

ספריות הטכניון *The Technion Libraries*

בית הספר ללימודים מוסמכים ע"ש ארוין וג'ואן ג'ייקובס
Irwin and Joan Jacobs Graduate School

©
All rights reserved to the author

This work, in whole or in part, may not be copied (in any media), printed, translated, stored in a retrieval system, transmitted via the internet or other electronic means, except for "fair use" of brief quotations for academic instruction, criticism, or research purposes only. Commercial use of this material is completely prohibited.

©
כל הזכויות שמורות למחבר/ת

אין להעתיק (במדיה כלשהי), להדפיס, לתרגם, לאחסן במאגר מידע, להפיצו באינטרנט, חיבור זה או כל חלק ממנו, למעט "שימוש הוגן" בקטעים קצרים מן החיבור למטרות לימוד, הוראה, ביקורת או מחקר. שימוש מסחרי בחומר הכלול בחיבור זה אסור בהחלט.

Functional Tracing of Discrete Vector Fields

Yair Reani

Functional Tracing of Discrete Vector Fields

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Yair Reani

Submitted to the Senate
of the Technion — Israel Institute of Technology
Shvat 5780 Haifa February 2020

This research was carried out under the supervision of Prof. Mirela Ben-Chen, in the Faculty of Computer Science.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

Abstract	1
-----------------	----------

List of Key Symbols	3
----------------------------	----------

1 Introduction	5
-----------------------	----------

1.1 Contributions	6
1.2 Overview	6

2 Related Work	7
-----------------------	----------

3 Background	9
---------------------	----------

3.1 Smooth Setup	9
3.1.1 Flowlines	9
3.1.2 The covariant derivative operator	10
3.2 Discrete Setup	11
3.3 Useful Definitions	11
3.4 The discrete covariant derivative operator	12

4 Flowlines Calculation	13
--------------------------------	-----------

4.1 Approach	13
4.1.1 Evolution of a function in time	13
4.1.2 Evolution the coordinate functions in time	14
4.2 The discrete case	14
4.3 Invariance to Transformations	16

5 Implementation Details	17
---------------------------------	-----------

5.1 Choosing the parameters	17
5.1.1 Normalization factor	17
5.1.2 The time of flow	18
5.2 The number of samples	18

6 Results	21
6.1 Comparison to Other Solutions	21
6.1.1 Analytical solutions	22
6.1.2 Tracing	23
6.1.3 Projected flowlines	24
6.2 Local Behaviour	26
6.2.1 Locality	26
6.2.2 Singularities	27
6.3 Application: Vector Field Visualization	28
6.3.1 Timings	28
7 Summary	33
7.1 Limitations	33
7.2 Conclusions	33
7.3 Future Work	33

List of Figures

3.1	Illustration of a flowline (the dashed line) of a tangent vector field v , here denoted by $\varphi_{v,\tau}(p)$ where $\tau \in [0, t]$. The tangent vector field represents a velocity field, and the blue point is a particle that started at p at $t = 0$. Figure reproduced with permission from [BCA19].	10
3.2	Tangent vector fields on triangle meshes.	11
3.3	Illustration of the operation of $D_v = I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \text{grad}$ on a function f . (a) A vertex based function f and a vector field v are given. (b) $\text{grad } f$ (red). (c) $[v]_{\bullet}^T \text{grad } f$. (d) $D_v f = I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \text{grad } f$	12
5.1	flowlines on a cat for different time values given in the characteristic length units. See the definition of the characteristic length r_M of a mesh in Section 5.1.2.	18
5.2	Illustration of r_M on several meshes. The lines on each mesh M are of length r_M . For the lines generation we used [Vax19].	19
6.1	Two cases for which the analytical solution for the vector field flow is known: the vector fields $\beta \nabla Z$ (left) and $\beta J \nabla Z$ (right) on the unit sphere, where $\beta \in \mathbb{R}$	22
6.2	Comparing to analytic: The results for the vector fields $\beta \text{grad } Z$ (top) and $\beta J \text{grad } Z$ (bottom) for several refinements, where β is the usual normalization, and $t = 0.2r_M$. The flowline error was measured by the maximum sample error.	23
6.3	Qualitative comparison of our method with field tracing. While the results are visually similar, our approach leads to flowlines which might not lie on the input shape.	24
6.4	Comparison to tracing, for different levels of refinement of a Teddy mesh, with $t = 0.2r_M$	25
6.5	Distance from the surface for 6 cylinders with a symmetry as the z axis. The vector field is $v = J \text{grad } Z$ for all refinements, and $t = 1r_M$ where $r_M = 0.5$ is the radius of the cylinders. The flowline error was measured by the distance from the end point to the mesh.	25
6.6	Distance from the surface for three levels of refinement of a teddy. The vector field $v = \beta(\text{grad } Z + J \text{grad } Y)$ where β is the same for all refinements, and set such that the vector field of the finest refinement satisfies Eq. (5.3). $t = 0.5r_M$ and The flowline error was measured by the maximum sample error.	26

6.7	For each mesh, we measure the locality error for 200 vertices. <i>Max. error</i> is the maximal error observed at a vertex. Note that the difference between the two flowlines at each vertex is small compared to the characteristic length.	28
6.8	Measuring the locality of a flowline calculation in single vertices.	29
6.9	Behaviour around different kinds of singular points.	29
6.10	Visualisation of a vector field. The direction represented as flowlines, and the norm represented as a color on the faces.	30
6.11	A vector field visualization on a non-regular mesh. It seems that our method produces more aesthetic results in regions which are more densely triangulated.	31
6.12	We see that the solution explodes in a small area of the mesh, due to a problematic triangulation in that region.	32
6.13	The calculation time as a function of the number of vertices for the shapes in SHREC07 with genus 0 and with no boundary for $t = 0.1r_M$	32

Abstract

We propose a method for approximating the flowlines of a discrete tangent vector field on a triangle mesh. Our method makes use of the recently proposed discrete representation of a vector field as a derivation operator. This representation allows us to state the problem of flowlines computation as the advection of the Euclidean coordinate functions by the vector field. By representing the vector field as a linear derivation operator, and discretizing both the vector field operator and the coordinate functions using Lagrange linear elements (or "hat functions"), the spatial discretization of the flowline equations leads to three linear systems of ordinary differential equations (ODEs), one system for each Euclidean coordinate function. These linear ODEs have a closed form solution as a function of time, thus the system can be solved without explicit time discretization, using an exponential integrator. Our approach requires only the construction of the derivative operator that represents the vector field, and multiplying the exponential of a sparse matrix by a vector, which can both be efficiently computed. For a given equally spaced time vector, we compute the flowlines from *all* the vertices of the mesh simultaneously. With this global definition of the problem, our method is characterized by making use of mostly global solutions, as opposed to algorithms that analyse local geometric details through the explicit generation of curves and intersecting line segments. We compare our approach to analytical solutions in cases where these are known, and to an iterative simple tracing algorithm. In addition we examine our solution from other aspects, such as invariance to global transformations, the distance of the flowlines from the mesh, and other local characteristics. Finally we use our method for the simple, robust and efficient visualization of discrete tangent vector fields on triangle meshes.

List of Key Symbols

Smooth Setup

\mathcal{M} : A smooth, orientable and compact 2-Manifold embedded in \mathbb{R}^3 .

$T\mathcal{M}$: The space of tangent vector fields on \mathcal{M} .

$\mathcal{X}_{\mathcal{M}} : \mathcal{M} \rightarrow \mathbb{R}^3$, The Euclidean embedding of \mathcal{M} .

$v : \mathcal{M} \rightarrow T\mathcal{M}$, A smooth tangent vector field on \mathcal{M} .

$f : \mathcal{M} \rightarrow \mathbb{R}$, A smooth real function on \mathcal{M} . That is $f \in C^\infty(\mathcal{M})$.

$\nabla : C^\infty(\mathcal{M}) \rightarrow T\mathcal{M}$, The gradient operator on \mathcal{M} .

$\nabla \cdot : T\mathcal{M} \rightarrow C^\infty(\mathcal{M})$, The divergence operator on \mathcal{M} .

$\Delta : C^\infty(\mathcal{M}) \rightarrow C^\infty(\mathcal{M})$, The Laplace-Beltrami operator.

$D_v : C^\infty(\mathcal{M}) \rightarrow C^\infty(\mathcal{M})$, The covariant derivative operator, of the vector field v .

$p \in \mathcal{M}$, A point on \mathcal{M} .

$\Phi_v^t(p) : A flowline of the vector field v from the point $p \in \mathcal{M}$ with time parameter $t \in \mathbb{R}$.$

$\Phi_v^t : \mathcal{M} \rightarrow \mathcal{M}$, The flowlines as a map for $t \in \mathbb{R}$.

$\mathcal{X}^t : \mathcal{M} \rightarrow \mathbb{R}^3$, The flow of the coordinate functions for $t \in \mathbb{R}$.

Discrete Setup

$M = (\mathcal{V}, \mathcal{F}, \mathcal{N})$. A triangle mesh, where \mathcal{V} is the set of vertices, \mathcal{F} is the set of triangular faces, and \mathcal{N} is the set of the normal vectors to the faces.

$|\mathcal{V}|$ The number of vertices.

$|\mathcal{F}|$ The number of faces.

$A_{\mathcal{F}} \in \mathbb{R}^{|\mathcal{F}|}$, The face areas.

$v \in \mathbb{R}^{3|\mathcal{F}|}$, A vector field, piecewise constant per face.

$X_M \in \mathbb{R}^{|\mathcal{V}| \times 3}$, The 3D Euclidean coordinates of the vertices.

$\text{grad} \in \mathbb{R}^{3|\mathcal{F}| \times |\mathcal{V}|}$, The discrete gradient operator.

$\text{div} \in \mathbb{R}^{|\mathcal{V}| \times 3|\mathcal{F}|}$, The discrete divergence operator.

$L \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, The discrete Laplace-Beltrami operator.

$\{\varphi_i\}_{i=1}^{|\mathcal{V}|} \in \mathbb{R}^{|\mathcal{V}|}$, The eigenfunctions of L , sorted in ascending order by eigenvalue.

$D_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, The discrete covariant derivative operator of the vector field v .

$I_{\mathcal{V}}^{\mathcal{F}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{F}|}$, An interpolation matrix from faces to vertices.

$X^t \in \mathbb{R}^{|\mathcal{V}| \times 3}$, The evolution of the coordinate functions for time $t \in \mathbb{R}^n$.

$[v]_{\bullet} \in \mathbb{R}^{3|\mathcal{F}| \times |\mathcal{F}|}$, A matrix that encodes the facewise multiplication of the vector field v with a face-based function.

$\|u\| : \mathbb{R}^3 \rightarrow \mathbb{R}$, The Euclidean norm of a 3D vector u .

$J \in \mathbb{R}^{3|\mathcal{F}| \times 3|\mathcal{F}|}$, A matrix that encodes the face-wise counterclockwise rotation of a vector field by $\pi/2$ around the faces normals.

$n \in \mathbb{N}$, The number of samples of a flowline.

$X_p \in \mathbb{R}^{n_p \times 3}$, samples of a flowline that originate at p and has n_p samples.

$p \in \mathcal{V}$, A vertex of M .

$E \in \mathbb{R}^{3|\mathcal{F}| \times |\mathcal{V}|}$, A matrix that encodes the linear combination of the rotated edge vectors using the opposite vertex coefficients

$G_{\mathcal{F}} \in \mathbb{R}^{3|\mathcal{F}| \times 3|\mathcal{F}|}$, A diagonal matrix that contains the areas of the faces, repeated 3 times for each face. Encodes the facewise multiplication of a vector field by the face areas.

$r_M \in \mathbb{R}$, The characteristic length that corresponds to M .

$A_M \in \mathbb{R}$, The area of M .

Chapter 1

Introduction

Tangent vector fields on surfaces play an important role in computer graphics and geometry processing. They are designed [FSDH07], visualized [ZHT06] and have many applications in areas such as physics modeling (e.g. fluid simulation [AWO⁺14]), and art (e.g. line-art rendering [HZ00]).

In the smooth setting, a tangent vector field on a 2-manifold is a smooth assignment of a tangent vector to each point of the manifold. In this context, a flowline is a curve with a parameter t such that at each point its velocity equals the vector field. Each point on the surface is associated with a flowline passing through it at $t = 0$. Flowlines calculation is useful for applications such as vector field visualization, segmentation, and quad meshing.

Another point of view, coming from differential geometry, is the representation of vector fields as operators. Specifically, the *covariant derivative operator* of a vector field associates with any smooth scalar function on the surface another function which is its directional derivative on the surface, by the vector field. This operator plays a role in the definition of flowlines on a 2-manifold embedded in a 3D Euclidean space through the evolution of the manifold's Euclidean coordinates functions.

As is common in many applications, our setup consists of a triangle mesh, and a piecewise constant tangent vector field as an input, that is an assignment of a Euclidean vector to each face of the mesh. In this setting the vector field is not smooth, and even not defined on edges and vertices. Thus, we cannot state the problem as finding a curves that equals the vector field at every point on the mesh. We present a way for an approximate calculation of flowlines that originate at each vertex of the mesh. We use a discrete representation of the covariant derivative operator that was recently suggested by [ABCCO13], and state the problem as a system of linear ODEs of order one, inspired by the definition of flowlines as the evolution of the mesh's Euclidean coordinates. This system has a closed form solution, given by the matrix exponential of the covariant derivative operator multiplied by the time parameter.

Most of the algorithms dealing with tracing lines along a vector field, use iterative processes that have to take care of local details such as intersections of lines with edges or vertices, and local corrections. Our algorithm, on the other hand, uses a global operator point of view, which

only requires the building the sparse covariant derivative matrix, and the calculation of the multiplication of a sparse matrix exponential operator with a vector.

1.1 Contributions

We present an algorithm that gets as input a vector field on a triangle mesh and a linearly sampled time interval, and computes the flowlines of the vector field, starting from all the vertices of the mesh, for the given time values. The algorithm requires only the construction of a discrete covariant derivative operator and the computation of a sparse matrix exponential times a vector. The flowlines are calculated directly from the mesh and vector field operator in closed form, without requiring a local analysis of geometric details or explicit calculations of line segment intersections. We demonstrate the applicability of our algorithm for visualizing vector fields on a mesh.

1.2 Overview

In Chapter 2 we review other papers that deal with integration of curves on a surface. In chapter 3 we present the smooth and the discrete setups, the definition of flowlines in the smooth setup, the definitions of the covariant derivative in both setups, and other useful definitions for the discrete case. In chapter 4 we present our global operator-based approach for flowlines calculation in the smooth case, and then we show how to apply this approach to the discrete case. We end this chapter with a closed form solution for the discrete case. In chapter 5 we explain how we set the free parameters of our method. In chapter 6 we evaluate the flowlines calculation by comparing it to other methods, and by examining its local behavior. Finally, we present a simple way to visualize vector field on a mesh using our method. In chapter 7 we discuss the limitations of our method, our conclusions, and ideas for future work.

Chapter 2

Related Work

Tracing flowlines. Iterative algorithms for calculating flowlines (or streamlines) were suggested in [CL93] (2D), [MKFI97] (3D) as a part of the Line Integral convolution (LIC) technique. LIC is a way of visualizing vector field by convolving white noise with a one dimensional filter along streamlines of the vector field.

Geodesic curves over triangle meshes. [MVC05] suggested an algorithm to compute the shortest geodesic curve between two points, over triangulated surfaces. The algorithm uses FMM (First Marching Method) for an initial curve, and then applies an iterative correction process in order to approach the true geodesic path. The correction process requires different procedures for the correction of the polygonal line's vertices. [PS06] introduced straightest geodesics on triangle meshes. They applied it to numerical integration methods for tangential vector fields. The concept of straightest geodesics deals with crossing edges and vertices. [SSK⁺05] presented methods for computing exact and approximate geodesics from a source point to one or all other points. The method uses as a first step the MMP algorithm which requires the partitioning of the mesh's edges to "windows" and the propagation of those windows across the faces of the mesh. It has to take special care of intersecting windows, intersections of "rays" and edges, and treat differently boundary points and saddle points.

Streamlines in Euclidean space. [RT11] proposed a new technique for visual exploration of streamlines of a 3D vector field. The setup is a vector field in a 2D or a 3D Euclidean space. The technique constructs a map from the space of streamlines to points in \mathbb{R}^n , while preserving the Hausdorff metric in streamline space, and cluster the points. Streamlines are integrated using fourth order Runge Kutta with adaptive step size control.

Quad layouts. [RRP15] introduced an approach for automatically computing pure quadrilateral patch layouts (quad layouts) on manifold meshes. The method constructs a singularity graph from the input, and calculates the quad layout by a constrained optimization. In the graph generation part it traces isolines of a parametrization induced from a curl-free frame field.

[PPM⁺16] Presented a method, that given a triangle mesh with a cross field, it computes a field aligned coarse quad layout. The method requires a construction of special graphs.

Polylines. [RS14] suggested an algorithm for tracing polylines that are oriented by a direction field over a triangle mesh. The algorithm builds a special data structure in order to compute how a line cross a triangle, and it requires a new direction field representation.

Chapter 3

Background

In this chapter, we first present the definition of flowlines and the definition of the covariant derivative operator in the smooth case. We need it in order to define flowlines from a global operator-based point of view. Then, we describe the discrete setup on a triangle mesh, and the definitions of the matrices that we use in the rest of the thesis. Finally, we recall the definition of the discrete covariant derivative operator in order to lay the groundwork for the discrete definition of the problem that we state in Chapter 4. Most of the content in this chapter is also available in [BCA19], and is repeated here for completeness. Note that we only describe the main topics that are of use for this thesis. For a more thorough overview of the mathematical aspects of derivations we refer the reader to [Mor01].

3.1 Smooth Setup

In the smooth setting we are given a compact smooth Riemannian manifold \mathcal{M} and a tangent vector field v , that is a smooth assignment of a tangent vector $v(p)$ to each point $p \in \mathcal{M}$. We work with functions in $C^\infty(\mathcal{M})$.

3.1.1 Flowlines

A smooth tangent vector field v defines a one-parameter family of curves which is the solution to the PDE [Mor01, Definition 1.43]

$$\begin{aligned} \frac{d\Phi_v^t(p)}{dt} &= v(\Phi_v^t(p)), \\ \Phi_v^{t=0}(p) &= p, \end{aligned} \tag{3.1}$$

for all $p \in \mathcal{M}$, and for $t \in \mathbb{R}$.

The *flowlines* of the vector field are this family of curves. Intuitively, if v describes the velocity of a fluid at every point on \mathcal{M} , then a single flowline $\Phi_v^t(p)$ is the trajectory of a particle passing at p at $t = 0$ and advected by the fluid (see Fig. 3.1).

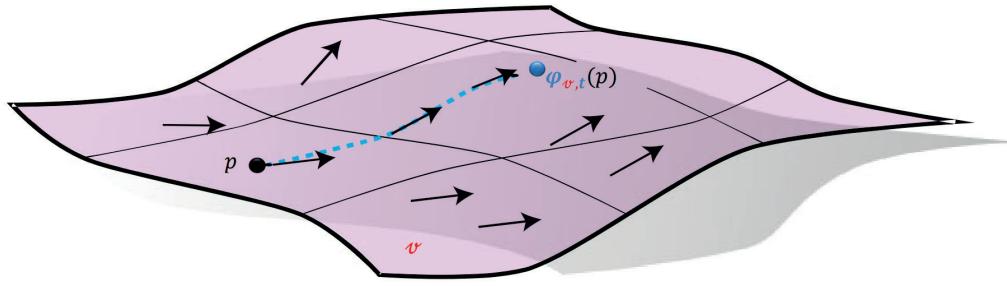


Figure 3.1: Illustration of a flowline (the dashed line) of a tangent vector field v , here denoted by $\varphi_{v,t}(p)$ where $\tau \in [0, t]$. The tangent vector field represents a velocity field, and the blue point is a particle that started at p at $t = 0$. Figure reproduced with permission from [BCA19].

3.1.2 The covariant derivative operator

Instead of the pointwise representation of vector fields our method makes use of the *global* operator-based representation of a vector field as a derivation operator acting on functions, specifically the one that was recently proposed in [ABCCO13]. As we show in Chapter 4 , this representation allows us to state the problem from a global point of view, as the evolution of functions in time. The covariant derivative is defined as follows.

Definition 3.1.1. Given a vector field v and a function $f \in C^\infty(\mathcal{M})$ the covariant derivative $\mathcal{D}_v : C^\infty(\mathcal{M}) \rightarrow C^\infty(\mathcal{M})$ of f with respect to v is a function $\mathcal{D}_v f : \mathcal{M} \rightarrow \mathbb{R}$ such that

$$(\mathcal{D}_v f)(\boldsymbol{\rho}) = \lim_{t \rightarrow 0} \frac{f(\Phi_v^t(\boldsymbol{\rho})) - f(\boldsymbol{\rho})}{t} = \frac{d}{dt} f(\Phi_v^t(\boldsymbol{\rho})) \Big|_{t=0}, \quad (3.2)$$

for any point $\boldsymbol{\rho} \in \mathcal{M}$.

The covariant derivative operator can be expressed through the gradient operator, using a classical identity in Riemannian geometry ([Mor01], p.148):

$$(\mathcal{D}_v f)_p = \langle v(p), \nabla f(p) \rangle_p. \quad (3.3)$$

This operator has a myriad of nice properties (e.g. linearity in the functions, linearity in the vector field), which were leveraged in [ABCCO13] to represent discrete tangent vector fields on triangle meshes.

3.2 Discrete Setup

In the discrete setup we are given a triangle mesh $M = (\mathcal{V}, \mathcal{F}, \mathcal{N})$ where \mathcal{V} is a set of vertices, \mathcal{F} is a set of triangular faces, and \mathcal{N} is a set of the normal vectors to the faces. The mesh is endowed with a piece-wise constant tangent vector field $v = \{v_r \in \mathbb{R}^3 | r \in \mathcal{F}, v_r \perp \mathcal{N}_r\}$ (see Fig. 3.2). In practice we represent the vector field as a vector in $\mathbb{R}^{3|\mathcal{F}|}$. For the flowlines calculation we work with piecewise linear vertex-based functions represented as a vector in $\mathbb{R}^{|\mathcal{V}|}$. For the representation of the vector field norm in Section 6.3, we use piecewise constant face-based functions represented as a vector in $\mathbb{R}^{|\mathcal{F}|}$.

3.3 Useful Definitions

$[v]_{\bullet} \in \mathbb{R}^{3|\mathcal{F}| \times |\mathcal{F}|}$, A matrix that encodes a point-wise multiplication of a vector field v with a face-based function. Its transpose encodes an inner product of the vector field with another vector field.

$\text{grad} \in \mathbb{R}^{3|\mathcal{F}| \times |\mathcal{V}|}$, $\text{div} \in \mathbb{R}^{|\mathcal{V}| \times 3|\mathcal{F}|}$, The discrete gradient and divergence respectively, defined in the standard way as in [BKP⁺10, Chapter 3].

$I_{\mathcal{V}}^{\mathcal{F}} \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{V}|}$, An interpolation matrix from face-based functions to vertex-based functions defined as $I_{\mathcal{F}}^{\mathcal{V}}(i, j) = \frac{A_{\mathcal{F}}(j)}{3A_{\mathcal{V}}(i)}$, where $A_{\mathcal{V}}(i) = \frac{1}{3} \sum_{r \ni i} A_{\mathcal{F}}(r)$.

$J \in \mathbb{R}^{3|\mathcal{F}| \times 3|\mathcal{F}|}$, A vector field rotation operator. On each triangle, rotates the vector by $\pi/2$ around the normal.

$E \in \mathbb{R}^{3|\mathcal{F}| \times |\mathcal{V}|}$, Encodes a weighted average of the rotated edges, with a vertex-based function. Specifically, on each face the output vector is given by $\sum_{i=1}^3 f(p_i) e_i^{\perp}$, where p_1, p_2, p_3 are the vertices of the face and e_i^{\perp} is the edge opposite to the vertex p_i rotated around the face normal by $\pi/2$.

$G_{\mathcal{F}} \in \mathbb{R}^{3|\mathcal{F}| \times 3|\mathcal{F}|}$, Encodes a face-wise multiplication of a vector field by the face areas.

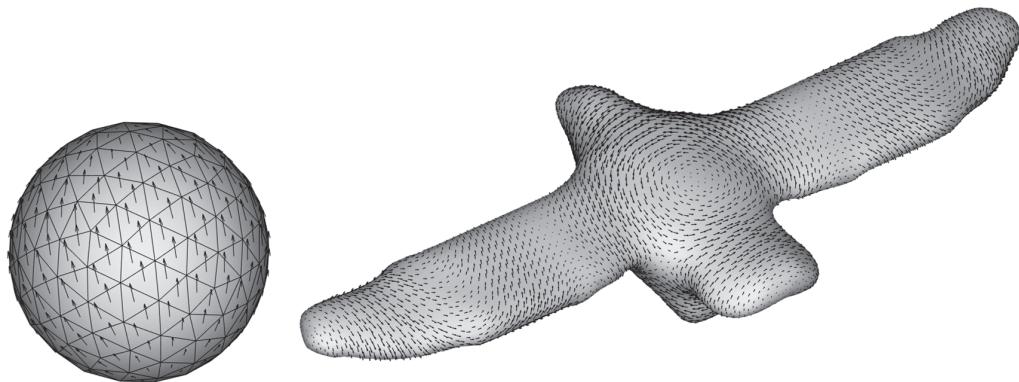


Figure 3.2: Tangent vector fields on triangle meshes.

3.4 The discrete covariant derivative operator

Given a discrete vector field v , we define the directional derivative operator $D_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, by discretizing Eq. (3.3) using linear finite elements [ABCCO13]:

$$D_v = I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \text{grad}. \quad (3.4)$$

Intuitively, this operator first computes the gradient of a vertex based function using the operator grad , yielding a vector field represented as a vector per face. The operator $[v]_{\bullet}^T$ then computes the facewise inner product between the gradient vector field and the given vector field v . This yields a piecewise constant function per face, which is then interpolated back to the vertices with the interpolation operator $I_{\mathcal{V}}^{\mathcal{F}}$. Note that in the discrete case, the interpolation is required in order to obtain a square operator, whereas in the smooth case the functions live in the same function space before and after applying the derivative operator (see Figure 3.3).

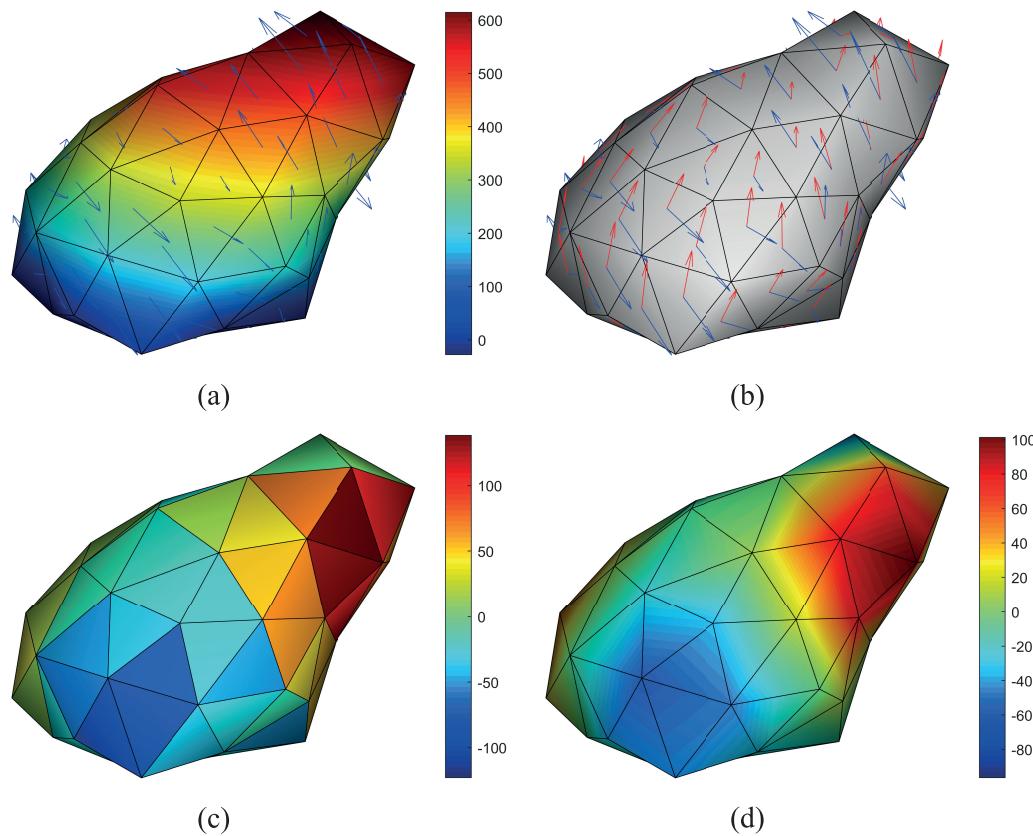


Figure 3.3: Illustration of the operation of $D_v = I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \text{grad}$ on a function f . (a) A vertex based function f and a vector field v are given. (b) $\text{grad } f$ (red). (c) $[v]_{\bullet}^T \text{grad } f$. (d) $D_v f = I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \text{grad } f$.

Chapter 4

Flowlines Calculation

In this chapter we present our approach to flowlines computation. Definition 3.1 defines flowlines from a local point of view. That is, it gives the local connection between a curve and the vector field *at a given point on the manifold*. If we consider \mathcal{M} as embedded in \mathbb{R}^3 , we can assign each point on \mathcal{M} with its 3D Euclidean coordinates by the embedding function $\mathcal{X}_{\mathcal{M}} : \mathcal{M} \rightarrow \mathbb{R}^3$. Then, using the covariant derivative operator, we can define flowlines from a global point of view as the evolution of $\mathcal{X}_{\mathcal{M}}$ in time.

4.1 Approach

4.1.1 Evolution of a function in time

For a given time t , we can think about the flowlines as a map $\Phi_v^t : \mathcal{M} \rightarrow \mathcal{M}$. Given a real function $f \in C^\infty(\mathcal{M})$, we define the function $f^t : \mathcal{M} \rightarrow \mathbb{R}$ as

$$f^t = f \circ \Phi_v^t, \quad (4.1)$$

where $f^0 = f$.

We want to understand how the function $f^t = f \circ \Phi_v^t$ evolves in time. The following result can be derived from Lemma 5 in Ref. [BCA19], using the connection $f^t = g(-t)$, where $g(t)$ is the function $f(t)$ used there:

$$\frac{d}{dt} f^t(p) = (\mathcal{D}_v f^t)(p). \quad (4.2)$$

The last equality holds for all the points on the manifold, thus

$$\begin{aligned} \frac{d}{dt} f^t &= \mathcal{D}_v f^t \\ f^{t=0} &= f. \end{aligned} \quad (4.3)$$

4.1.2 Evolution the coordinate functions in time

Our surface is embedded in \mathbb{R}^3 , so that we can describe a point $p \in \mathcal{M}$ by its 3D embedding:

$$\mathcal{X}_{\mathcal{M}}(p) = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})(p), \quad (4.4)$$

where $\mathcal{X}, \mathcal{Y}, \mathcal{Z} : \mathcal{M} \rightarrow \mathbb{R}^3$ are the Euclidean coordinate functions. By the flow of each of the coordinate functions to the time t we will get a map $\mathcal{X}^t : \mathcal{M} \rightarrow \mathbb{R}^3$ such that

$$\mathcal{X}^t = \mathcal{X}_{\mathcal{M}} \circ \Phi_v^t. \quad (4.5)$$

By the definition of \mathcal{X} , $\mathcal{X}^t(p)$ are the coordinates of the point $\Phi_v^t(p)$.

Applying the last connection to each of the coordinate functions we get

$$\begin{aligned} \frac{d}{dt} \mathcal{X}^t &= \mathcal{D}_v \mathcal{X}^t \\ \mathcal{X}^{t=0} &= \mathcal{X}_{\mathcal{M}}, \end{aligned} \quad (4.6)$$

where $\mathcal{D}_v \mathcal{X}^t = (\mathcal{D}_v \mathcal{X}^t, \mathcal{D}_v \mathcal{Y}^t, \mathcal{D}_v \mathcal{Z}^t)$.

4.2 The discrete case

In the discrete case the flow of the coordinates at time t is represented a matrix $X^t \in \mathbb{R}^{|\mathcal{V}| \times 3}$. In the discrete case, Eq. (4.6) is given by:

$$\begin{aligned} \frac{d}{dt} X^t &= D_v X^t \\ X^{t=0} &= X_M. \end{aligned} \quad (4.7)$$

Eq. (4.7) is a system of first order linear ODEs. It has a closed form solution, given by [HO10]:

$$X^t = \exp(tD_v)X_M = \sum_{k=0}^{\infty} \frac{(tD_v)^k X_M}{k!}, \quad (4.8)$$

where $\exp(\cdot)$ is the matrix exponent.

In practice we used the implementation in "<https://github.com/higham/expmv>" that is based on [AMH11], which given a *sparse* matrix $A \in \mathbb{R}^{k \times k}$, a discrete span with uniform spacing $T \in \mathbb{R}^l$, and a vector $u \in \mathbb{R}^k$ calculates

$$u^t \equiv \exp(tA)u \quad (4.9)$$

for every $t \in T$, without the costly operation of building the full matrix $\exp(tA)$.

We presented the method given a mesh, a vector field, and a time vector. In the next chapter we will explore the different parameters of our approach. We will give a way to set them in order

to achieve aesthetic visual results, and to achieve a uniform setting that allows us to compare between results on different meshes.

4.3 Invariance to Transformations

The smooth equation that characterizes the flowlines is *intrinsic*, i.e. independent of the mesh embedding. However, our definition of the flowlines makes use of the embedding, and is thus *extrinsic*. In this section we examine the effect of the embedding of M on the flowlines calculations. We show analytically the invariance of the flowlines calculation to global translation, rotation and scaling of the 3D Euclidean coordinates of the mesh, when the vector field is transformed correspondingly. These transformations do not change the *ratios* between any pair of edges lengths of the mesh. By invariance we refer to the commutativity of the flowlines calculation and these transformations.

Theorem 4.1. *Given a mesh $M = (\mathcal{V}, \mathcal{F})$ with embedding $X_M \in \mathbb{R}^{|\mathcal{V}| \times 3}$ and a tangent vector field v , consider a global similarity transformation, given by a matrix $A = \alpha R$ and a translation vector $d \in \mathbb{R}^3$, where $\alpha \in \mathbb{R}_{>0}$ is the global scale and $R \in \mathbb{R}^{3 \times 3}$ is an orthogonal rotation matrix. Now, let $M_{A,d} = (\mathcal{V}_A, \mathcal{F})$ be the mesh with embedding $X_{A,d}(p) = X_M(p)A^T + d^T$, for any vertex $p \in \mathcal{V}$, and let v_A be the vector field given by $Av(r)$ for any face $r \in \mathcal{F}$. Then, we have that:*

$$(\exp(tD_{v_A})X_{A,d})(p) = ((\exp(tD_v)X_M)(p))A^T + d^T \quad \forall p \in \mathcal{V},$$

for any $t \in \mathbb{R}$.

Proof. By definition we have that $X_{A,d}(p) = X_M(p)A^T + d^T$, thus $X_{A,d} = X_M A^T + d_V$, where $d_V \in \mathbb{R}^{|\mathcal{V}| \times 3}$ repeats d^T on $|\mathcal{V}|$ rows. Hence, we need to show:

$$\exp(tD_{v_A})(X_M A^T + d_V) = \exp(tD_v)X_M A^T + d_V.$$

Let $f \in \mathbb{R}^{|\mathcal{V}| \times 1}$ be a constant function on the vertices of M . Since $\text{grad } f = 0$ we have that $D_v f = 0$ and thus also $D_v^k f = 0$, for any $k > 0$. Further, by definition we have that $\exp(tD_v)f = I_{|\mathcal{V}|}f + \sum_{k=1}^{\infty} \frac{1}{k!} D_v^k f = f$ for any constant function f . Since d_V has constant columns we have:

$$\exp(tD_{v_A})(X_M A^T + d_V) = \exp(tD_{v_A})X_M A^T + d_V.$$

Hence, we need to show that $D_{v_A} = D_v$, namely that D_v does not change under a global similarity transformation of the mesh, as long as the vector field v is modified accordingly. This invariance can be derived by extending Lemma 4.2 of [ABCCO13], as follows. \square

Lemma 4.3.1. *Let M, v, A, v_A be as in the previous Theorem. Then $D_v = D_{v_A}$.*

Proof. From Equation (3.4) we have that $D_v = I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \text{grad}$ and $D_{v_A} = I_{\mathcal{V}_A}^{\mathcal{F}}[v_A]_{\bullet}^T \text{grad}_A$. Since for a global similarity transformation the areas only change by a global scale, the interpolation from faces to vertices does not change and we have $I_{\mathcal{V}}^{\mathcal{F}} = I_{\mathcal{V}_A}^{\mathcal{F}}$. From Lemma 4.2 in [ABCCO13] we get that $[v]_{\bullet}^T \text{grad} = [v_A]_{\bullet}^T \text{grad}_A$, and thus $D_v = D_{v_A}$.

Chapter 5

Implementation Details

5.1 Choosing the parameters

Given a vector field v on a triangle mesh, our method has free parameters which are the time of the coordinates evolution t , and the number of samples on each flowline n . The length of the flowline depends on the vector field and on t , and the distribution of the lengths of the flowlines from all the vertices that is required in order to achieve clear visual results differs from one mesh to another. We wish to validate our approach by running our flowline computation algorithm on random meshes and random vector fields, thus need a method to automatically define the lengths by setting v and t .

Lemma 5.1.1. *Multiplying v by a scalar is equivalent to multiplying t by the same scalar.*

Proof. Given a scalar α , from the linearity of D_v in the vector field we get

$$D_{\alpha v} = \alpha D_v. \quad (5.1)$$

Thus,

$$\exp(tD_{\alpha v}) = \exp((\alpha t)D_v). \quad (5.2)$$

Based on the last Lemma, instead of explaining how to set t we can give a clearer explanation of how to set t and β , where $\beta \in \mathbb{R}$ is a normalization factor for v .

5.1.1 Normalization factor

Here we denote by v the given vector field multiplied by the normalization factor β . Usually we normalize the vector field v such that the average norm over M is 1. That is, such that the following equality holds

$$\frac{1}{A_M} \int_M \|v\| ds = \frac{1}{A_M} \sum_{r \in \mathcal{F}} \|v(r)\| A_{\mathcal{F}}(r) = 1, \quad (5.3)$$

where $A_M = \sum_{r \in \mathcal{F}} A_{\mathcal{F}}(r)$ is the area of the mesh, and $A_{\mathcal{F}}(r)$ is the area of a single triangle $r \in \mathcal{F}$.

The intuition comes from the fact that for a unit velocity field length the path of each particle is proportional to t .

5.1.2 The time of flow

Keeping in mind that t is the time we flow the coordinates for, we understand that the larger the value of t the longer the flowlines. When t is too small we will get flowlines that are hard to notice, and when t is too large it will be hard for us to distinguish between flowlines and it will increase calculation errors as illustrated in Fig. 5.1.

Following the logic of our vector field's normalization, we set $0.1r_M \leq t \leq 0.5r_M$, where r_M depends on the specific mesh and equals the radius of a 3D sphere with the same area:

$$r_M = \sqrt{\frac{A_M}{4\pi}}. \quad (5.4)$$

Illustration of r_M on several meshes is shown in figure 5.2.

5.2 The number of samples

We set n to be dense enough such the the flowlines look smooth to an observer. From our experience, setting n such that $nr_M/t \approx 300$ is more than enough.

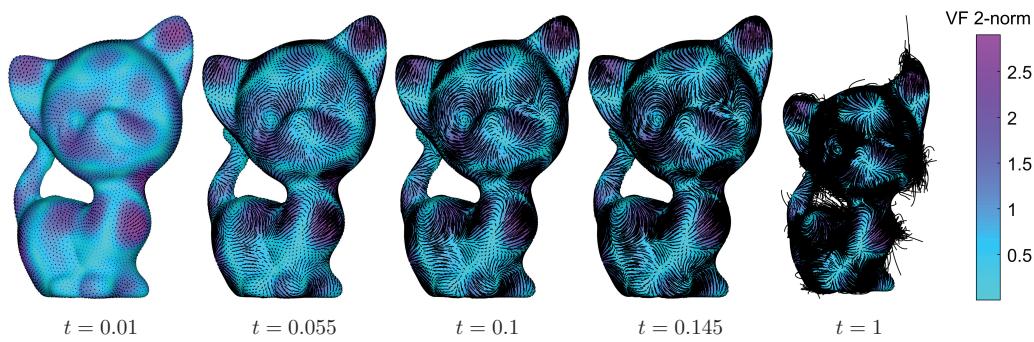


Figure 5.1: flowlines on a cat for different time values given in the characteristic length units. See the definition of the characteristic length r_M of a mesh in Section 5.1.2.

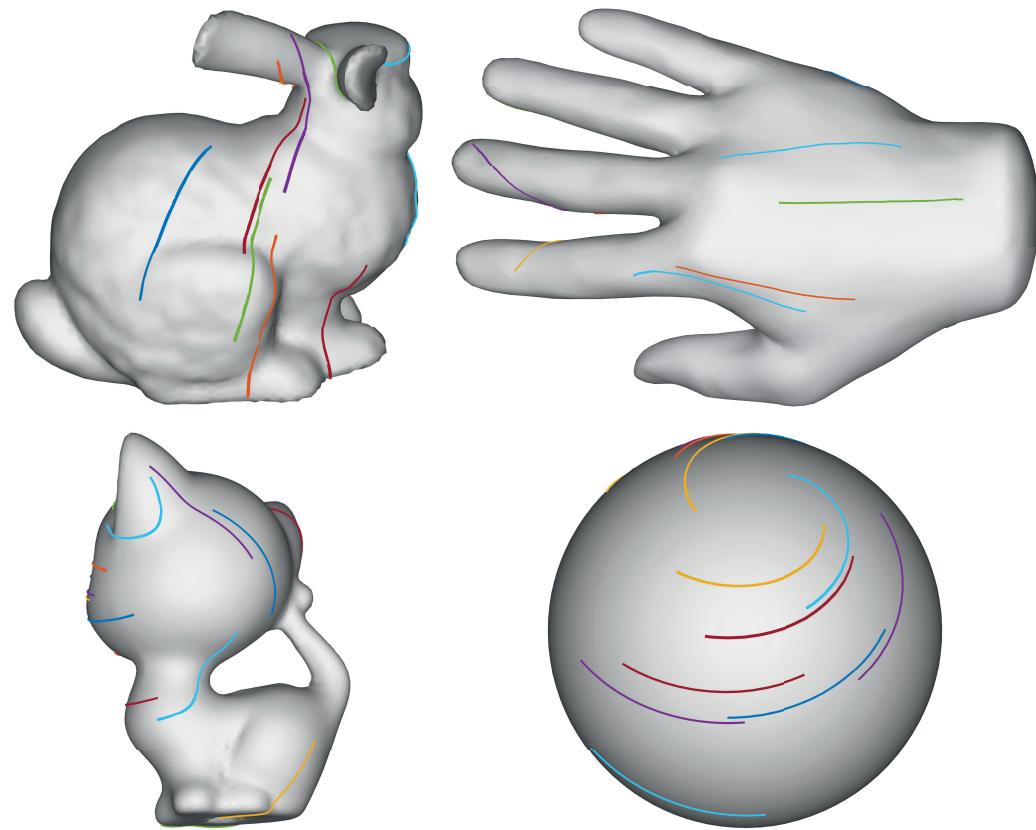


Figure 5.2: Illustration of r_M on several meshes. The lines on each mesh M are of length r_M . For the lines generation we used [Vax19].

Chapter 6

Results

This chapter includes measurements of our flowlines calculation. We examine the flowlines calculation from several different aspects. In order to simplify the measurements, we treat each flowline as a series of points. Comparing points is straightforward by using the Euclidean norm in 3D.

In section 6.1 we first explain in general how we compare between our method and alternative approaches to flowlines calculation. Afterwards we show results of comparisons to analytical solutions, to a tracing algorithm, and to the projection of our flowlines on the mesh. In Section 6.2 we measure the locality of our flowlines calculation, and show the behaviour near singularity points of the vector field. In section 6.3 we visualize vector fields on triangle meshes using our method.

6.1 Comparison to Other Solutions

In this section we compare our method to other approaches of flowlines calculation. First we explain in general how we compare our method to another given method.

For a set of origins on the mesh, we calculate the flowlines of our method and the flowlines of the other method such that for each origin p we calculate two flowlines $X_p \in \mathbb{R}^{n_p \times 3}$ and $\tilde{X}_p \in \mathbb{R}^{n_p \times 3}$, calculated by our method and the other method respectively. Each flowline has n_p samples, and each row of it contains the 3D Euclidean coordinates of the sample. The error of a single flowline calculated by our method is discussed in the following.

For a given flowline originating at p , we define the corresponding error vector $e_p \in \mathbb{R}^{n_p}$ by

$$e_p(i) = \frac{\|X_p^i - \tilde{X}_p^i\|}{r_M}, \quad (6.1)$$

where X_p^i (\tilde{X}_p^i) is the i -th row of X_p (\tilde{X}_p), that is the i -th sample. Now that we have the error of a single flowline we define $E(e_{th})$ for a given error threshold e_{th} as

$$E(e_{th}) = \frac{\sum_{p \in \mathcal{V}} I(\epsilon(e_p) < e_{th})}{|\mathcal{V}|} \cdot 100, \quad (6.2)$$

where $I(\cdot)$ equals 1 when the condition inside the parenthesis is satisfied and 0 otherwise, and $\epsilon(e_p)$ is a scalar that depends on the specific experiment. Finally, we present the results as a plot of E vs. e_{th} .

In this manner we compare our flowlines to three different ways of calculating flowlines: the analytical solution on the sphere, tracing and projected flowlines.

6.1.1 Analytical solutions

In cases where an analytical solution is known, we can compare the points of our flowlines to the points calculated analytically. We examine our method on two cases for which we know the analytical solution: the vector fields $\beta\nabla Z$, and $\beta J\nabla Z$ on the unit sphere, where Z is the z coordinate of the 3D embedding of the unit sphere, J is the rotation of a tangent vector field by $\pi/2$ around the normal at a point, and $\beta \in \mathbb{R}$ (see Fig. 6.1).

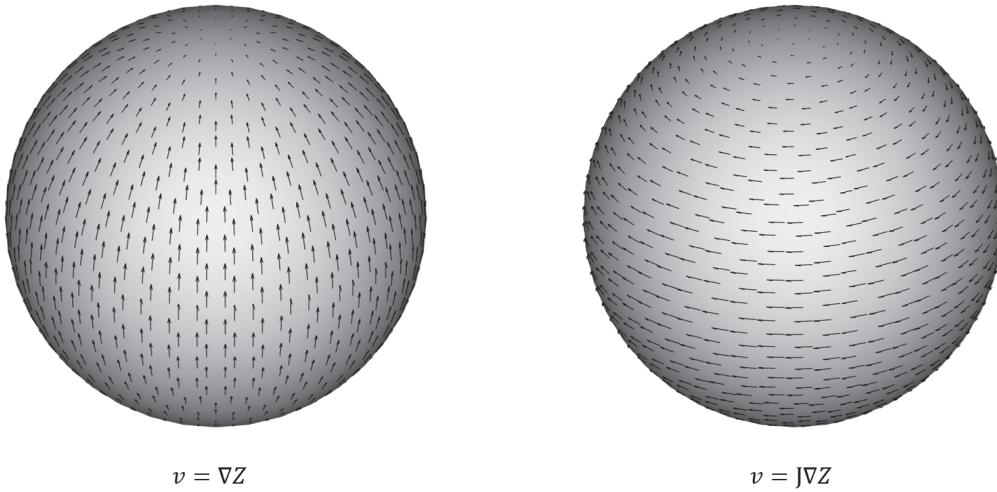


Figure 6.1: Two cases for which the analytical solution for the vector field flow is known: the vector fields $\beta\nabla Z$ (left) and $\beta J\nabla Z$ (right) on the unit sphere, where $\beta \in \mathbb{R}$.

When the sphere is centered at the origin, the two vector fields are given by the polar coordinates

$$\begin{aligned}\beta\nabla Z(\rho(\theta, \phi)) &= \beta \sin \theta(-\cos \theta \cos \phi \hat{x} - \cos \theta \sin \phi \hat{y} + \sin \theta \hat{z}), \\ \beta J\nabla Z(\rho(\theta, \phi)) &= \beta \sin \theta(\sin \phi \hat{x} - \cos \phi \hat{y}),\end{aligned}$$

where we denote by $\rho(\theta, \phi)$ the point on the sphere that corresponds to the spherical coordinates $\theta \in [0, \pi], \phi \in [0, 2\pi]$. The solutions of Eq. (3.1) in this case are

$$\begin{aligned}\Phi_{\beta\nabla Z}^t(\rho(\theta, \phi)) &= \rho(2 \cot^{-1}(\cot(\frac{\theta}{2}) \exp(\beta t)), \phi) \\ \Phi_{\beta J\nabla Z}^t(\rho(\theta, \phi)) &= \rho(\theta, \phi - \beta t).\end{aligned}$$

For each case we set β such that the vector field satisfies Eq. (5.3), and calculate a flowline for each vertex p of the mesh, with $t = 0.2r_M$ (here $r_M = 1$ by definition) and $n_p = 30$. The samples of each flowline were taken at equally spaced intervals of time. We repeat the experiment for several refinements of the sphere and the results are shown in Fig. 6.2.

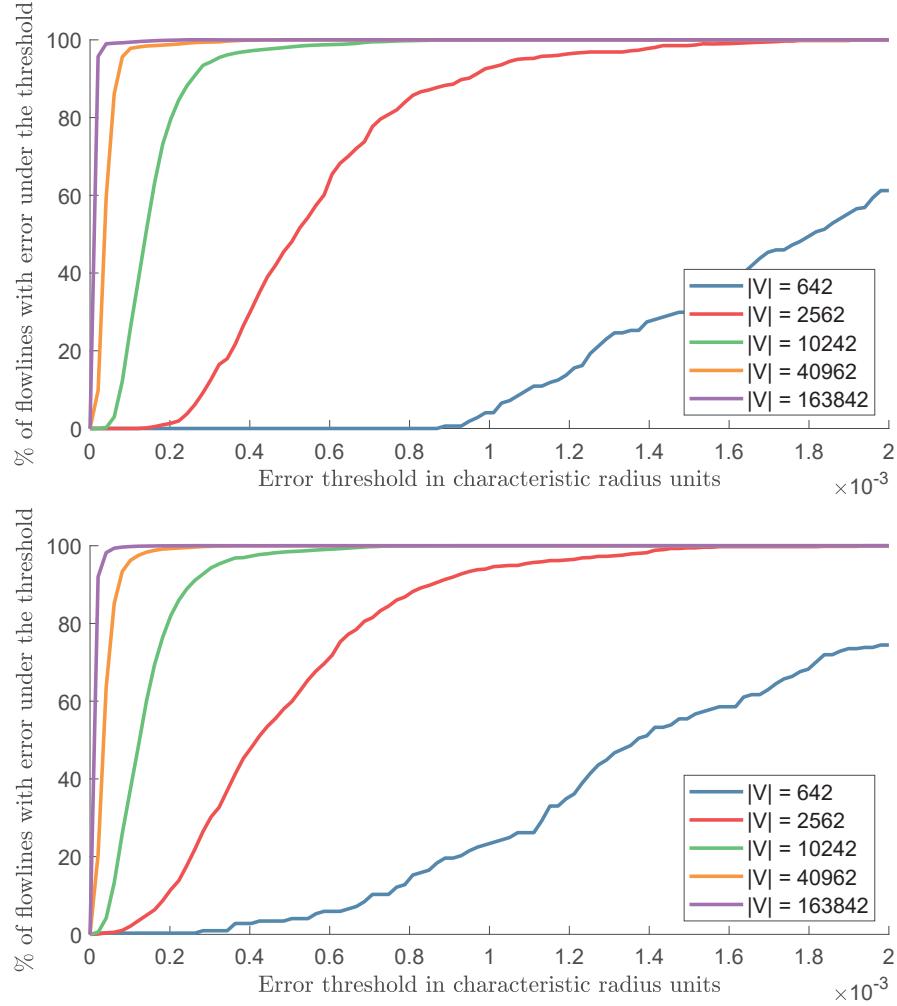


Figure 6.2: Comparing to analytic: The results for the vector fields $\beta \operatorname{grad} Z$ (top) and $\beta J \operatorname{grad} Z$ (bottom) for several refinements, where β is the usual normalization, and $t = 0.2r_M$. The flowline error was measured by the maximum sample error.

6.1.2 Tracing

Tracing the flowlines is the calculation of flowlines by "walking" in the vector field's direction from each flowline origin. We want to measure how much our solution deviates from a simple tracing. This measurement aims to check whether a tracing solution can be replaced by our solution in particular cases. Qualitative results of the given tracing algorithm, alongside our

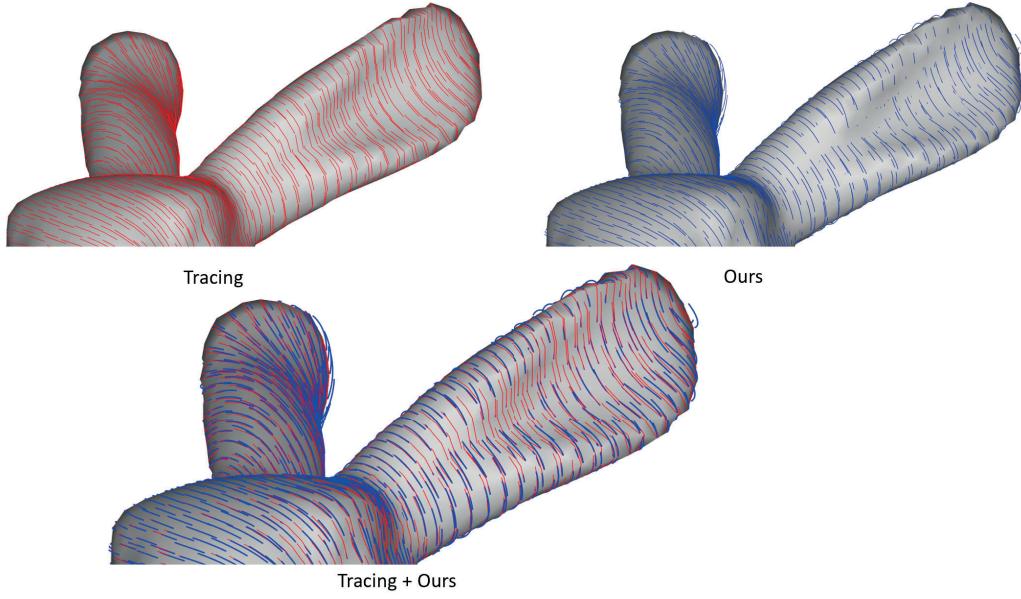


Figure 6.3: Qualitative comparison of our method with field tracing. While the results are visually similar, our approach leads to flowlines which might not lie on the input shape.

method's results are shown in Fig. 6.3. The following paragraph discusses the quantitative comparison.

For a given vector field, we calculate for each vertex p a flowline with $t = 0.2r_M$ and $n_p = 30$, and remove its first sample. We do it because the tracing algorithm requires origins in the interior of the mesh's faces. We denote the result as X'_p . We calculate the closest point on the mesh \tilde{p} to the first sample of X'_p . If \tilde{p} is on edge or a vertex we remove the flowline. Otherwise, We calculate a flowline from \tilde{p} by running the tracing algorithm (as implemented in [Vax19]) for 500 iterations in order to get a very long flowline, and we denote the result by \tilde{X}'_p . Finally, we cut the two polylines calculated from X'_p and \tilde{X}'_p to the length of the shorter one, and resample it 29 times at equally spaced intervals of length. We denote the resulting flowlines by X_p and \tilde{X}_p . For these flowlines we calculate the error as explained in the beginning of this chapter, where $\epsilon(e_p)$ is the maximum value of e_p . Results for several refinements of a Teddy 3D model are shown in Fig. 6.4. For all the refinements, we used the vector field v_{LB} as defined in Eq. (6.5), with the same coefficients.

6.1.3 Projected flowlines

The calculated points of the flowlines should lie on the surface, or at least "close to the surface" if we consider the triangle mesh as an approximation of a smooth surface. Therefore we compare our flowlines to their projection on the mesh. This is a measurement of one characteristic of ideal flowlines. Of course it is not enough as a single quality criterion because it does not take into account the correspondence to the vector field. Factors for the deviation are a coarse mesh discretization and numerical errors in the matrix exponential calculation.

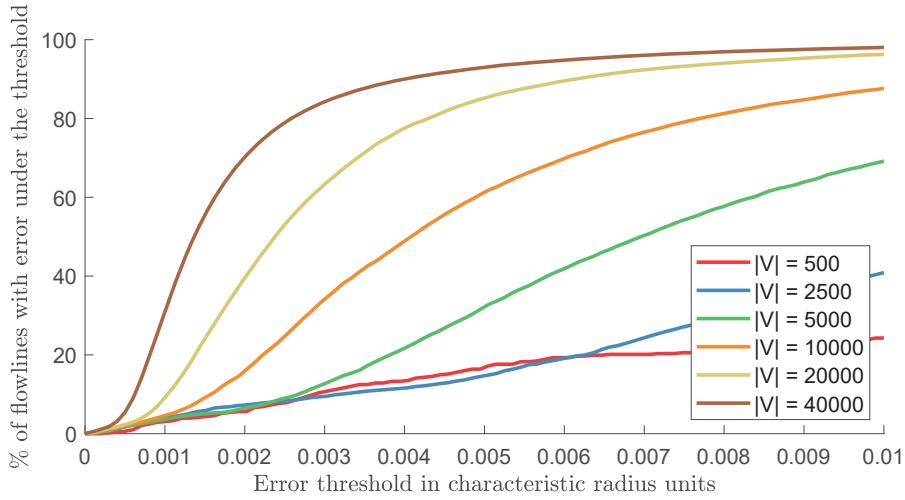


Figure 6.4: Comparison to tracing, for different levels of refinement of a Teddy mesh, with $t = 0.2r_M$.

In order to check how the distance from the surface depends on the refinement, we compute flowlines on a capped cylinder with a symmetry axis as the z axis, a radius 0.5, and with caps at $z = 0$ and $z = 2$. We compute the flowlines for the vector field $J \operatorname{grad} Z$, for different levels of mesh refinement. We calculate the error only for the flowlines between $z = 0.5$ to $z = 1.5$, to avoid the artifacts that the caps might introduce. The error is computed as explained in the beginning of this chapter, where $\epsilon(e_p)$ is the last value of e_p . The results are shown in Fig. 6.6. Note that as the resolution of the cylinders increases, and the discrete mesh better approximates a smooth cylinder, the distance of the computed flowline from the surface decreases.

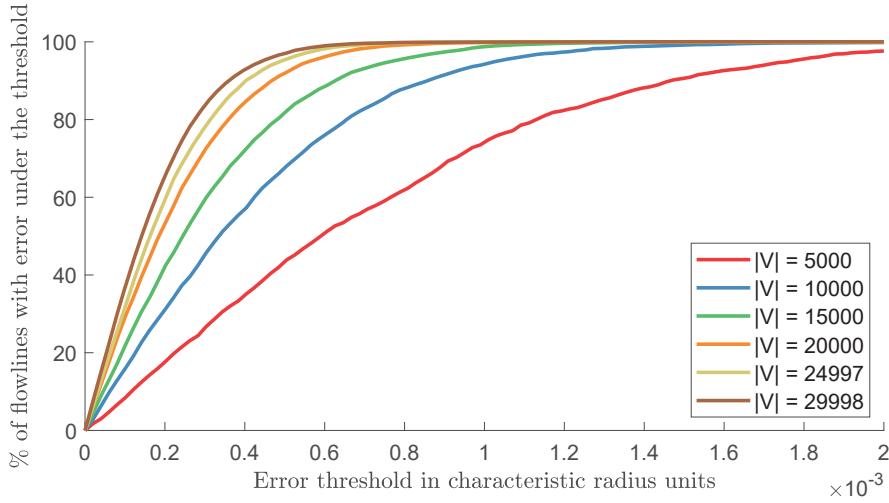


Figure 6.5: Distance from the surface for 6 cylinders with a symmetry as the z axis. The vector field is $v = J \operatorname{grad} Z$ for all refinements, and $t = 1r_M$ where $r_M = 0.5$ is the radius of the cylinders. The flowline error was measured by the distance from the end point to the mesh.

We repeat the experiment with a Teddy 3D model and three levels of refinement. We use the vector field $v = \beta(\text{grad } Z + J \text{ grad } Y)$ where β is the same for all refinements, and set such that the vector field of the finest refinement satisfies Eq. (5.3). The results are shown in Figure 6.6. Note that while the error is larger than for the cylinders, as the discrete model is refined, the projection error is reduced.

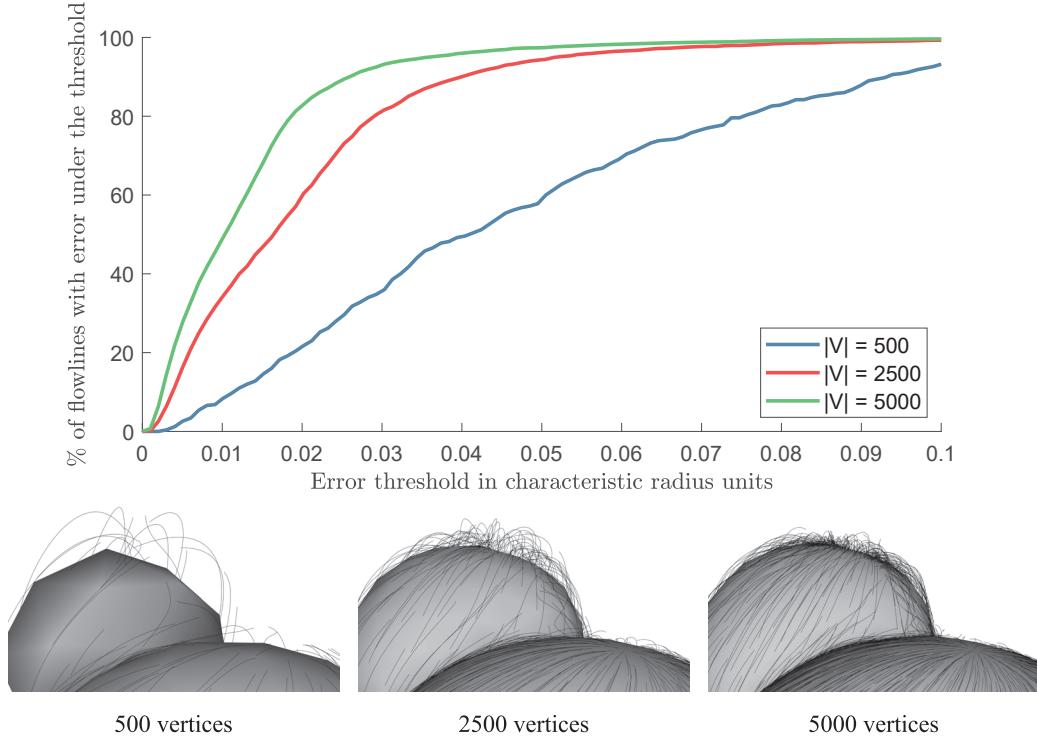


Figure 6.6: Distance from the surface for three levels of refinement of a teddy. The vector field $v = \beta(\text{grad } Z + J \text{ grad } Y)$ where β is the same for all refinements, and set such that the vector field of the finest refinement satisfies Eq. (5.3). $t = 0.5r_M$ and The flowline error was measured by the maximum sample error.

6.2 Local Behaviour

In this section we examine the local behaviour of the computed flowlines from two aspects. Namely, we examine the dependence of the calculation on the vector field values outside an area which contains the flowline, and the behaviour of the flowlines near singularities.

6.2.1 Locality

We say that a flowline calculation is *local* if it is not affected by the vector field outside an area that contains the flowline. Intuitively, we expect that the trajectory of a particle floating in a fluid will not be influenced by the velocity of the fluid outside the region containing the trajectory. Locality can also help us to analyze the flowlines behavior in local settings.

To investigate the locality property for a given vertex $p_{src} \in \mathcal{V}$, we generate k flowlines from this vertex as follows. We first set a vector field v_{in} in an area around p_{src} , consisting of all faces $\mathcal{F}_{in}(p_{src}, d) = \{r \in \mathcal{F} \mid \exists p \in r : d_{geo}(p, p_{src}) < d\}$, where $d \in \mathbb{R}_{>0}$, and $d_{geo}(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ is the geodesic distance between two vertices, calculated based on [CWW13]. We normalize v_{in} such that

$$\frac{1}{\sum_{r \in \mathcal{F}_{in}(p_{src}, d)} A_{\mathcal{F}}(r)} \sum_{r \in \mathcal{F}_{in}(p_{src}, d)} \|v_{in}(r)\| A_{\mathcal{F}}(r) = 1. \quad (6.3)$$

We then repeat the following process k times: In the area $\mathcal{F}_{out}(p_{src}, d) = \mathcal{F} \setminus \mathcal{F}_{in}(p_{src}, d)$ we generate a vector field v_{out} , and normalize it such that

$$\frac{1}{\sum_{r \in \mathcal{F}_{out}(p_{src}, d)} A_{\mathcal{F}}(r)} \sum_{r \in \mathcal{F}_{out}(p_{src}, d)} \|v_{out}(r)\| A_{\mathcal{F}}(r) = 1. \quad (6.4)$$

Then, we calculate a flowline from p_{src} by the vector field v which equals v_{in} in $\mathcal{F}_{in}(p_{src}, d)$ and v_{out} in $\mathcal{F}_{out}(p_{src}, d)$. After this process we have k flowlines from the vertex p_{src} , and we can measure the similarity between them.

In Fig. 6.7 we show the results for several meshes. For a given mesh we choose randomly 200 vertices, and for each vertex calculate $k = 2$ flowlines as explained above. We set $d = 0.4r_M$ and v_{in} is set randomly as in Eq. (6.5), with the additional normalization that we described. v_{out} is set randomly for each flowline in the same way as v_{in} and is normalized as explained above. Finally, the flowlines are calculated for $t = 0.2r_M$ and $n = 60$. We measure the error at a vertex as the maximal distance between two corresponding samples of the flowlines. We get a sense of the area \mathcal{F}_{in} and its relation to the flowline from the examples in Fig. 6.8. In this figure we show the results for several vertices on different meshes, where for each vertex we calculate $k = 100$ flowlines as explained above, with d , v_{in} , v_{out} and t as in the previous experiment, and $n = 30$. Both results support our conclusion that under these circumstances, the difference between different flowlines from the same vertex is small compared to the characteristic length, and thus the computation is indeed local as expected.

6.2.2 Singularities

Behaviour around singular points can be seen in Fig. 6.9. We see that our method is able to produce reasonable results around different kinds of singularities for $t = 0.2r_M$, and that at the singular point itself the flowline is very short or barely detectable.

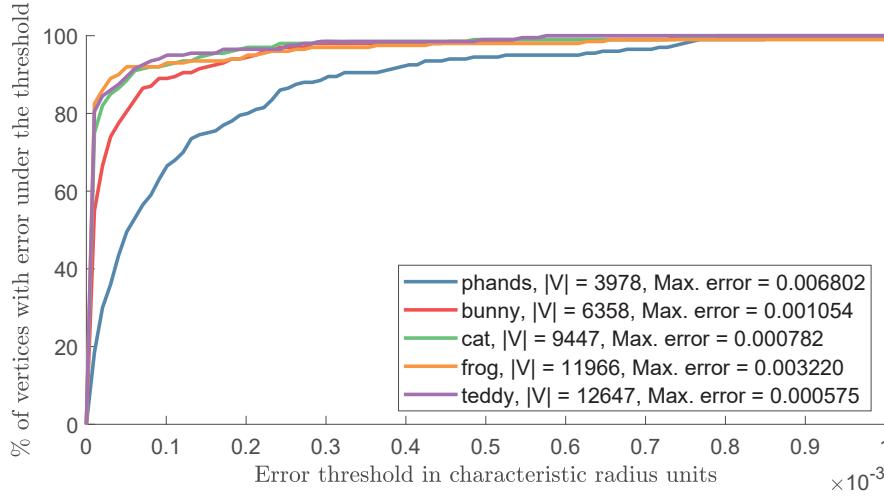


Figure 6.7: For each mesh, we measure the locality error for 200 vertices. *Max. error* is the maximal error observed at a vertex. Note that the difference between the two flowlines at each vertex is small compared to the characteristic length.

6.3 Application: Vector Field Visualization

In this chapter we show how to apply our flowlines calculation in order to visualize a vector field on a mesh, and also examine robustness qualitatively by showing visual results for irregular meshes. Here we take advantage of the fact that our computation is global, by computing the flowlines from *all the vertices* simultaneously.

We define a family of vector fields v_{LB} that we use to demonstrate the vector field visualization:

$$v_{LB} = \beta \sum_{k=1}^{100} \alpha_k \operatorname{grad} \varphi_k + \beta_k J \operatorname{grad} \varphi_k, \quad (6.5)$$

where $\{\varphi_k\}$ are the eigenvectors of the Laplace Beltrami operator sorted in ascending order by eigenvalue, $\{\alpha_k\}$ and $\{\beta_k\}$ are scalars in the segment $(-0.5, 0.5)$, and $\beta > 0$ is a normalization factor set as in Eq. (5.3).

The visualization of a vector field is generated by drawing a flowline from each vertex with $t = 0.1r_M$, and showing the vector norm underneath it as a face-based function. Results on a variety of meshes are shown in Fig. 6.10.

Results on non-regular meshes are shown in Fig. 6.11. It seems that our method produces more aesthetic results in regions which are more densely triangulated.

A case of problematic result is presented in Fig. 6.12. The problem appears in a small number of vertices, in a coarse region of the mesh.

6.3.1 Timings

The calculation time is a few seconds even for very fine meshes, as shown in Fig. 6.13.

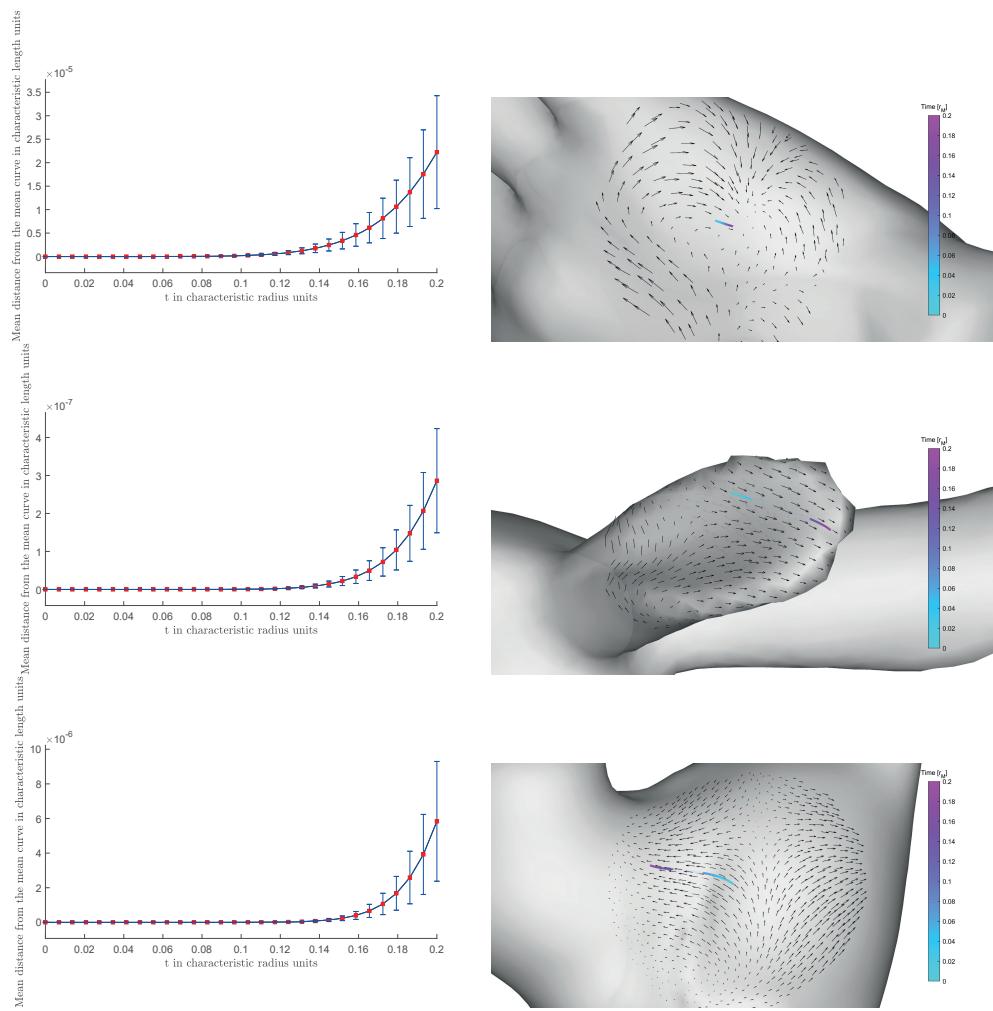


Figure 6.8: Measuring the locality of a flowline calculation in single vertices.

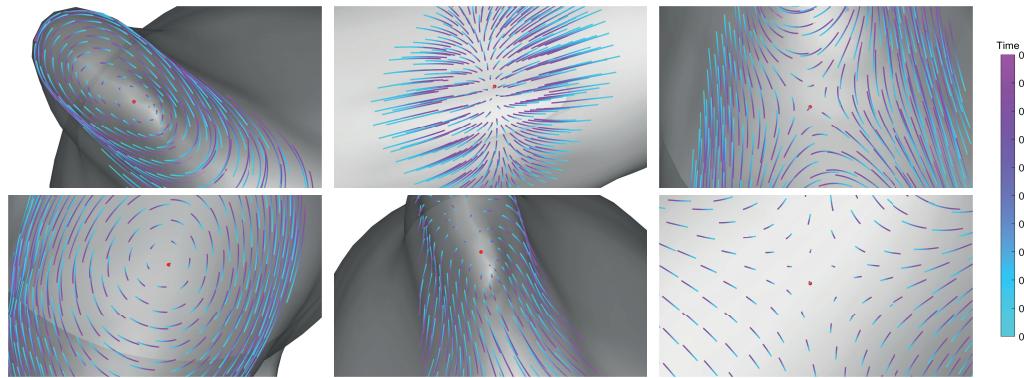


Figure 6.9: Behaviour around different kinds of singular points.

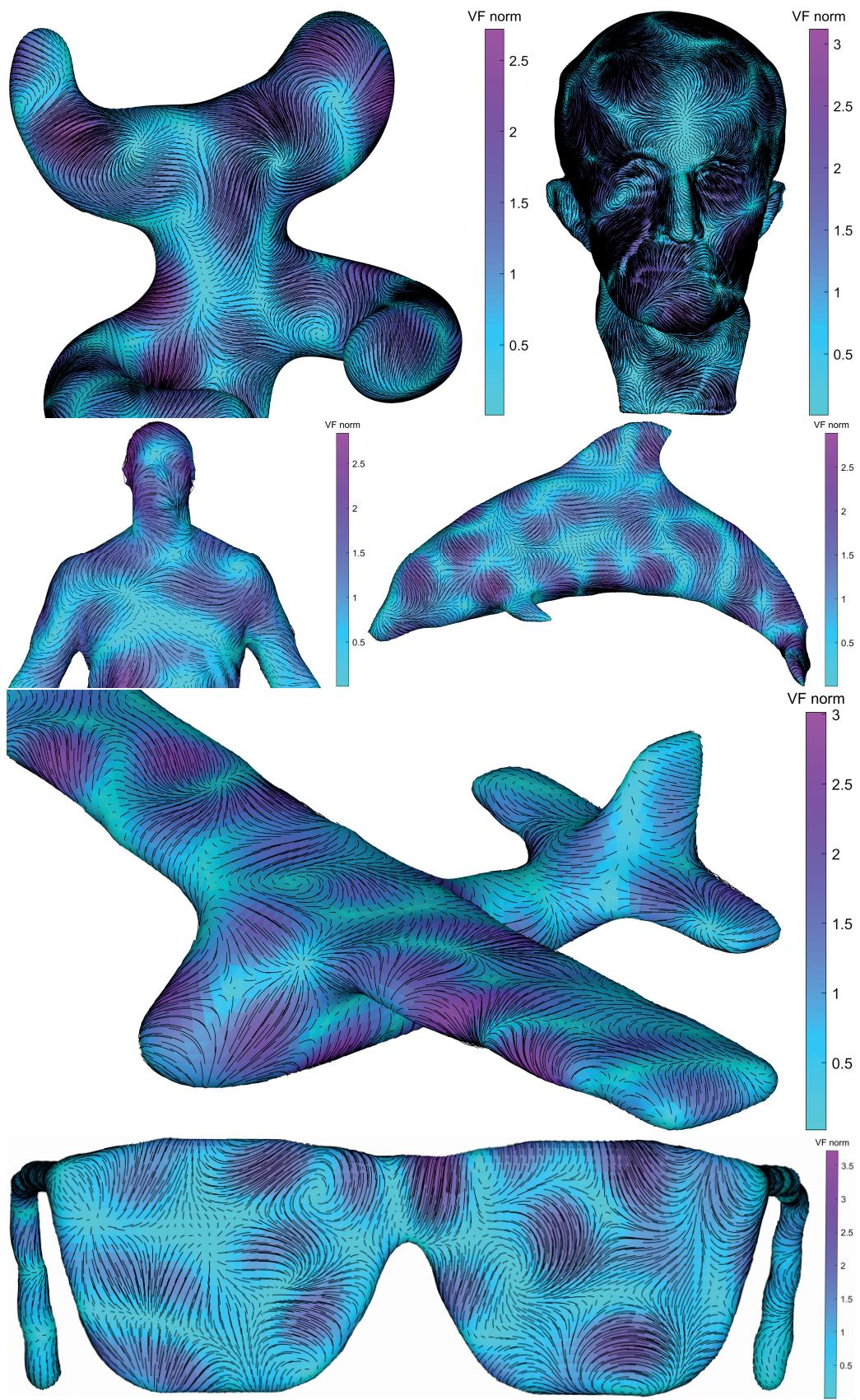


Figure 6.10: Visualisation of a vector field. The direction represented as flowlines, and the norm represented as a color on the faces.
30

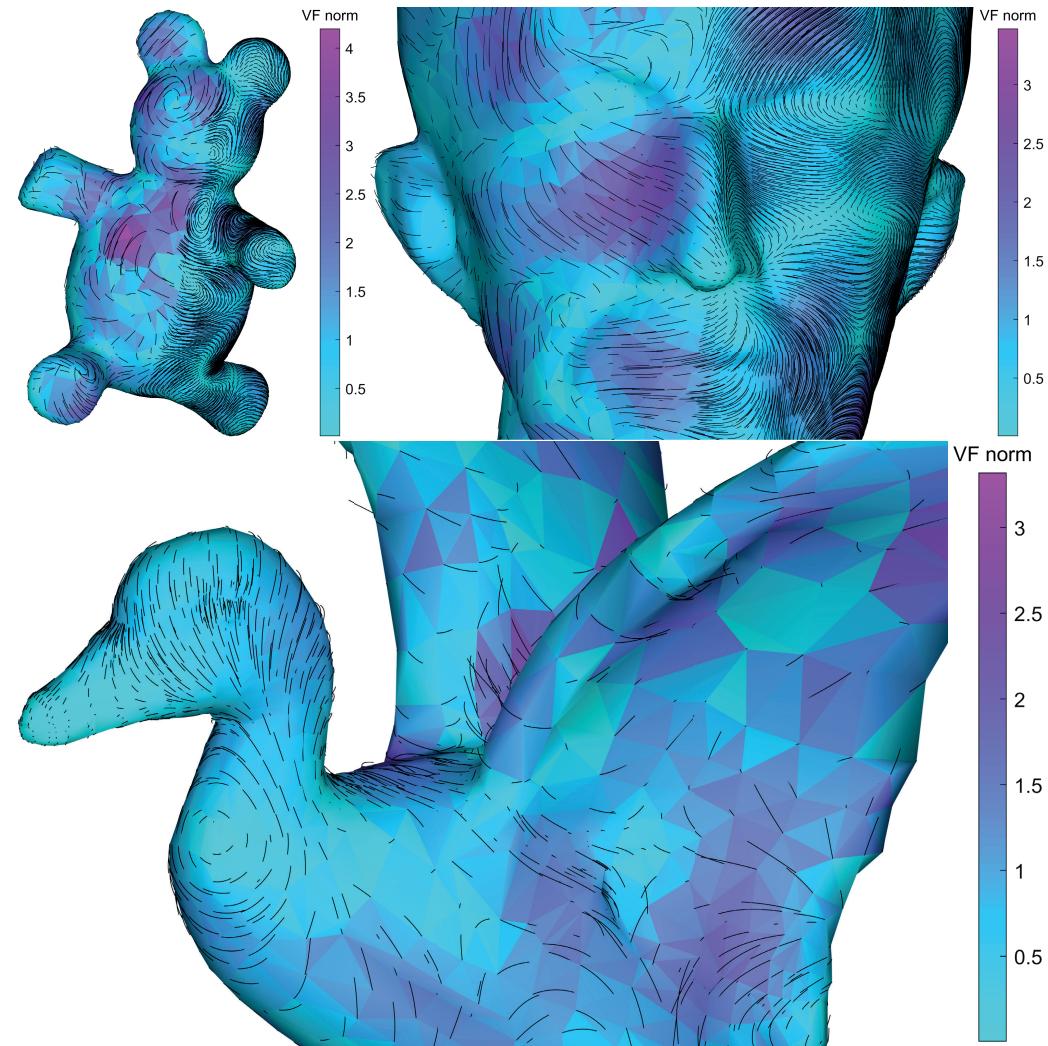


Figure 6.11: A vector field visualization on a non-regular mesh. It seems that our method produces more aesthetic results in regions which are more densely triangulated.

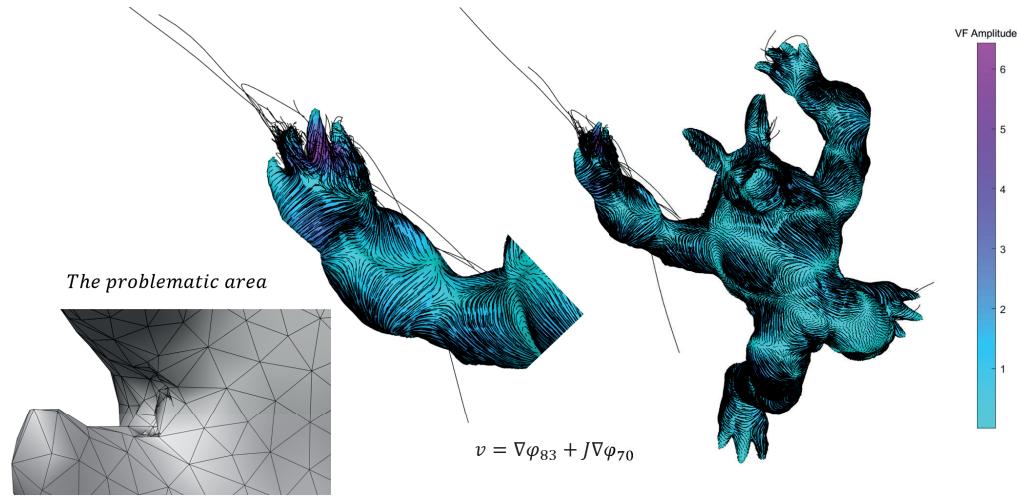


Figure 6.12: We see that the solution explodes in a small area of the mesh, due to a problematic triangulation in that region.

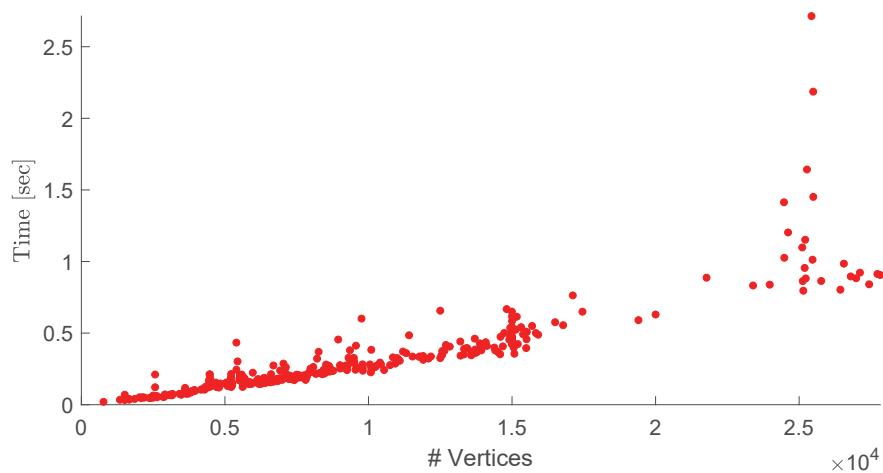


Figure 6.13: The calculation time as a function of the number of vertices for the shapes in SHREC07 with genus 0 and with no boundary for $t = 0.1r_M$.

Chapter 7

Summary

7.1 Limitations

Our method works only with piecewise constant vector fields on a triangle mesh. We note that this is due to the specific representation of the discrete covariant derivative operator, and the functions in our setup. For another setup, once we have corresponding discrete representations of the covariant derivative operator as a square matrix, we can try applying our main part of our method - the matrix exponential times the coordinate functions, to this setup. A problem concerning our method is that the flowlines are not guaranteed to lie on the surface, and the error grows for a large t and coarse surfaces. Another limitation of our approach is that flowlines can originate only at vertices. We note that generating a flowline from a vertex can also be an advantage, because the vector field is not defined on vertices, a fact that challenges tracing algorithms.

7.2 Conclusions

We present a simple method for calculating flowlines of a vector field on a triangle mesh. The method makes sense from several aspects such as comparing to an analytical solution in a known case, qualitatively comparing to tracing, locality, behavior around singularities and on non-regular meshes. We can also use it to visualize vector field on meshes.

7.3 Future Work

Ideas for future work are applying of our method to geodesics computation given a geodesic vector field, generalization of our method to multi-vectors (i.e. cross fields), or incorporating this approach in other applications that require flowlines (e.g. quad meshing).

Bibliography

- [ABCCO13] Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Maks Ovsjanikov. An operator approach to tangent vector field processing. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, pages 73–82. Eurographics Association, 2013.
- [AMH11] Awad H Al-Mohy and Nicholas J Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM journal on scientific computing*, 33(2):488–511, 2011.
- [AWO⁺14] Omri Azencot, Steffen Weißmann, Maks Ovsjanikov, Max Wardetzky, and Mirela Ben-Chen. Functional fluids on surfaces. In *Computer Graphics Forum*, volume 33, pages 237–246. Wiley Online Library, 2014.
- [BCA19] Mirela Ben-Chen and Omri Azencot. Operator-based representations of discrete tangent vector fields. 2019.
- [BKP⁺10] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. AK Peters/CRC Press, 2010.
- [CL93] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. Technical report, Lawrence Livermore National Lab., CA (United States), 1993.
- [CWW13] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):1–11, 2013.
- [FSDH07] Matthew Fisher, Peter Schröder, Mathieu Desbrun, and Hugues Hoppe. Design of tangent vector fields. *ACM transactions on graphics (TOG)*, 26(3):56–es, 2007.
- [HO10] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.
- [HZ00] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 517–526, 2000.

- [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.
- [Mor01] Shigeyuki Morita. *Geometry of differential forms*. Number 201. American Mathematical Soc., 2001.
- [MVC05] Dimas Martínez, Luiz Velho, and Paulo C Carvalho. Computing geodesics on triangular meshes. *Computers & Graphics*, 29(5):667–675, 2005.
- [PPM⁺16] Nico Pietroni, Enrico Puppo, Giorgio Marcias, Roberto Scopigno, and Paolo Cignoni. Tracing field-coherent quad layouts. In *Computer Graphics Forum*, volume 35, pages 485–496. Wiley Online Library, 2016.
- [PS06] Konrad Polthier and Markus Schmies. *Straightest geodesics on polyhedral surfaces*. ACM, 2006.
- [RRP15] Faniry H Razafindrazaka, Ulrich Reitebuch, and Konrad Polthier. Perfect matching quad layouts for manifold meshes. In *Computer Graphics Forum*, volume 34, pages 219–228. Wiley Online Library, 2015.
- [RS14] Nicolas Ray and Dmitry Sokolov. Robust polylines tracing for n-symmetry direction field on triangulated surfaces. *ACM Transactions on Graphics (TOG)*, 33(3):30, 2014.
- [RT11] Christian Rossl and Holger Theisel. Streamline embedding for 3d vector field exploration. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):407–420, 2011.
- [SSK⁺05] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. *ACM transactions on graphics (TOG)*, 24(3):553–560, 2005.
- [Vax19] Amir Vaxman. Directional: A library for directional field synthesis, design, and processing, 2019.
- [ZHT06] Eugene Zhang, James Hays, and Greg Turk. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):94–107, 2006.

הגישה שלנו דורשת רק את הבניה של אופרטור גזירה שמייצג את השדה הוקטורי, והכפלה של האקספונט של מטריצה דיליה בוקטור, ואלה שתי פעולות שניתן לחשב ביעילות. עבור וקטור זמן מחולק באופן שווה, אנו מחשבים את קווי הזרימה מכל הצמתים של המשטח בו זמן. על ידי הגדרה גlobלית זו של הבעיה, השיטה שלנו מאופיינת על ידי שימוש בעיקר בפתרונות גלובליים, בניגוד לאלגוריתמים אחרים שמנתחים את הגאומטריה באופן מקומי תוך כדי חישוב מפורש של העקומים ותוך כדי חישוב חיתוך בין קטעי קוים ובניות מסובכות.

mdiידת השיטה שלנו נעשית במספר בחינות שונות. אנו מראים באופן אנלטי שהאלגוריתם שלנו מתחלב עם טרנספורמציה אפינית גלובלית של המשטח ושל השדה הוקטורי. כאמור, תוצאה ביצוע הטרנספורמציה וחישוב קווי הזרימה לא תליה בסדר ביצוע הפעולות. אנו משווים את חישוב קווי הזרימה לפתרונות אנליטיים במרקם ידועים. אנו משווים את השיטה שלנו לאלגוריתם עקיבה איטרטיבי פשוט. עיקרונו הפעולה של אלגוריתם העקיבה, הוא "הליכה" מנקודת ההתחלה של כל קו זרימה, שהיא נקודה פנימית של מושלש, בכו ישר בכיוון השדה, עד שהמשולש פוגש גבול של מושלש. ממש האלגוריתם מחשב את המושלש הבא וחוזר על אותו תהליך מספר איטרציות נתון. אנו בוחנים את מרחק קווי הזרימה מהמשטח, שכפי שהם מוגדרים בגאומטריה דיפרנציאלית, אמורים להיות על גבי המשטח. בנוסף, אנו בוחנים שני מאפיינים מקומיים של קווי הזרימה שאנו מחשבים. מאפיין מקומי אחד הוא הלוקליות של החישוב, כלומר כמה החישוב באזור מסוים על המשטח תלוי בשדה הוקטורי מחוץ לאזור, ומאפיין מקומי אחר הוא התנהגות קווי הזרימה בסביבת נקודות סינגולריות של השדה הוקטורי. לבסוף אנו משתמשים בשיטה שלנו עבור ויזואליזציה של שדה ווקטורי משיק על משולשי באופן פשוט, רובסטי, ויעיל.

תקציר

אנו מציעים שיטה לחישוב קווי הזרימה של שדה ווקטורי דיסקרטי המשיק למשטה המיצג כמש מושלשי, ככלומר המורכב ממושלים המוחברים ביניהם על ידי צלעות משותפות או קודקודים משותפים. באופן תיאורטי, כאשר אנו מדברים על משטה ושדה ווקטורי "חלקיים", קווי הזרימה של השדה הם עוקומים פרמטריים על המשטה שבכל נקודה מהירותם שווה לשדה הווקטורי בנקודה. באופן אינטואיטיבי, אם השדה הווקטורי מייצג מהירות של נוזל בכל נקודה על המשטה, אז קווי הזרימה הם מסלולים של חלקיקים שימושיים על ידי הנוזל. חישוב קווי הזרימה של שדה ווקטורי משיק למשטה היא פוליה נפוצה בעיבוד גיאומטרי ובגרפייה ממוחשבת. היא משמשת בין השאר לביצוע סגמנטציה של משטחים, ויזואלייזציה של שדה ווקטורי בנסיבות שונות.

השיטה שלנו עשויה שימוש בייצוג של השדה הווקטורי כאופרטור גזירה. ככלומר, אופרטור שפועל על פונקציה על המשטה ומחזיר פונקציה על המשטה, שהיא נגזרת הפונקציה המקורית בכיוון השדה המשיק בכל נקודה. ייצוג זה של השדה הווקטורי מאפשר לנו לתאר את הבעה של חישוב קווי הזרימה כהסעה של פונקציות על גבי המשטה.

באופן מפורט יותר, המשטה משוכן למרחב תלת-ממדי ולכן ניתן לתאר מיקום כל נקודה על המשטה באמצעות שלוש קואורדינטות קרטזיות. פונקציית קואורדינטה קרטזית על המשטה היא ההתאמנה של הקואורדינטה הקרטזית לכל נקודה על המשטה. חישוב קווי הזרימה על המשטה ניתן לחישוב, כפי שניתן באופן תיאורטי בתזה, על ידי ההסעה של פונקציות הקואורדינטות الكرטזיות על המשטה. ניתן להבין הסעה של פונקציה באופן אינטואיטיבי, כנסיעה של חלקיקים הנושאים את ערכי הפונקציה על גבי שדה מהירות המשיך על ידי השדה הווקטורי המשיק. זהה הסתכלות גlobלית על חישוב קווי הזרימה. במקום לחשב על מסלול חלקיק בודד עשויה על המשטה עם הזמן, אנו חושבים על ההסתכלות בזמן של פונקציות על גבי המשטה.

אופרטור הגזירה בו אנו עושים שימוש בפועל מיוצג על ידי מטריצה ריבועית דיליה, באופן שהוציא לאחורונה. על ידי הצגה של השדה הווקטורי כאופרטור גזירה, ועל ידי דיסקרטיזציה של פונקציות הקואורדינטות באופן מקובל – ערך עבור כל קודקוד של המש, אנו מקבלים שלוש מערכות לינאריות של משוואות דיפרנציאליות רגילות, מערכת אחת עבור כל פונקציית קואורדינטה קרטזית. למערכות אלה יש פיתרון סגור כפונקציה של הזמן, וכן המערכת יכולה להיפתר בלי דיסקרטיזציה מפורשת של הזמן. בפועל זה נעשה על ידי שימוש באקספוננט של מטריצה דיליה שהיא אופרטור הגזירה מוכפל בזמן.

המחקר בוצע בהנחייתה של פרופסור מירלה בונץון, בפקולטה למדעי המחשב.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

מעקב פונקציונלי של שדות וקטוריים דיסקרטיים על משטח

חיבור על מחקר

לשם מלאי חלקו של הדרישות לקבלת התואר
מגיסטר למדעים במדעי המחשב

יאיר ריעאני

הוגש לסנט הוכניון --- מכון טכנולוגי לישראל
שבט התש"פ חיפה פברואר 2020

מעקב פונקציונלי של שדות ווקטוריים דיסקרטיים על משטח

יאיר ריעאני