . . . . .	26



# Bibliography

- [Car92] Manfredo Perdigao do Carmo. *Riemannian geometry*. Birkhäuser, 1992.
- [CL93] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. *Lawrence Livermore National Lab., CA (United States)*, 1993.
- [Had19] Markus Hadwiger. *Scientific Visualization, Lecture 22: Flow Visualization, Pt. 3*, 2019 (accessed May 07, 2019). [https://faculty.kaust.edu.sa/sites/markushadwiger/Documents/CS247\\_spring2019\\_lecture\\_22.pdf](https://faculty.kaust.edu.sa/sites/markushadwiger/Documents/CS247_spring2019_lecture_22.pdf).
- [JL97] Bruno Jobard and Wilfrid Lefer. Creating evenly-spaced streamlines of arbitrary density. In *Visualization in Scientific Computing'97*, pages 43–55. Springer, 1997.
- [Jon19] Mattias Jonsson. *Non-Archimedean Geometry*, 2013 (accessed April 12, 2019). <http://www.maths.usyd.edu.au/u/austms2013/talks/aust13Jonsson.pdf>.
- [LJH03] Robert S Laramée, Bruno Jobard, and Helwig Hauser. Image space based visualization of unsteady flow on surfaces. In *IEEE Visualization, 2003.*, pages 131–138. IEEE, 2003.
- [LL19] Almut Gassmann Leonidas Linardakis, Daniel Reinert. *ICON Grid Documentation*, 2011 (accessed April 25, 2019). [https://www.dkrz.de/SciVis/hd-cp-2/en-icon\\_grid.pdf?lang=en](https://www.dkrz.de/SciVis/hd-cp-2/en-icon_grid.pdf?lang=en).
- [MKFI97] Xiaoyang Mao, Makoto Kikukawa, Noboru Fujita, and Atsumi Imamiya. Line integral convolution for 3d surfaces. In *Visualization in Scientific Computing'97*, pages 57–69. Springer, 1997.
- [MPSS05] Ovidio Mallo, Ronald Peikert, Christian Sigg, and Filip Sadlo. Illuminated lines revisited. In *IEEE Visualization, 2005.*, pages 19–26. IEEE, 2005.
- [Rea19] Simon Rea. *Lecture Notes of Simon Rea: Geometric Anatomy of Theoretical Physics*, 2017 (accessed April 17, 2019). <https://drive.google.com/file/d/1nchF1fRGSY3R3rP1QmjUg7fe28tAS428/view>.

- [Sch80] Bernard F Schutz. *Geometrical methods of mathematical physics*. Cambridge university press, 1980.
- [Sch19] Frederic Schuller. *Lectures on Geometrical Anatomy of Theoretical Physics*, 2016 (accessed May 09, 2019). [https://www.youtube.com/playlist?list=PLPH7f\\_7ZlzxTi6kS4vCmv4ZKm9u8g5yic](https://www.youtube.com/playlist?list=PLPH7f_7ZlzxTi6kS4vCmv4ZKm9u8g5yic).
- [SH97] Detlev Stalling and Hans-Christian Hege. *LIC on surfaces. Texture Synthesis with Line Integral Convolution*, 1997.
- [Sol15] Justin Solomon. *Numerical algorithms: methods for computer vision, machine learning, and graphics*, pages 311 – 315. CRC Press, 2015.
- [Tak19] Kenshi Takayama. *Introduction to Computer Graphics*, 2015 (accessed April 30, 2019). <http://research.nii.ac.jp/~takayama/teaching/utokyo-iscg-2017/slides/iscg-2017-06-animation2-en.pdf>.
- [TB96] Greg Turk and David Banks. Image-guided streamline placement. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 453–460. ACM, 1996.
- [Tro19] Jakob Troidl. *Flow Visualization on curved Manifolds*, 2019 (accessed May 26, 2019). [https://youtu.be/5B0BHSpi\\_6g](https://youtu.be/5B0BHSpi_6g).
- [Wij03] Jarke J van Wijk. Image based flow visualization for curved surfaces. In *Proceedings of the 14th IEEE Visualization 2003*, page 17. IEEE Computer Society, 2003.
- [ZSH96] Malte Zockler, Detlev Stalling, and HC Hege. Interactive visualization of 3D-vector fields using illuminated stream lines. In *Proceedings of Seventh Annual IEEE Visualization'96*, pages 107–113. IEEE, 1996.