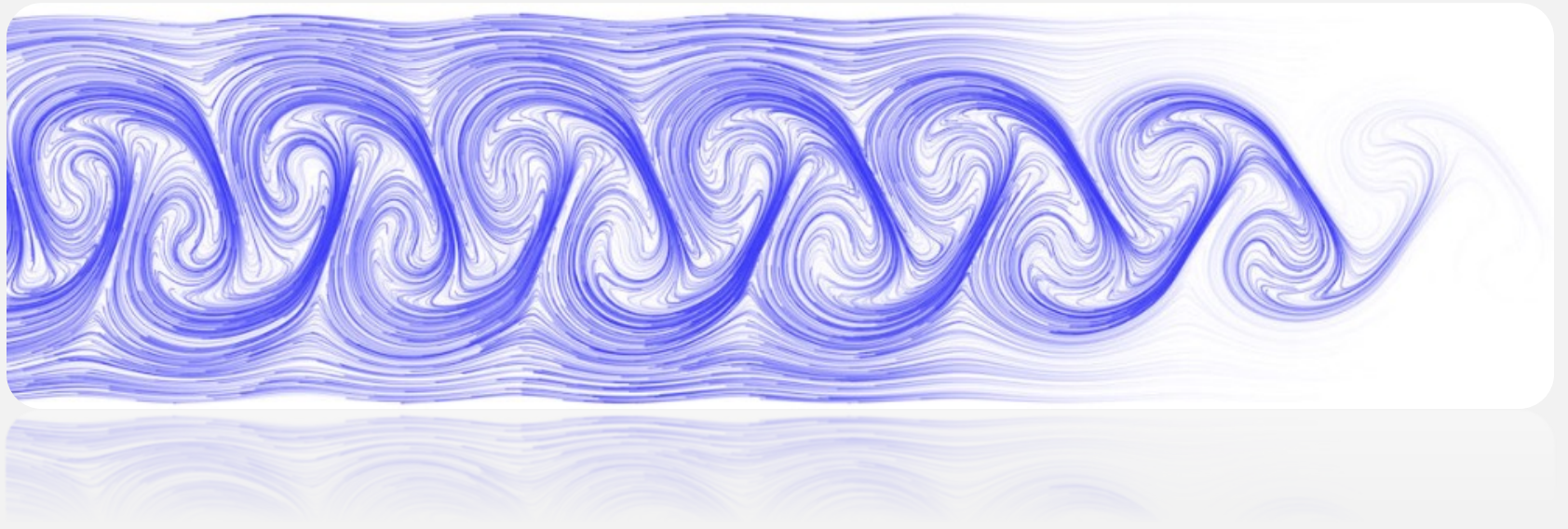


# IFT 6113

## VECTOR FIELDS

[tiny.cc/ift6113](http://tiny.cc/ift6113)



Mikhail Bessmeltsev

# Lots of Material/Slides From...

## *Vector Field Processing on triangle meshes*

Fernando de Goes

Pixar Animation Studios

Mathieu Desbrun

Caltech

Yiying Tong

Michigan State University



Render the Possibilities  
**SIGGRAPH2016**

**Check out  
course notes!**

# Additional Nice Reference

EUROGRAPHICS 2016  
J. Madeira and G. Patow  
(Guest Editors)

Volume 35 (2016), Number 2  
STAR – State of The Art Report

## Directional Field Synthesis, Design, and Processing

Amir Vaxman<sup>1</sup> Marcel Campen<sup>2</sup> Olga Diamanti<sup>3</sup> Daniele Panozzo<sup>2,3</sup> David Bommes<sup>4</sup> Klaus Hildebrandt<sup>5</sup> Mirela Ben-Chen<sup>6</sup>

<sup>1</sup>Utrecht University <sup>2</sup>New York University <sup>3</sup>ETH Zurich <sup>4</sup>RWTH Aachen University <sup>5</sup>Delft University of Technology <sup>6</sup>Technion

---

### Abstract

*Direction fields and vector fields play an increasingly important role in computer graphics and geometry processing. The synthesis of directional fields on surfaces, or other spatial domains, is a fundamental step in numerous applications, such as mesh generation, deformation, texture mapping, and many more. The wide range of applications resulted in definitions for many types of directional fields: from vector and tensor fields, over line and cross fields, to frame and vector-set fields. Depending on the application at hand, researchers have used various notions of objectives and constraints to synthesize such fields. These notions are defined in terms of fairness, feature alignment, symmetry, or field topology, to mention just a few. To facilitate these objectives, various representations, discretizations, and optimization strategies have been developed. These choices come with varying strengths and weaknesses. This report provides a systematic overview of directional field synthesis for graphics applications, the challenges it poses, and the methods developed in recent years to address these challenges.*

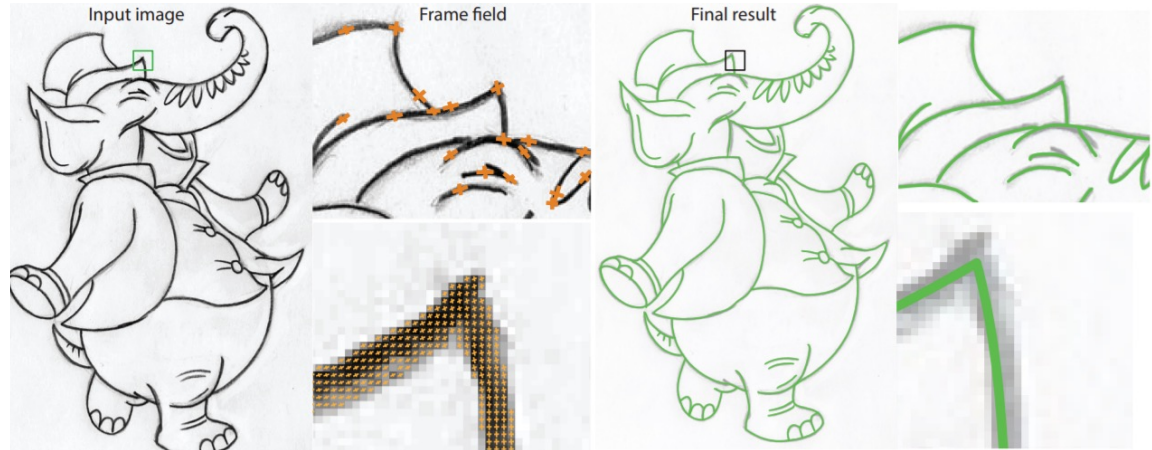
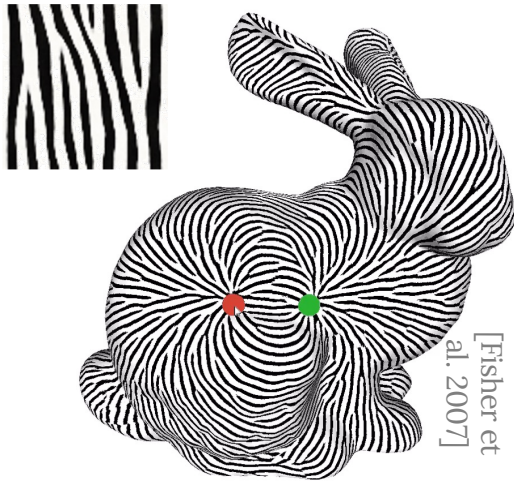
Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—

---

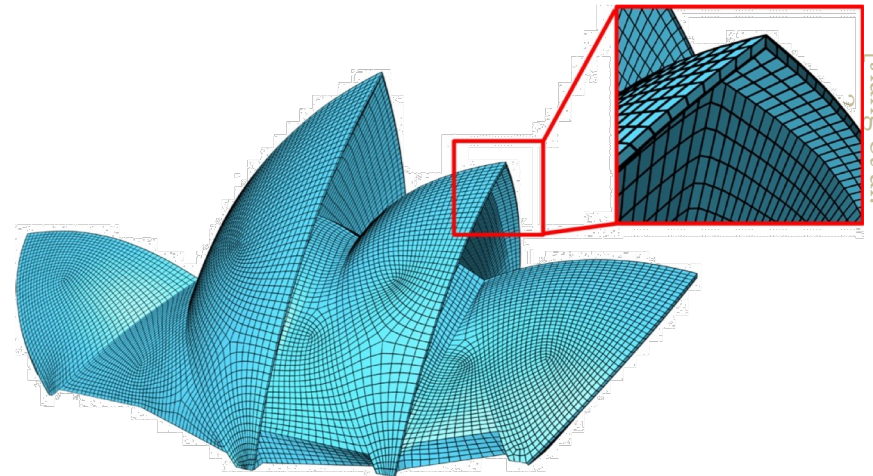
### 1. Introduction

There have been significant developments in directional field synthesis over the past decade. These developments have been driven

# Why Vector Fields?

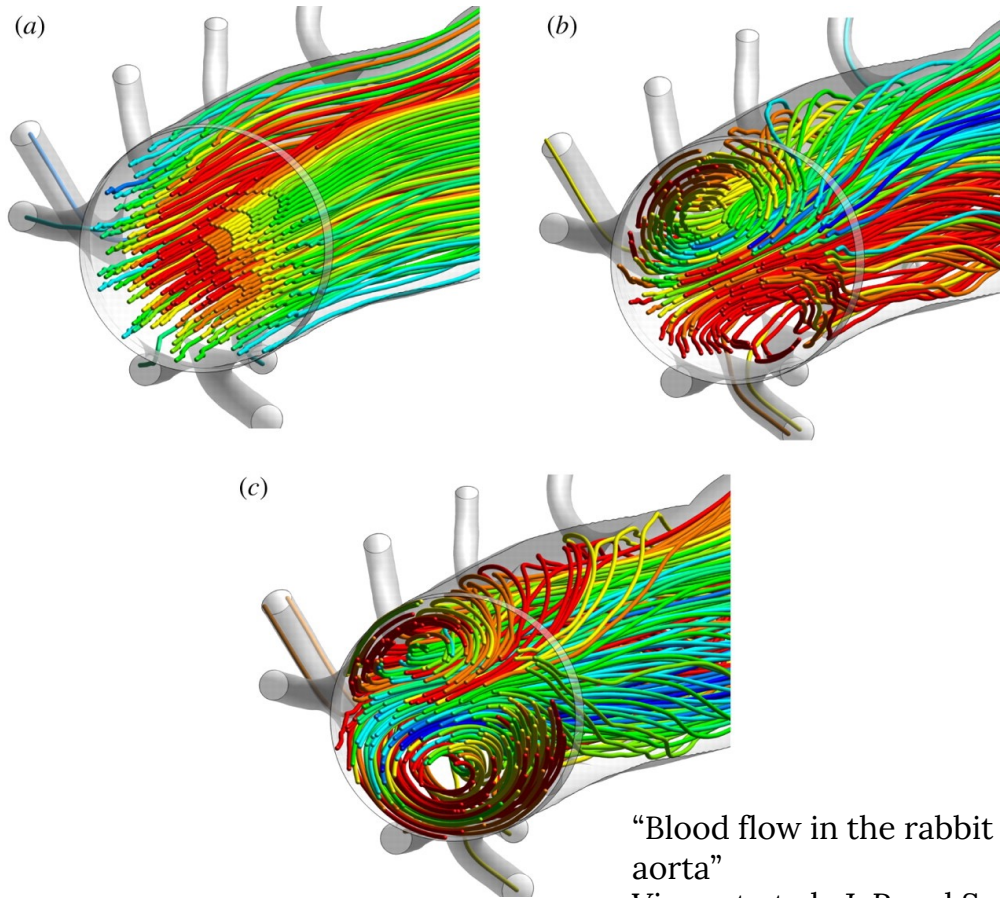


© Disney/Pixar



# Graphics

# Why Vector Fields?

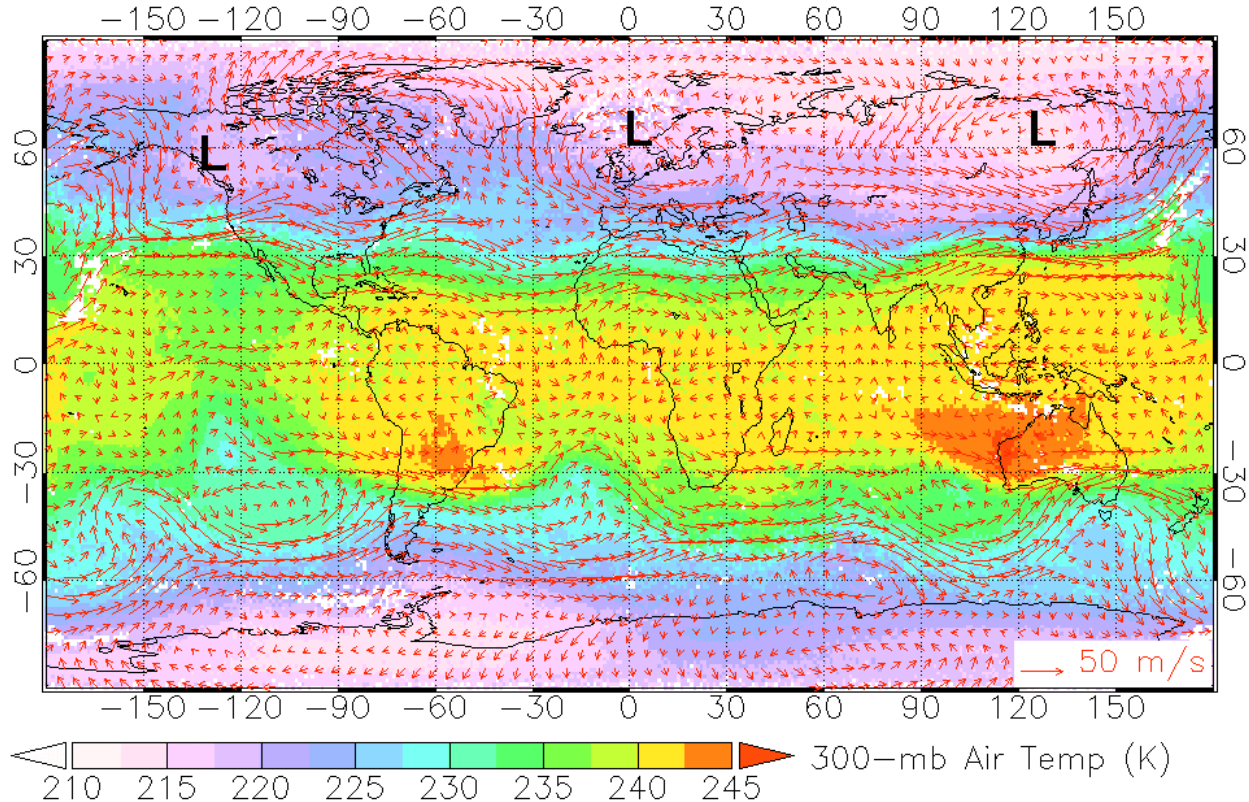


“Blood flow in the rabbit aortic arch and descending thoracic aorta”

Vincent et al.; J. Royal Society 2011

***Biological science and imaging***

# Why Vector Fields?



<https://disc.gsfc.nasa.gov/featured-items/airs-monitors-cold-weather>

# Weather modeling

# Fluid modeling



# Why Vector Fields?



<https://forum.unity3d.com/threads/megaflow-vector-fields-fluid-flows-released.278000/>

*Simulation and engineering*



# Plan

***Crash course***

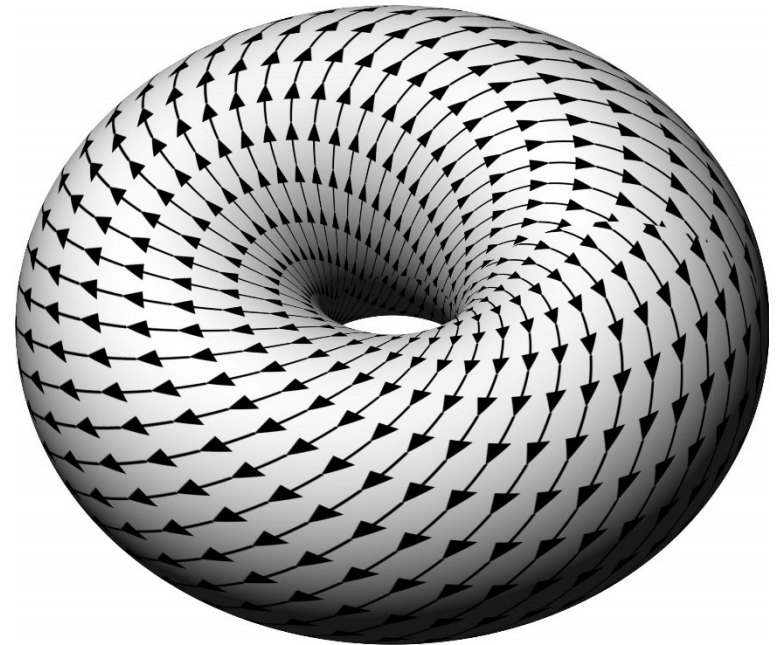
***in theory/discretization of vector fields.***

**CONTINUOUS**

# Studying Vector fields

How to

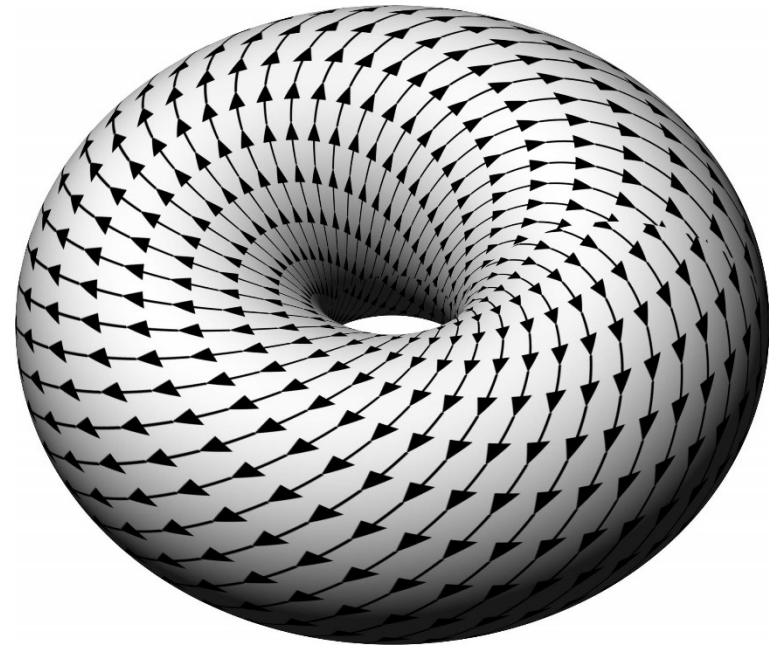
- define a VF on a surface?
- Differentiate it?
- Integrate?
- Define its topology?



# Studying Vector fields

How to

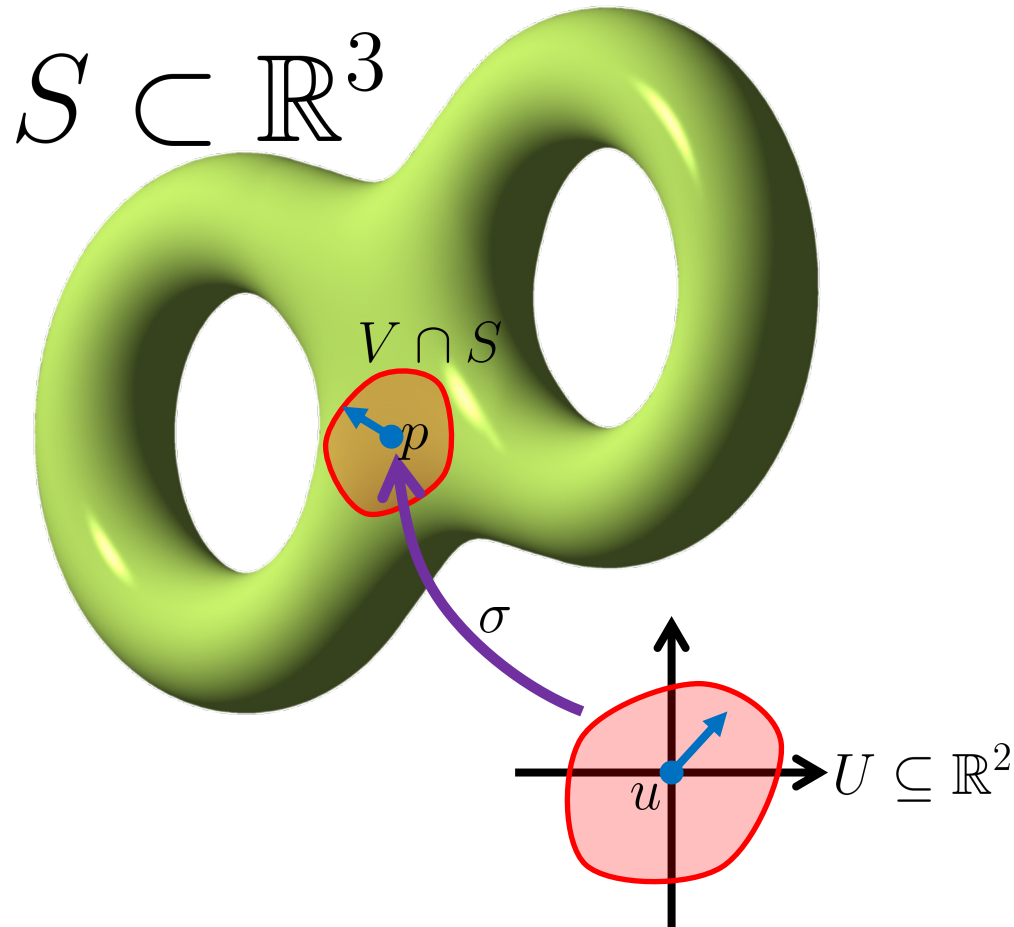
- ***define a VF on a surface?***
- Differentiate it?
- Integrate?
- Define its topology?



*Recall:*

# Tangent Space

$$T_p S := \text{Image}(D\sigma_u)$$



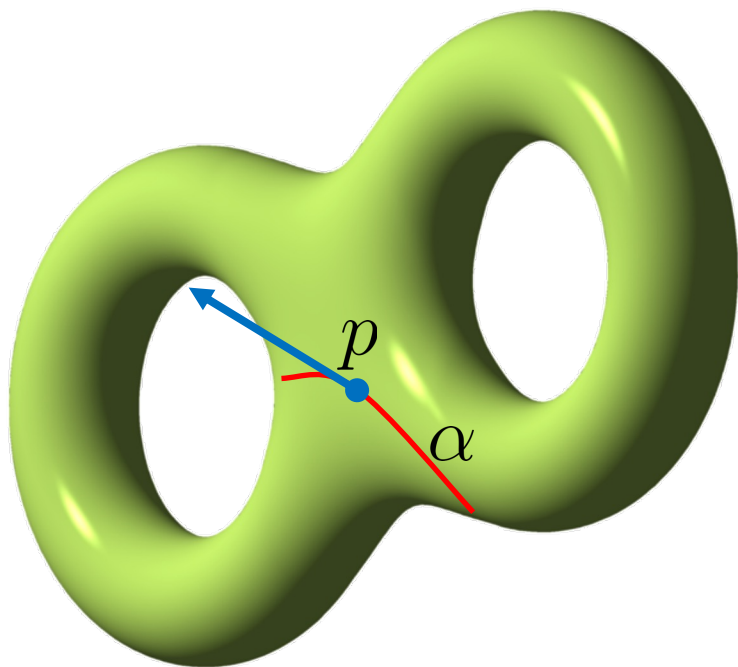
*Recall:*

# Tangent Space: Coordinate-Free

$$v \in T_p S \iff$$

there exists curve  $\alpha : (-\varepsilon, \varepsilon) \rightarrow S$

with  $\alpha(0) = p, \alpha'(0) = v$



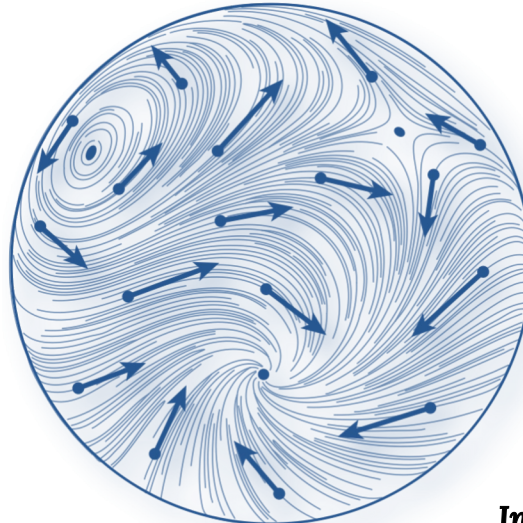
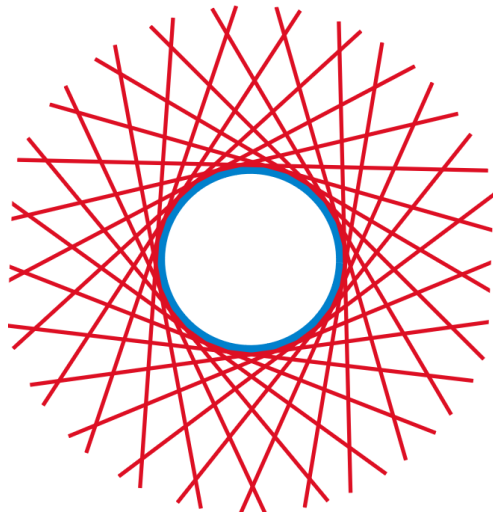
# Some Definitions

Tangent bundle:

$$TM := \{(p, v) : v \in T_p M\}$$

Vector field:

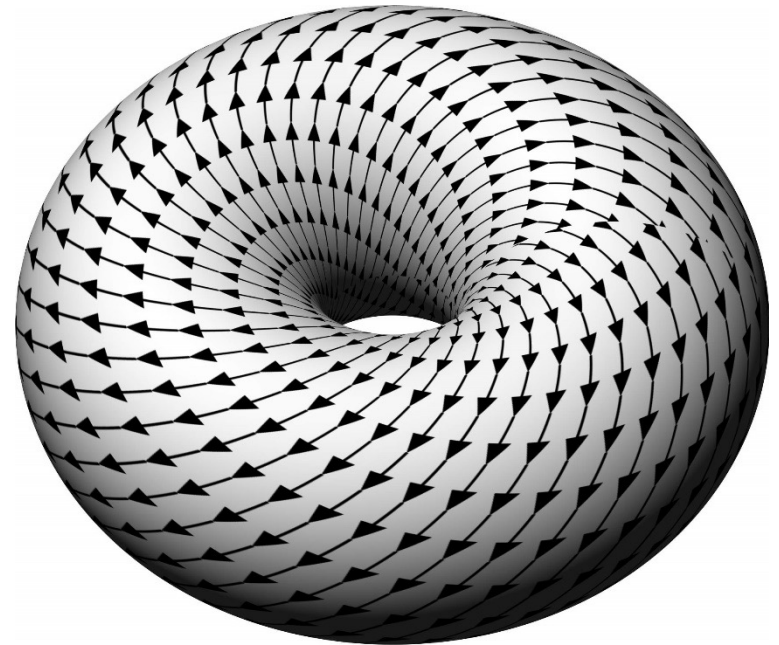
$$u : M \rightarrow TM \text{ with } u(p) = (p, v), v \in T_p M$$



# Studying Vector fields

How to

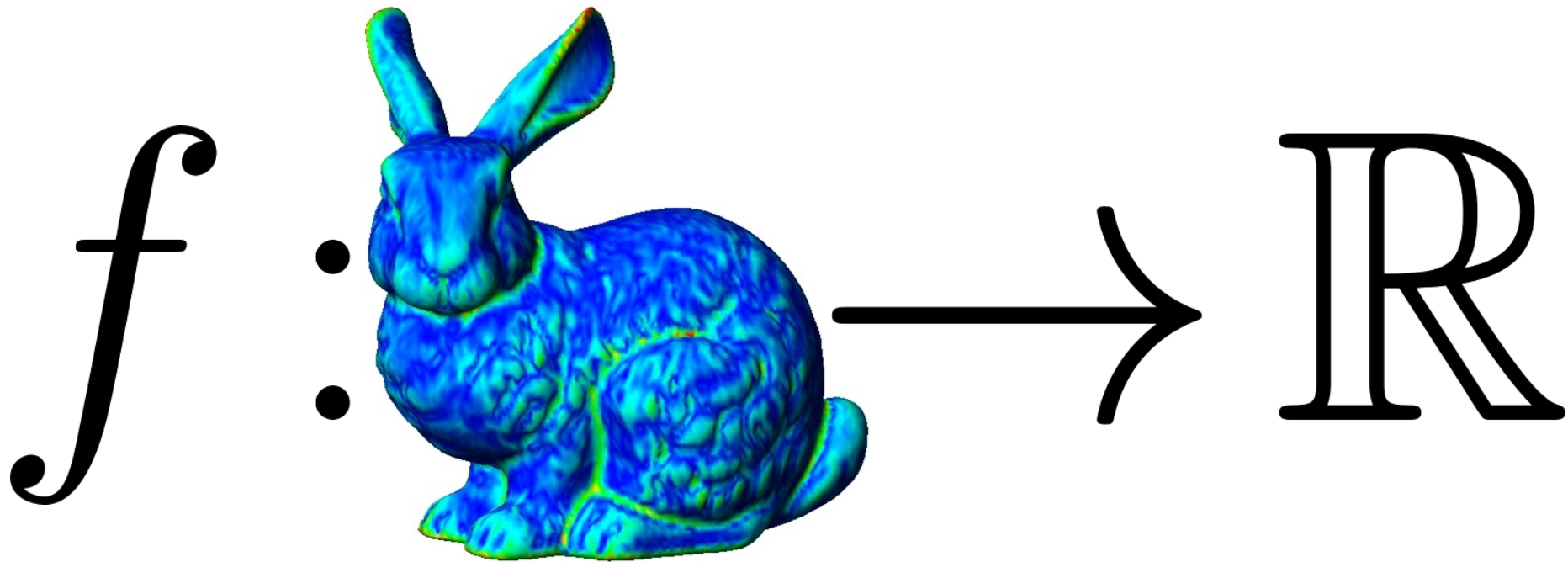
- define a VF on a surface?
- ***Differentiate it?***
- ***Integrate?***
- Define its topology?





*Recall:*

# Scalar Functions



[http://www.ieeta.pt/polymeco/Screenshots/PolyMeCo\\_OneView.jpg](http://www.ieeta.pt/polymeco/Screenshots/PolyMeCo_OneView.jpg)

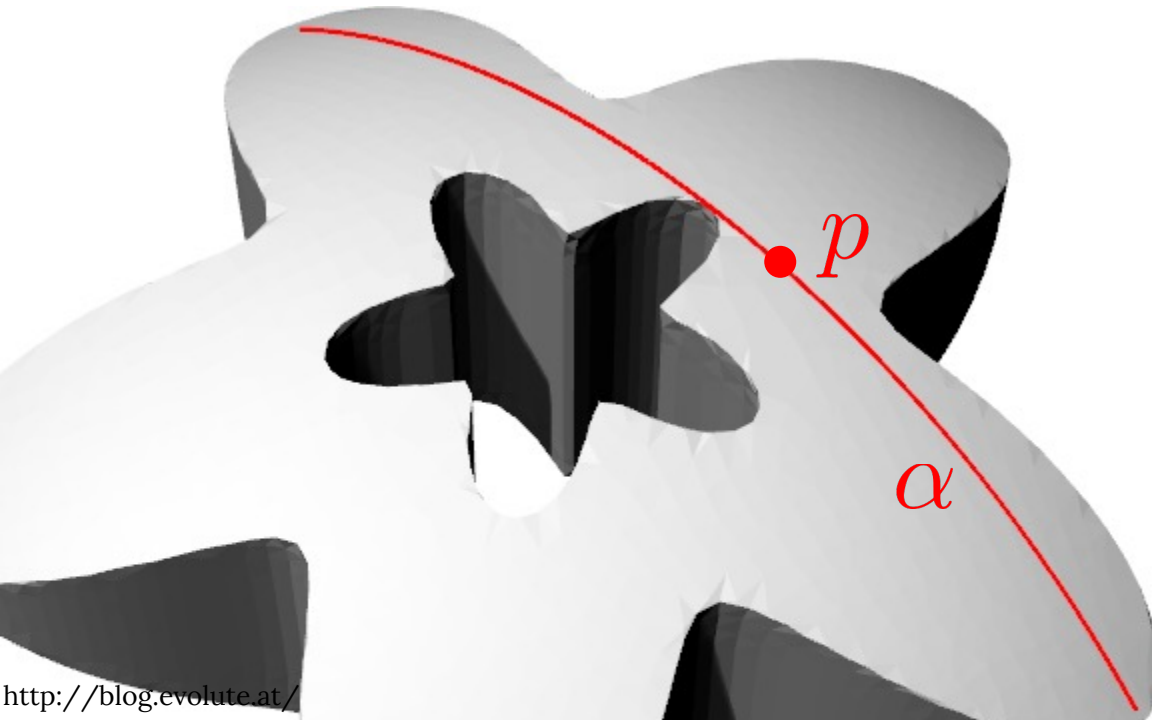
***Map points to real numbers***

*Recall:*

# Differential of a Map

Suppose  $f: S \rightarrow \mathbb{R}$  and take  $p \in S$ . For  $v \in T_p S$ , choose a **curve**  $\alpha: (-\varepsilon, \varepsilon) \rightarrow S$  with  $\alpha(0) = p$  and  $\alpha'(0) = v$ . Then the differential of  $f$  is  $df: T_p S \rightarrow \mathbb{R}$  with

$$(df)_p(v) := \left. \frac{d}{dt} \right|_{t=0} (f \circ \alpha)(t) = (f \circ \alpha)'(0).$$



**On the board (time-permitting):**

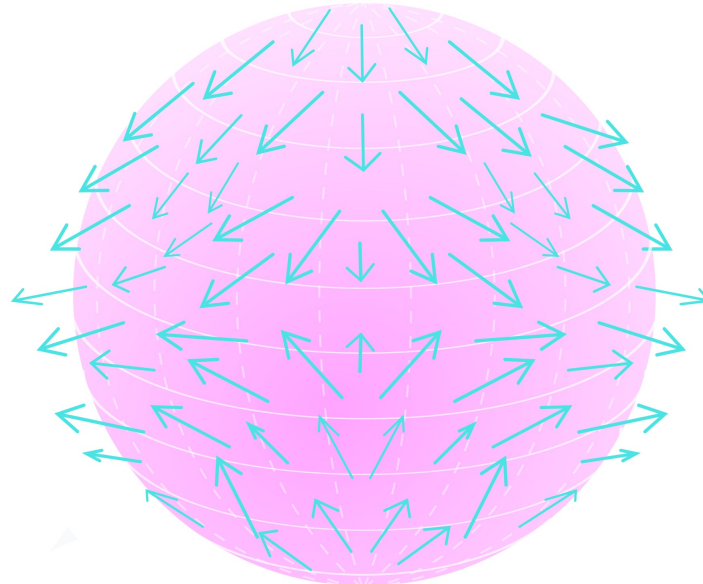
- Does not depend on choice of  $\alpha$
- Linear map

*Recall:*

# Gradient Vector Field

$\nabla f : S \rightarrow \mathbb{R}^3$  with

$$\begin{cases} \langle (\nabla f)(p), v \rangle = (df)_p(v), v \in T_p S \\ \langle (\nabla f)(p), N(p) \rangle = 0 \end{cases}$$

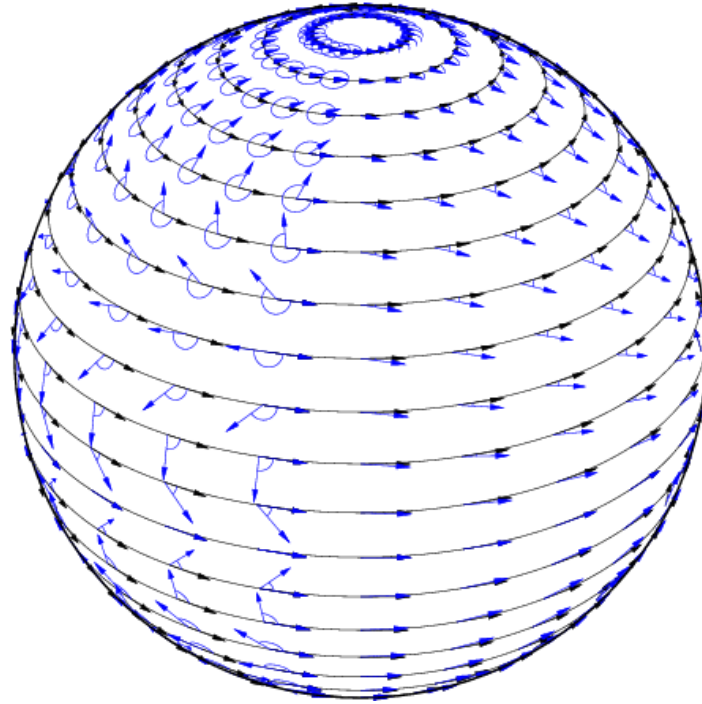


***How do you  
differentiate  
a vector field?***

Answer

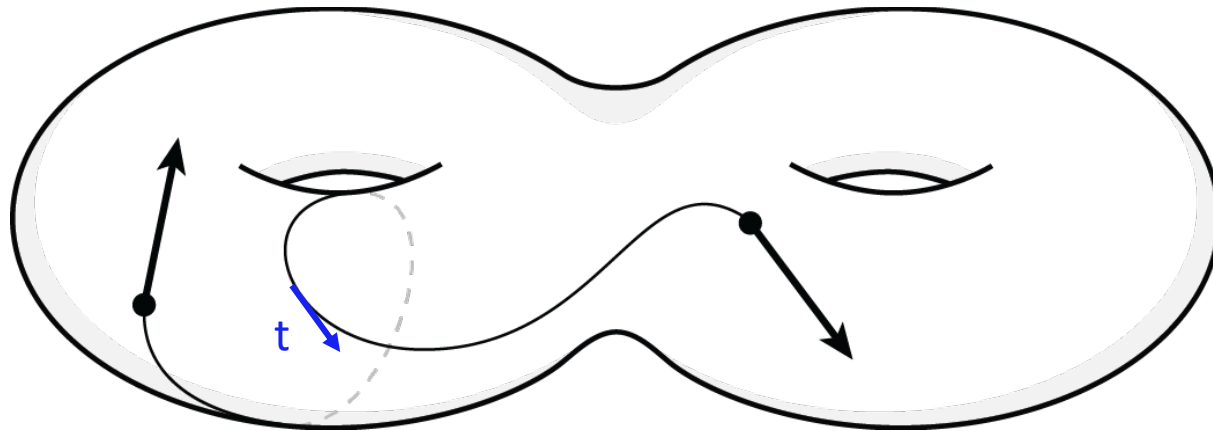


# What's the issue?



## What's a 'constant' VF on a surface?

# What's the issue?



***How to identify different  
tangent spaces?***

# *Many* Notions of Derivative

- ***Differential*** of covector  
(defer for now)

- ***Lie*** derivative

**Weak structure, easier to compute**

- ***Covariant*** derivative

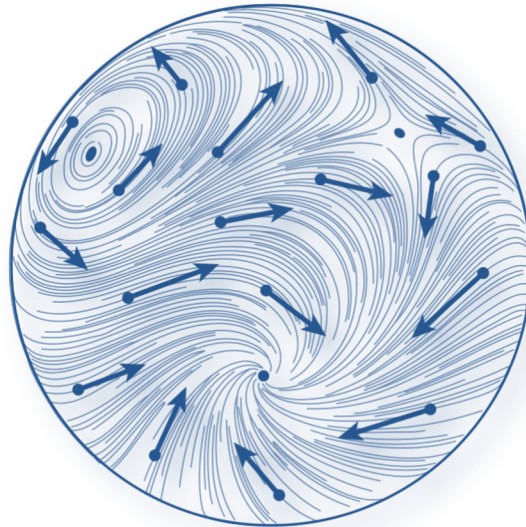
**Strong structure, harder to compute**



# Vector Field Flows: Diffeomorphism

$$\frac{d}{dt}\psi_t = V \circ \psi_t$$

Useful property:  $\psi_{t+s}(x) = \psi_t(\psi_s(x))$   
*Diffeomorphism with inverse  $\psi_{-t}$*



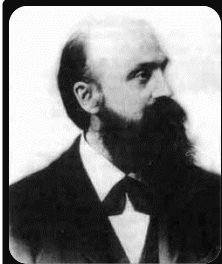
**Group  
structure!**

*Fun example:*

# Killing Vector Fields (KVF's)

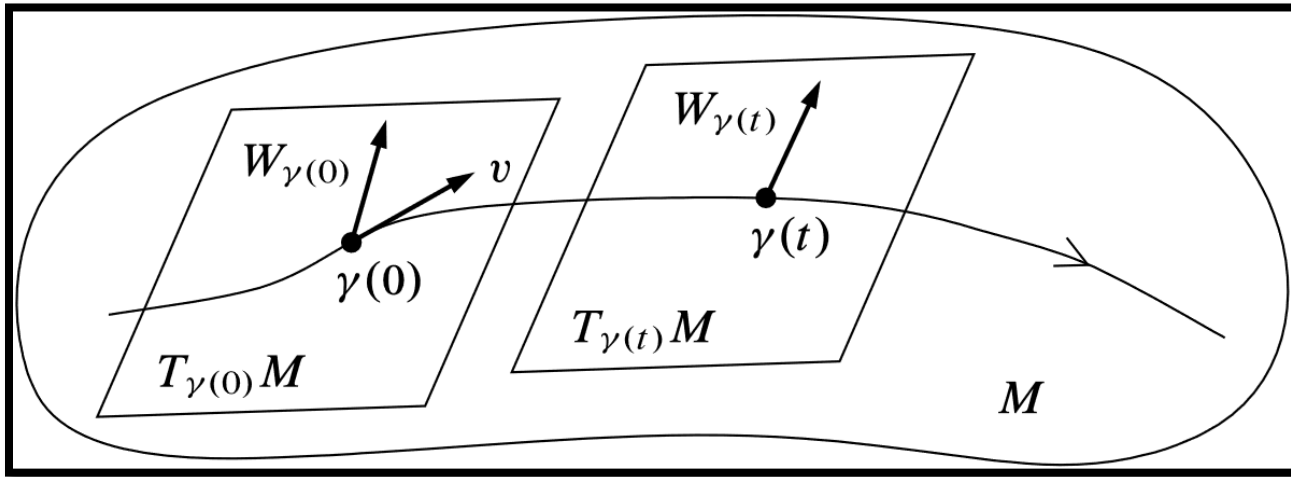


**Preserves  
distances  
infinitesimally**



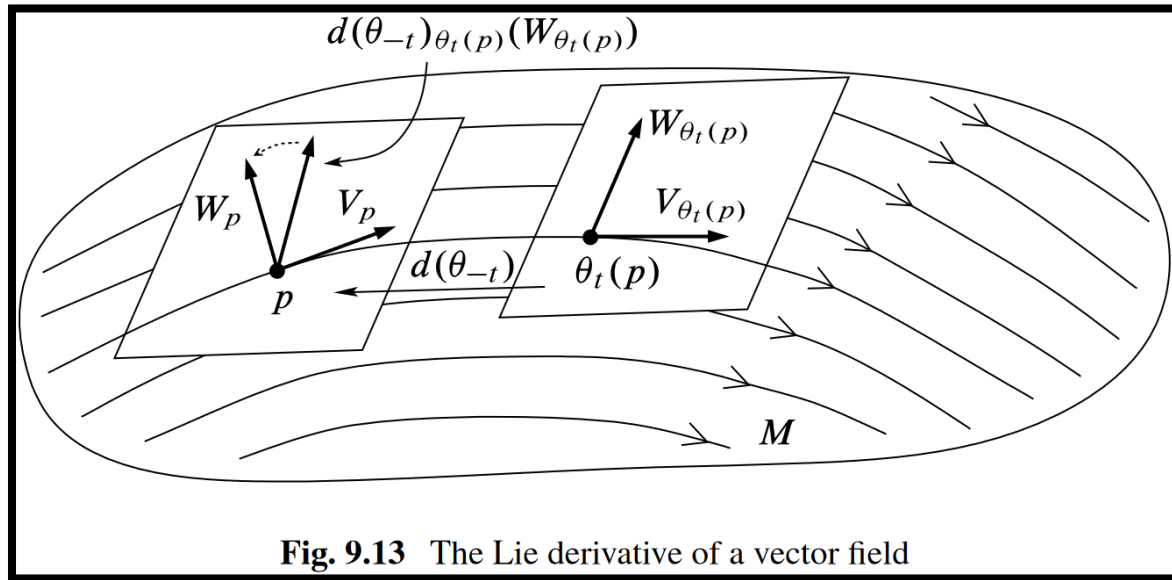
**Wilhelm Killing**  
1847-1923  
Germany

# Differential of Vector Field Flow



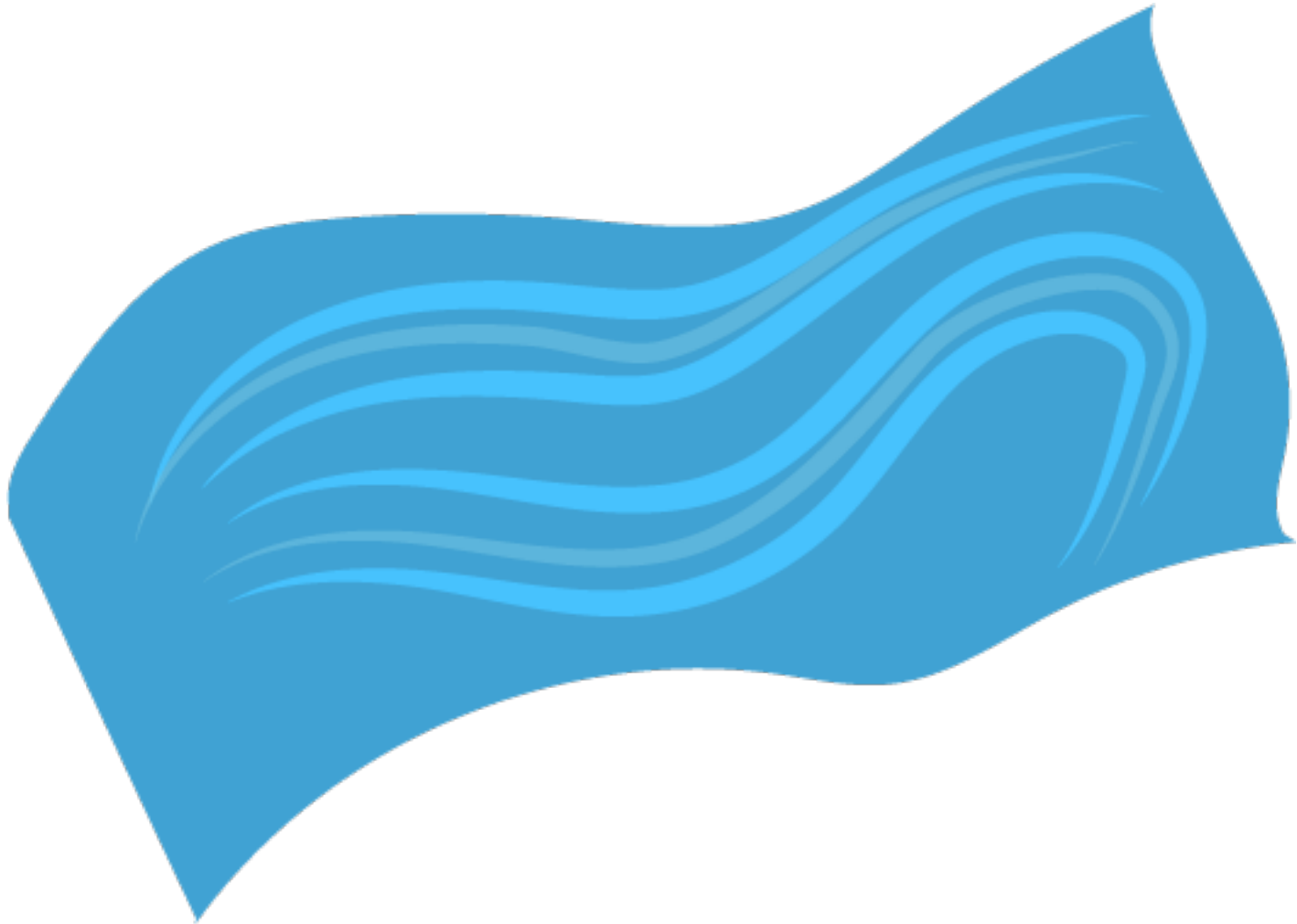
$$d\psi_t(p) : T_p M \rightarrow T_{\psi_t(p)} M$$

# Lie Derivative

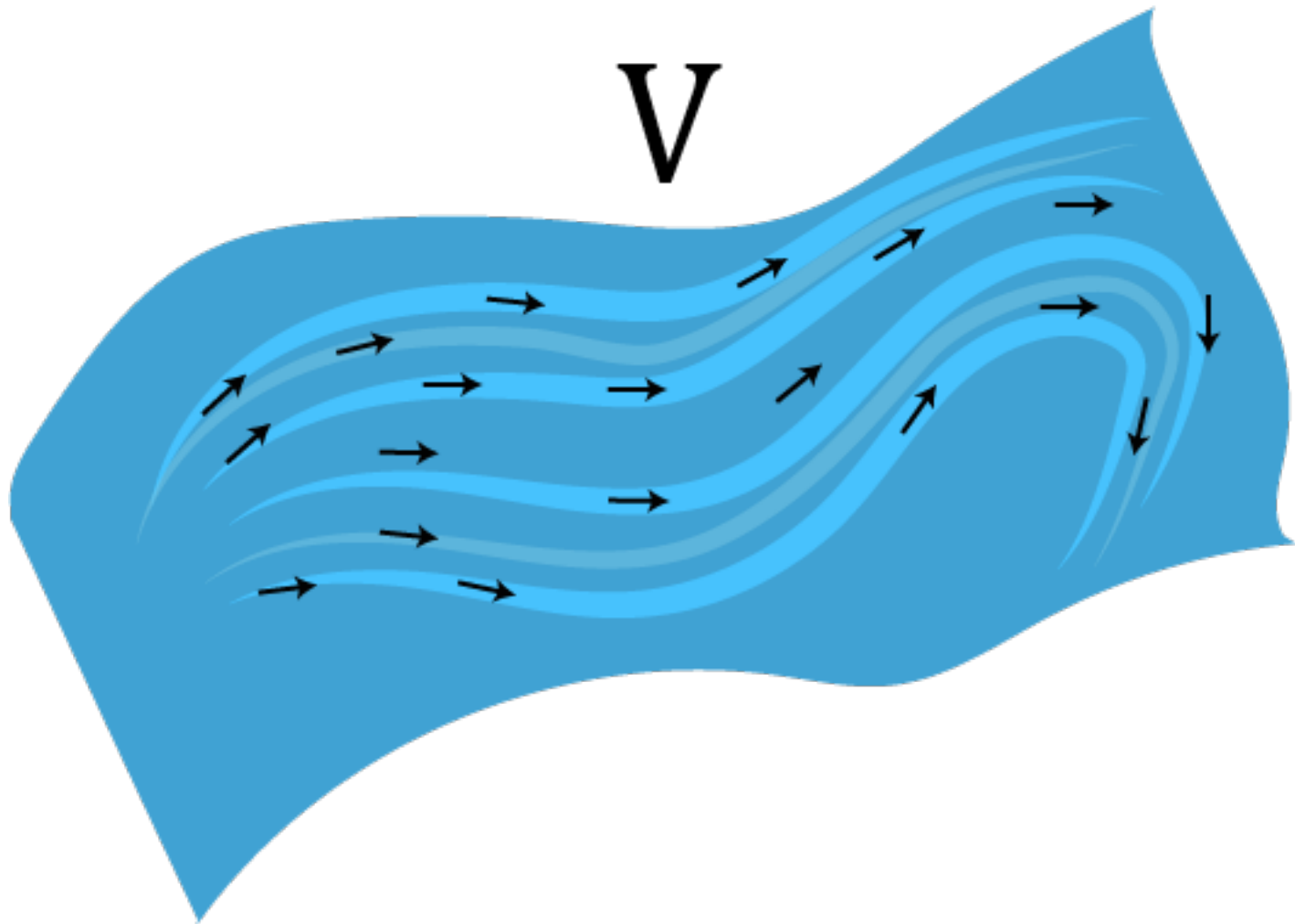


$$(\mathcal{L}_V W)_p := \lim_{t \rightarrow 0} \frac{1}{t} \left[ (d\psi_{-t})_{\psi_t(p)} (W_{\psi_t(p)}) - W_p \right]$$

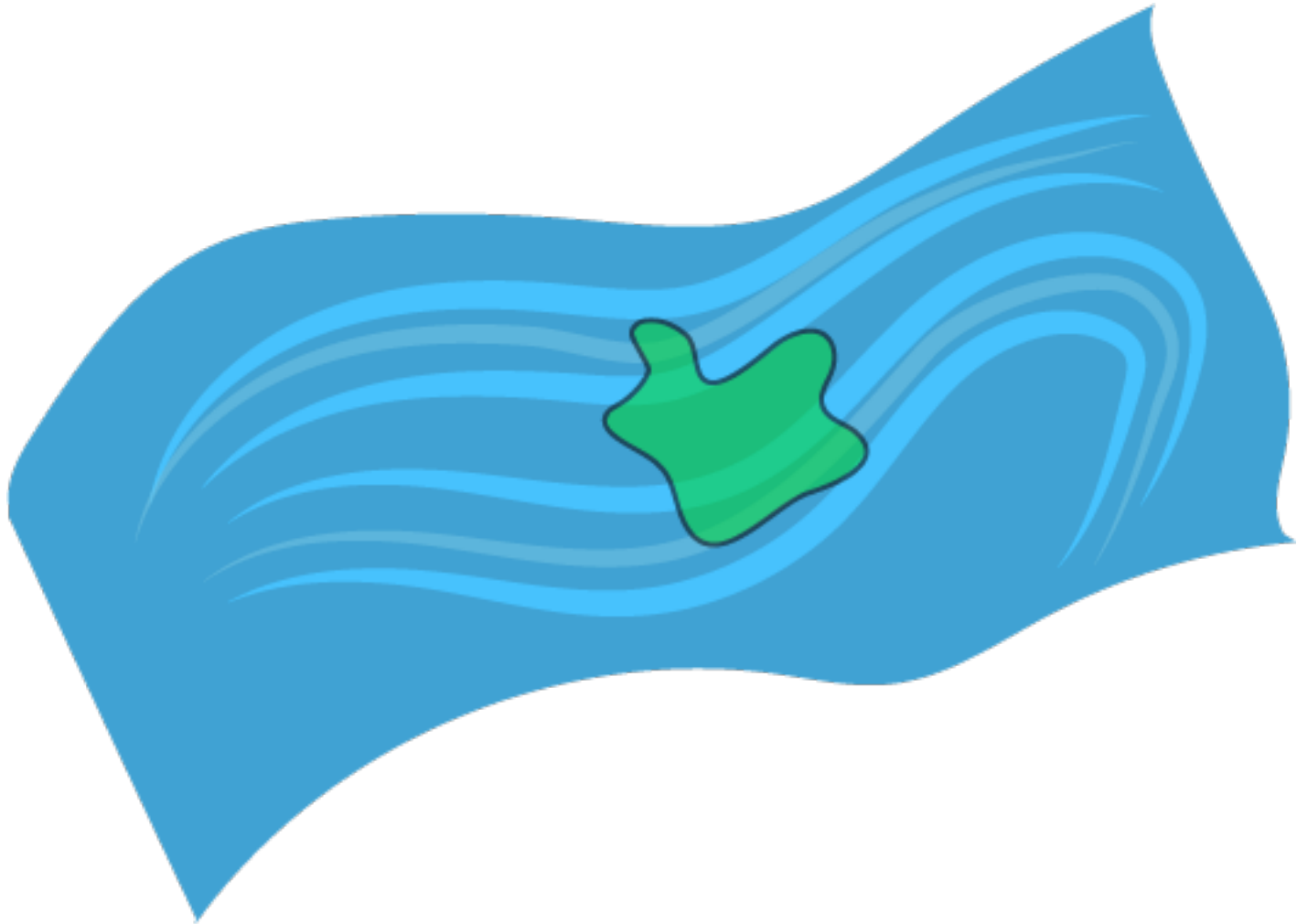
# Amoeba example



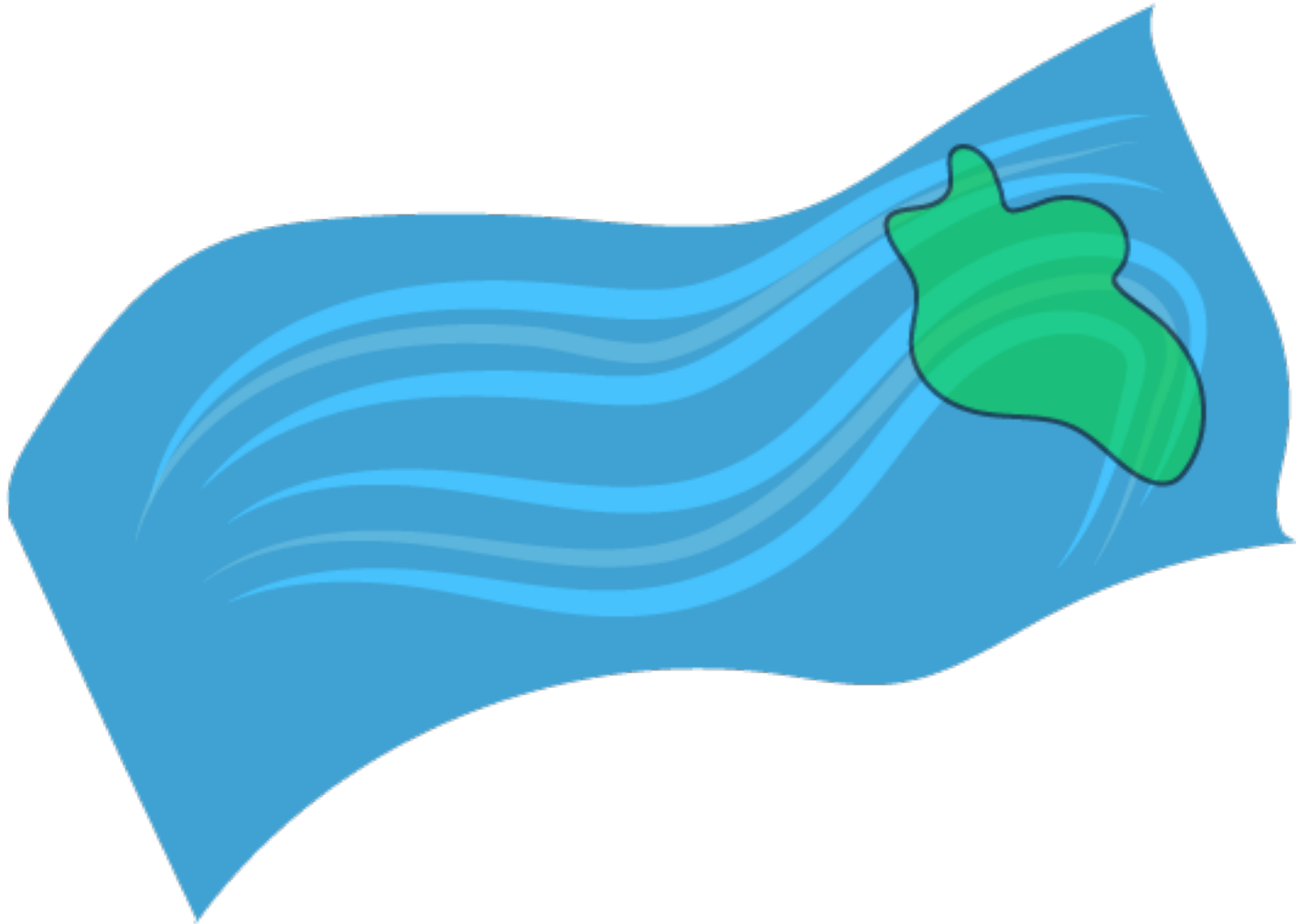
# Amoeba example



# Amoeba example

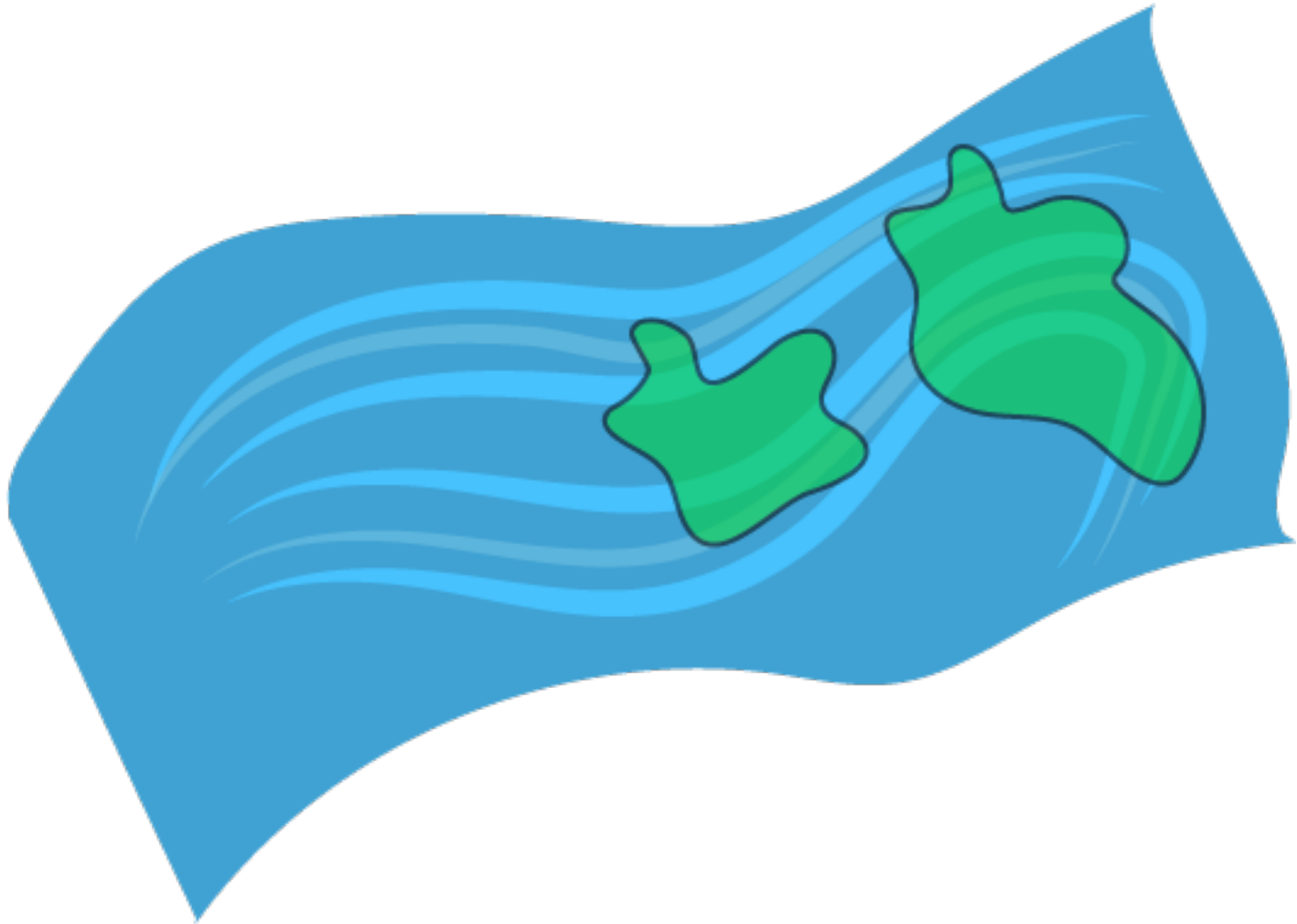


# Amoeba example

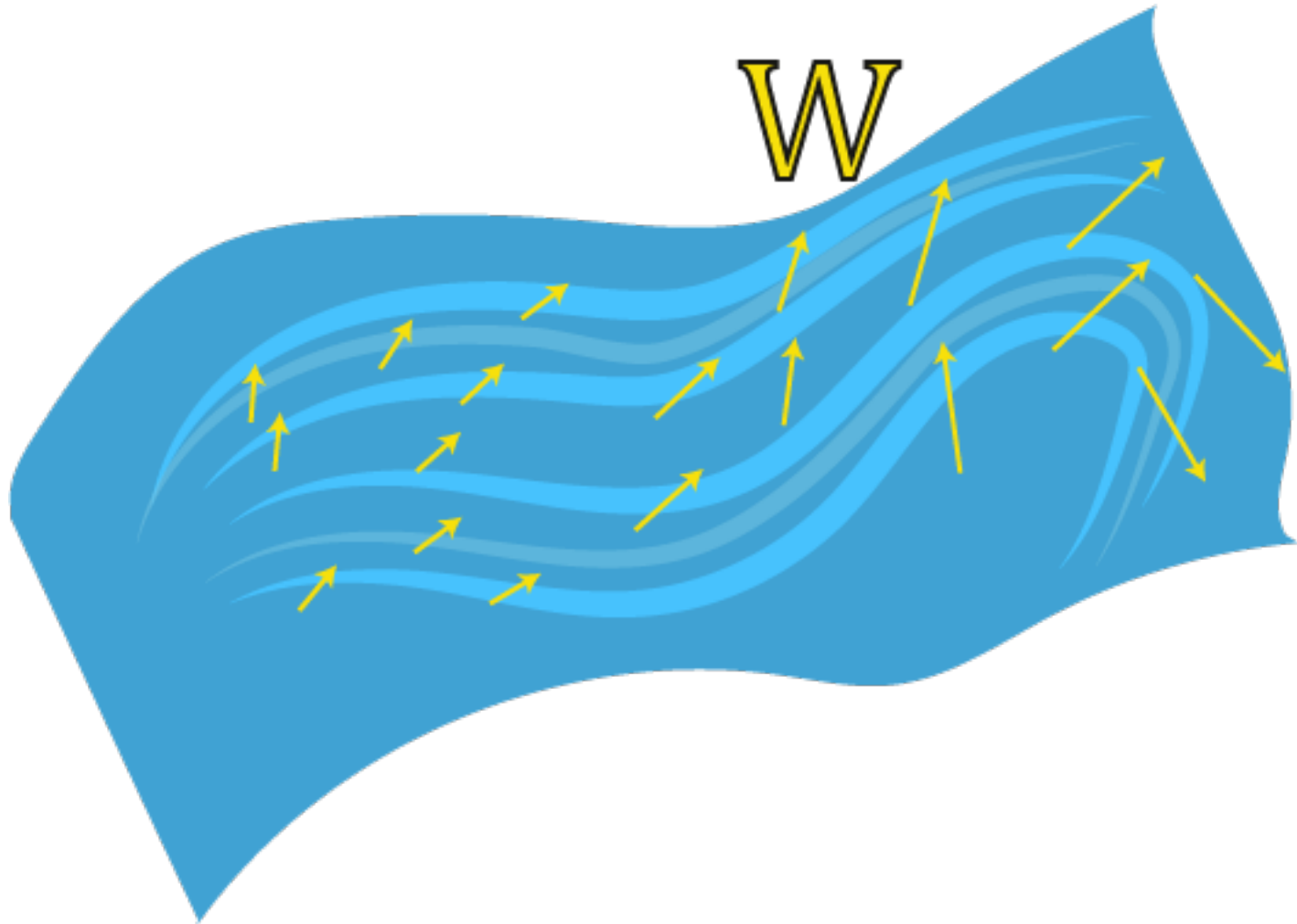




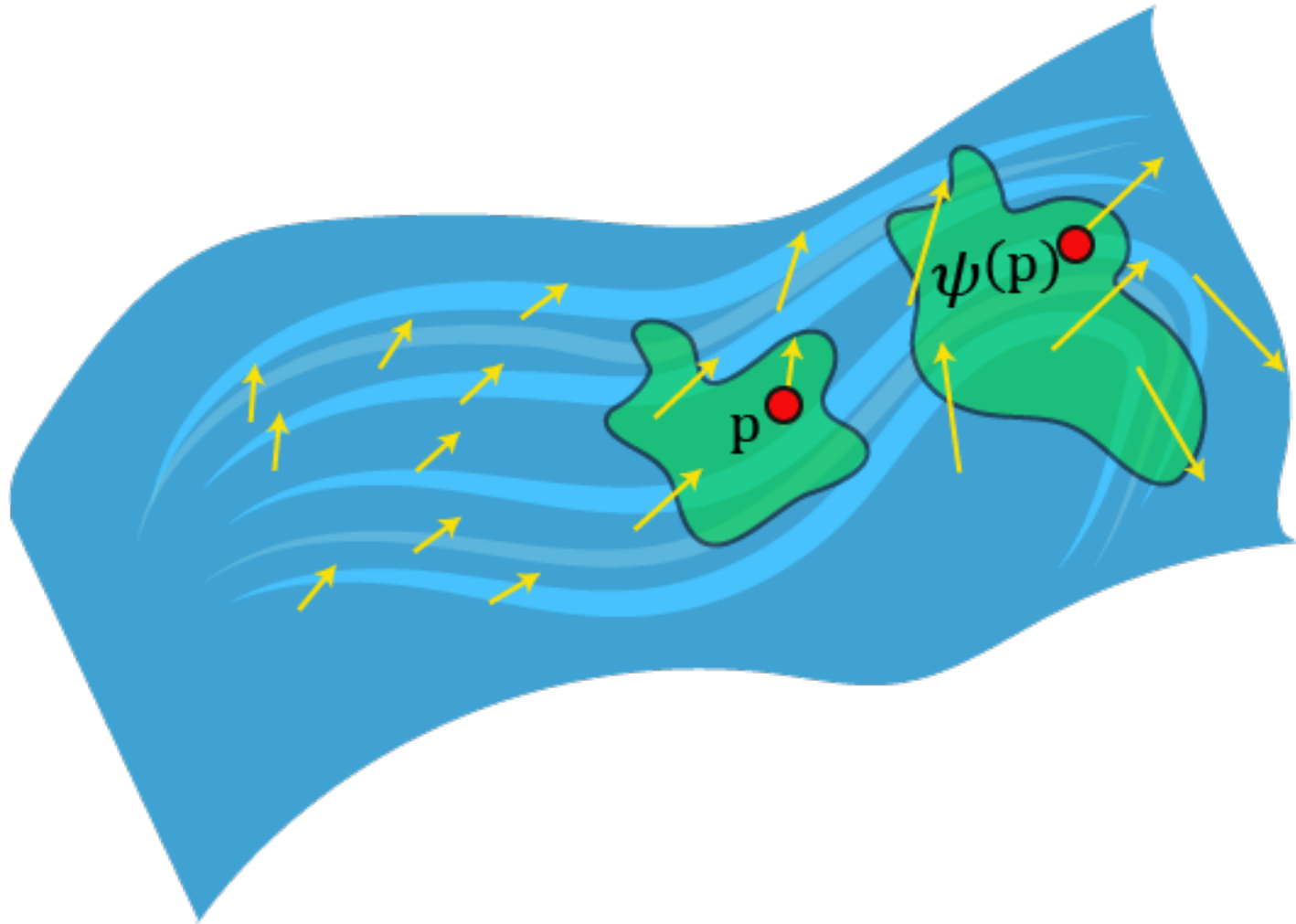
# Amoeba example



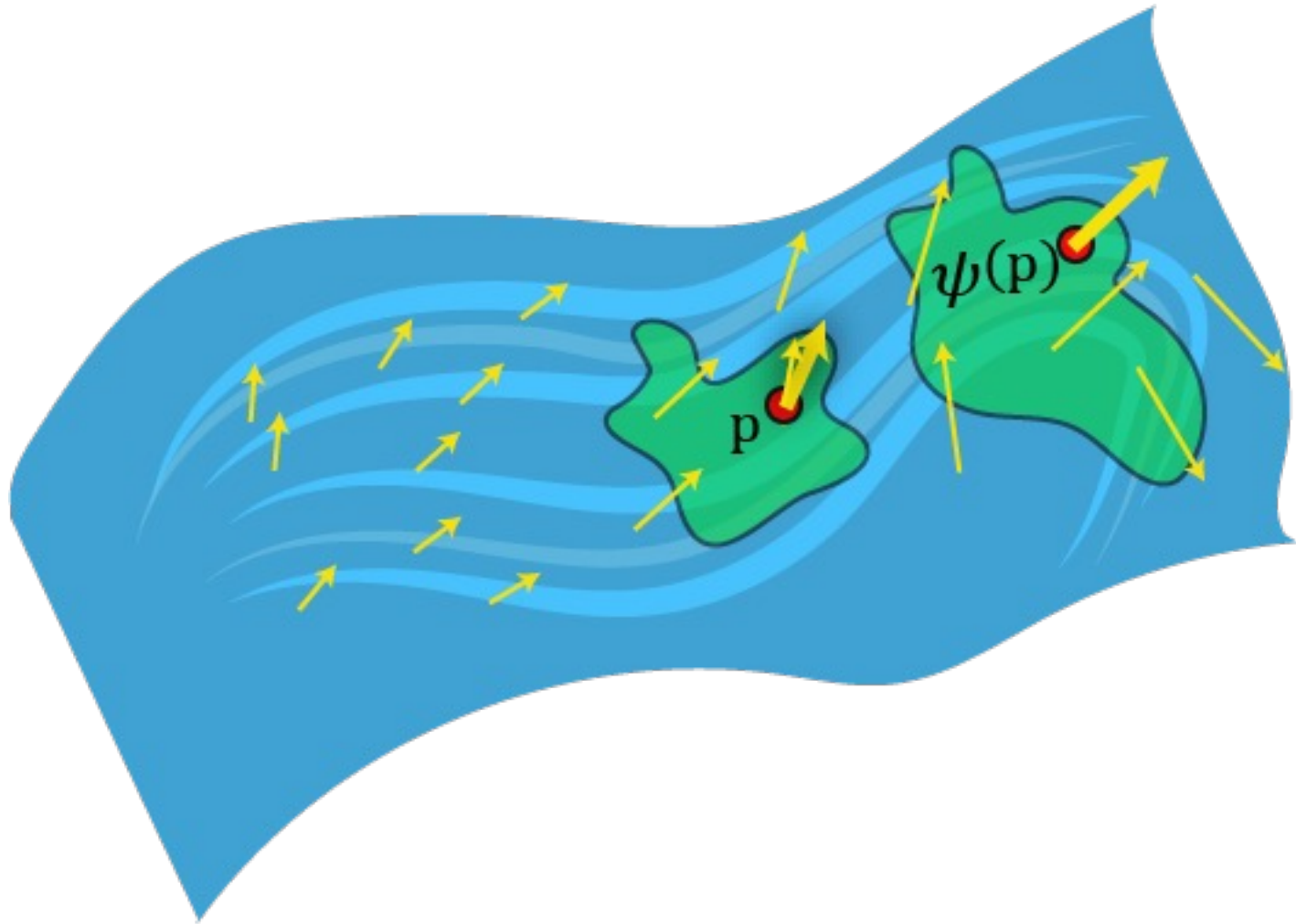
# Amoeba example



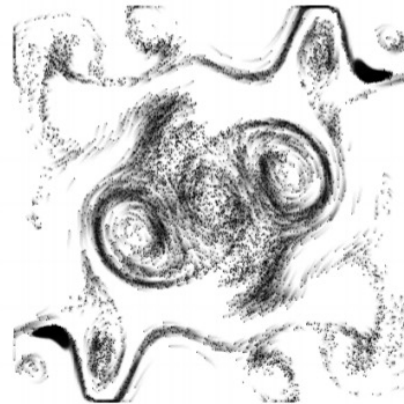
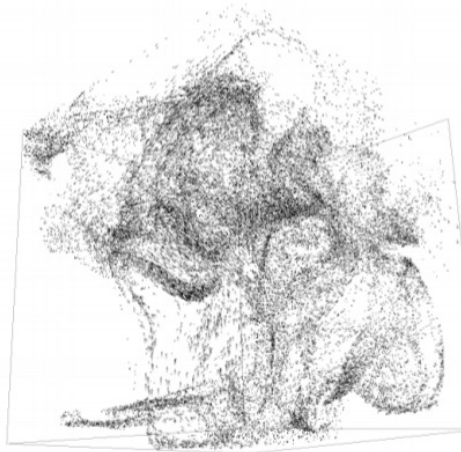
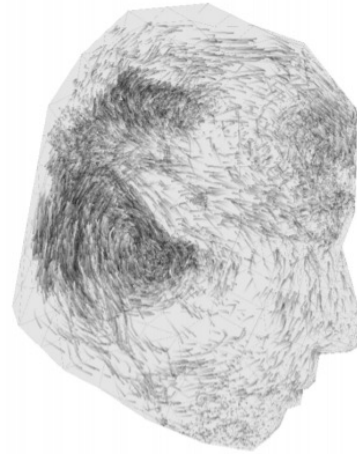
# Amoeba example



# Amoeba example



# Fluid Simulation



# Navier-Stokes Equation

*for vorticity*

Liquid velocity:  $u$

Vorticity:  $\omega = \text{curl } u$

Vector  
Laplacian

$$\dot{\omega} = \text{Adv}(u, \omega) + \nu \Delta \omega + \text{curl } f$$

Advection  
term

External  
forces

$$\text{Adv}(u, \omega) := \text{curl } \omega \times u = \mathcal{L}_u \omega$$

How to discretize?

$$\text{Adv}(u, \omega) := \text{curl } \omega \times u = \mathcal{L}_u \omega$$

How to discretize?

**FEM!**

**Basis?**



# Laplacian eigenfunctions!

We want:

$$\begin{aligned}\Delta\Phi_k &= \lambda_k \Phi_k \\ \operatorname{div} \Phi_k &= 0 \\ \Phi_k \cdot n &= 0 \text{ at } \partial D\end{aligned}$$

Then our velocity field  $u$  is represented as

$$u = \sum \omega_i \Phi_i$$

Requirements on basis functions

# Laplacian eigenfunctions!

$$u = \sum \omega_i \Phi_i$$

Then (***skipping computations***)

$$\omega = \operatorname{curl} u = \sum \omega_i \phi_i$$

where

$$\Delta \phi_i = \lambda_i \phi_i$$

$$\phi_i \times n = 0 \text{ at } \partial D$$

Vector  
Laplacian

$$\dot{\omega} = \text{Adv}(u, \omega) + \nu \Delta \omega + \text{curl } f$$

Advection  
term

External  
forces

$$\sum_k \dot{\omega}_k \phi_k = \sum_k \omega^T C_k \omega + \nu \Delta \omega + \text{curl } f$$

# Fluid Simulation using Laplacian Eigenfunctions

TYLER DE WITT, CHRISTIAN LESSIG and EUGENE FIUME

University of Toronto

We present an algorithm for the simulation of incompressible fluid phenomena that is computationally efficient and leads to visually convincing simulations with far fewer degrees of freedom than existing approaches. Rather than using an Eulerian grid or Lagrangian elements, we represent vorticity and velocity using a basis of global functions defined over the entire simulation domain. We show that choosing Laplacian eigenfunctions for this basis provides benefits, including correspondence with spatial scales of vorticity and precise energy control at each scale. We perform Galerkin projection of the Navier-Stokes equations to derive a time evolution equation in the space of basis coefficients. Our method admits closed form solutions on simple domains but can also be implemented efficiently on arbitrary meshes.

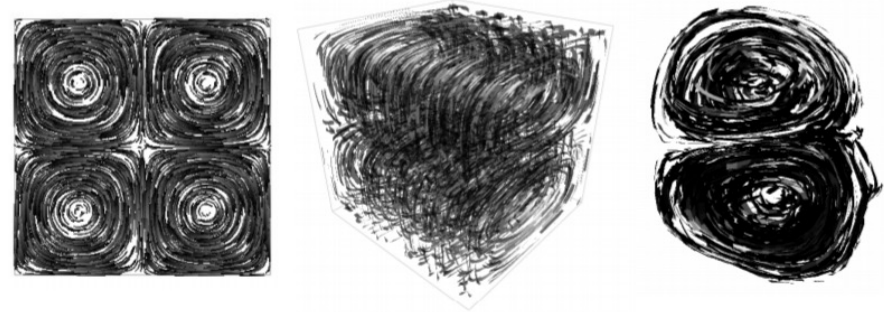


Fig. 1. Examples of Laplacian eigenfunction velocity basis fields on various domains. They are stationary solutions to the Navier-Stokes equations.



# What's Wrong with Lie Derivatives?

$$(\mathcal{L}_V W)_p := \lim_{t \rightarrow 0} \frac{1}{t} \left[ (d\psi_{-t})_{\psi_t(p)} (W_{\psi_t(p)}) - W_p \right]$$

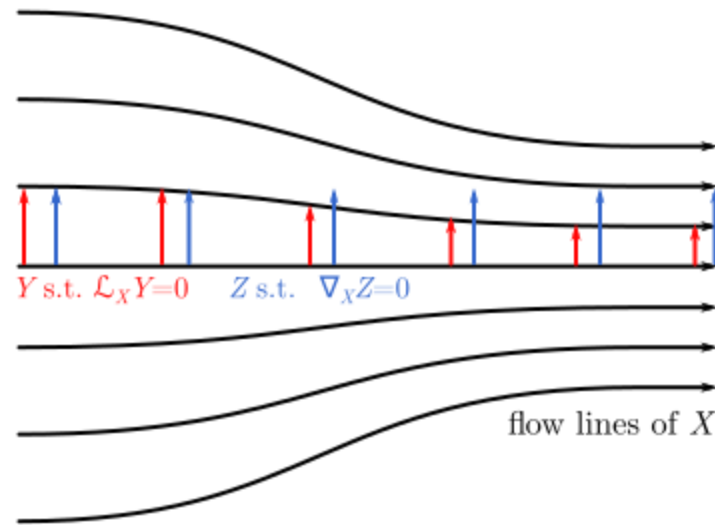
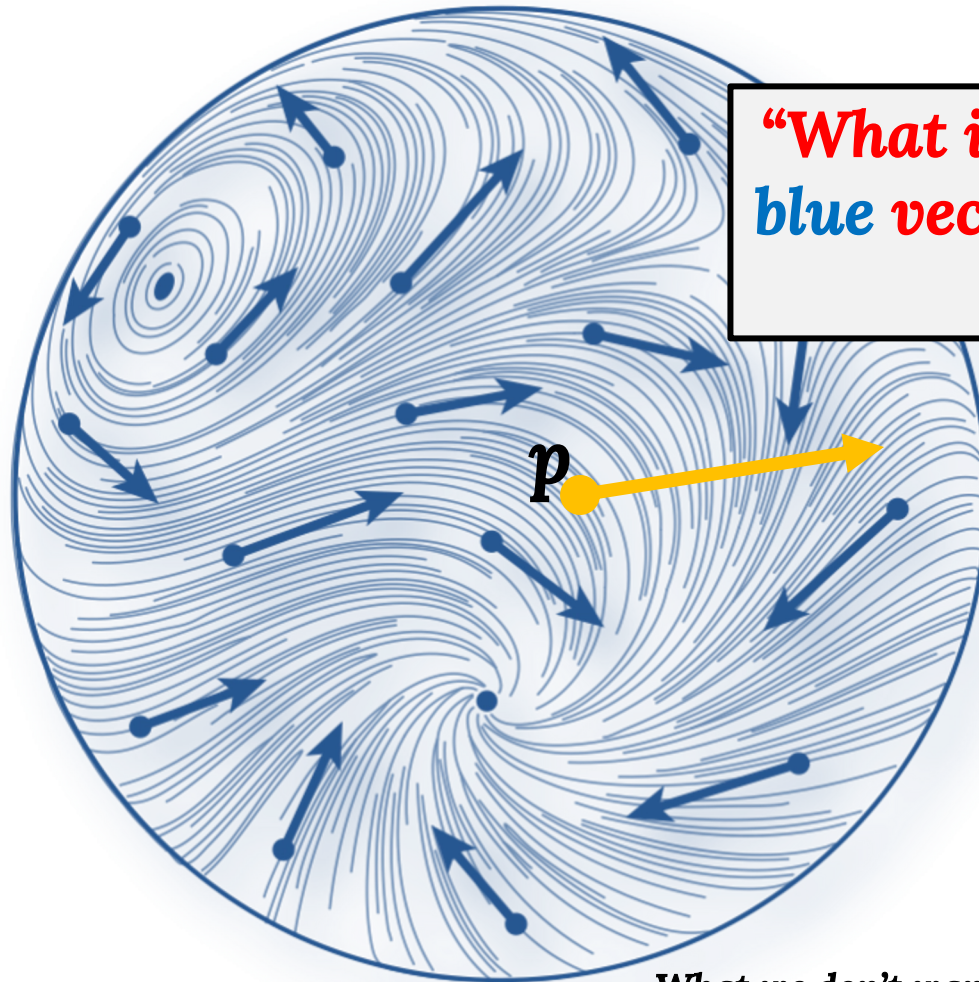


Image courtesy A. Carape

**Depends on structure of V**

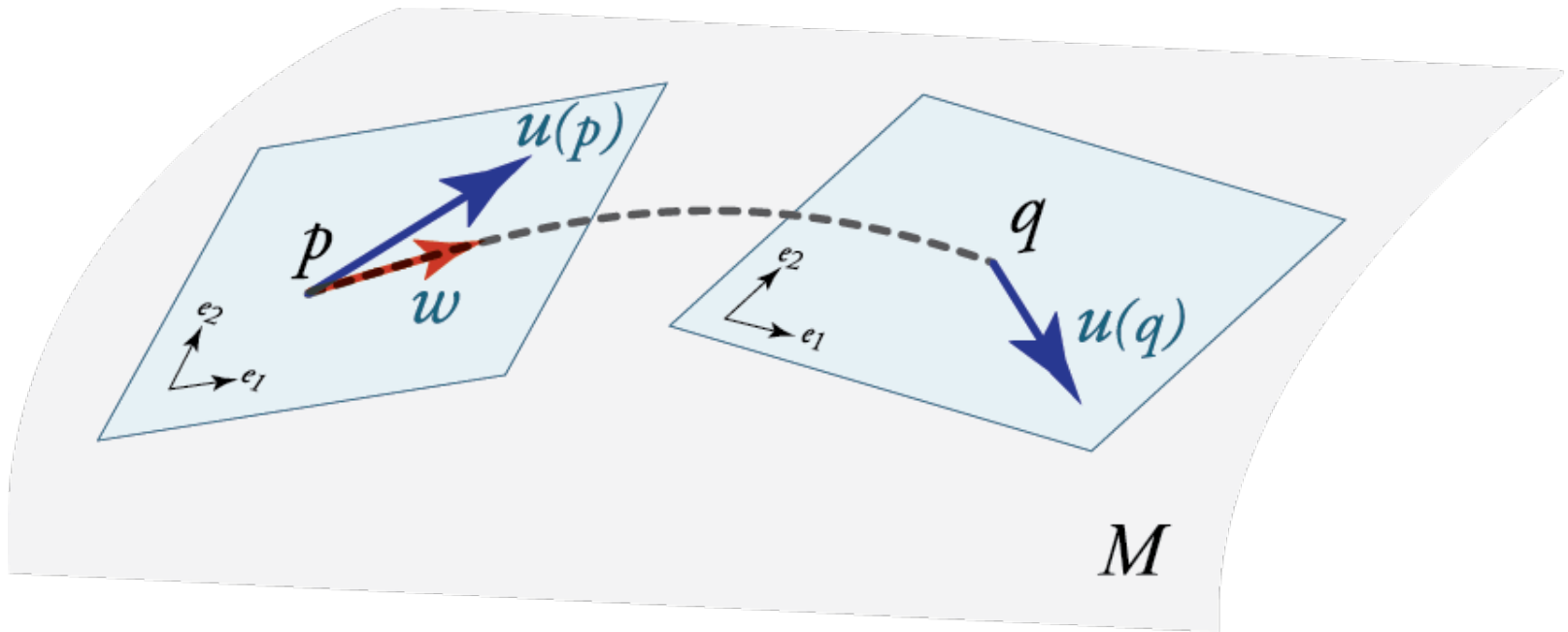
# What We Want



**“What is the derivative of the blue vector field in the orange direction?”**

What we don't want:  
Specify **orange** direction anywhere but at  $p$ .

# Parallel Transport



**Canonical identification of  
tangent spaces**

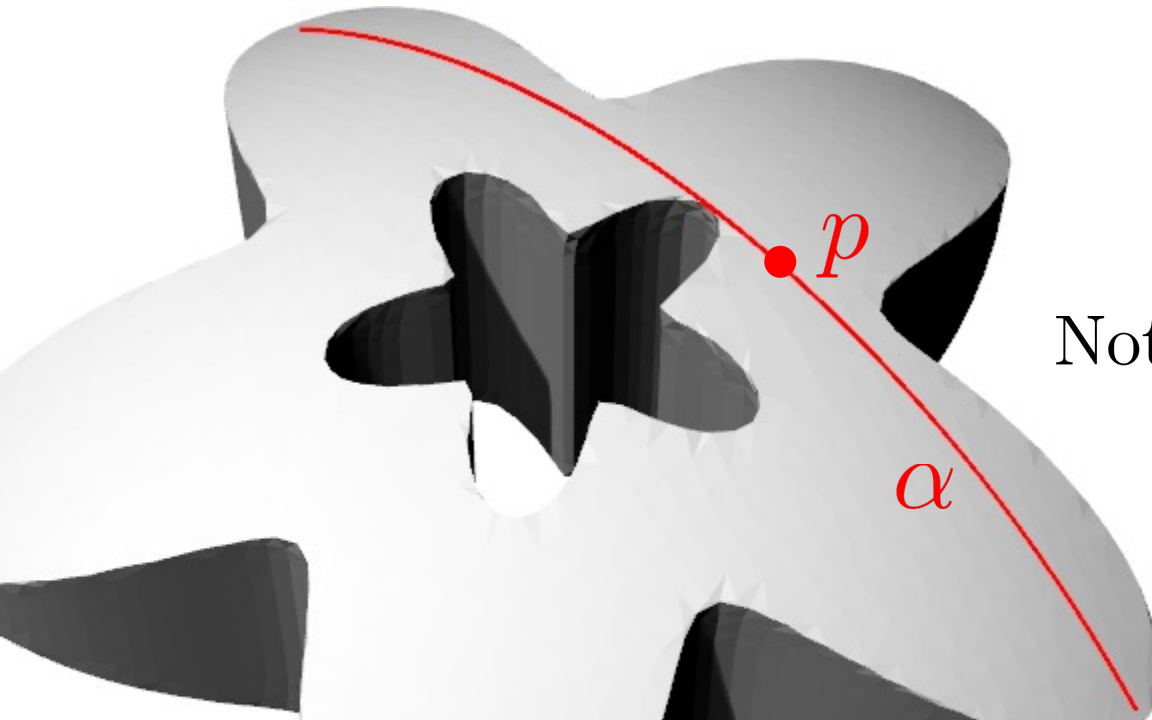


# Covariant Derivative (Embedded)

$$\nabla_V W := [D_V W]^\parallel = \text{proj}_{T_p S} (W \circ \alpha)'(0)$$

Integral curve of  $V$  through  $p$

**Synonym:** (Levi-Civita) Connection



Note:  $[D_V W]^\perp = \mathbb{I}(V, W)N$

# Some Properties

## Properties of the Covariant Derivative

As defined,  $\nabla_V Y$  depends only on  $V_p$  and  $Y$  to first order along  $c$ .

Also, we have the **Five Properties**:

1.  $C^\infty$ -linearity in the  $V$ -slot:

$$\nabla_{V_1 + fV_2} Y = \nabla_{V_1} Y + f \nabla_{V_2} Y \text{ where } f : S \rightarrow \mathbb{R}$$

2.  $\mathbb{R}$ -linearity in the  $Y$ -slot:

$$\nabla_V (Y_1 + aY_2) = \nabla_V Y_1 + a \nabla_V Y_2 \text{ where } a \in \mathbb{R}$$

3. Product rule in the  $Y$ -slot:

$$\nabla_V (f Y) = f \cdot \nabla_V Y + (\nabla_V f) \cdot Y \text{ where } f : S \rightarrow \mathbb{R}$$

4. The metric compatibility property:

$$\nabla_V \langle Y, Z \rangle = \langle \nabla_V Y, Z \rangle + \langle Y, \nabla_V Z \rangle$$

5. The “torsion-free” property:

$$\nabla_{V_1} V_2 - \nabla_{V_2} V_1 = [V_1, V_2]$$

### The Lie bracket

$$[V_1, V_2](f) := D_{V_1} D_{V_2}(f) - D_{V_2} D_{V_1}(f)$$

Defines a vector field, which is **tangent** to  $S$  if  $V_1, V_2$  are!

# Geodesic Equation

$$\text{proj}_{T_{\gamma(s)}S} [\gamma''(s)] = 0$$

- ***The only acceleration is out of the surface***
  - ***No steering wheel!***



# Intrinsic Geodesic Equation

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0$$

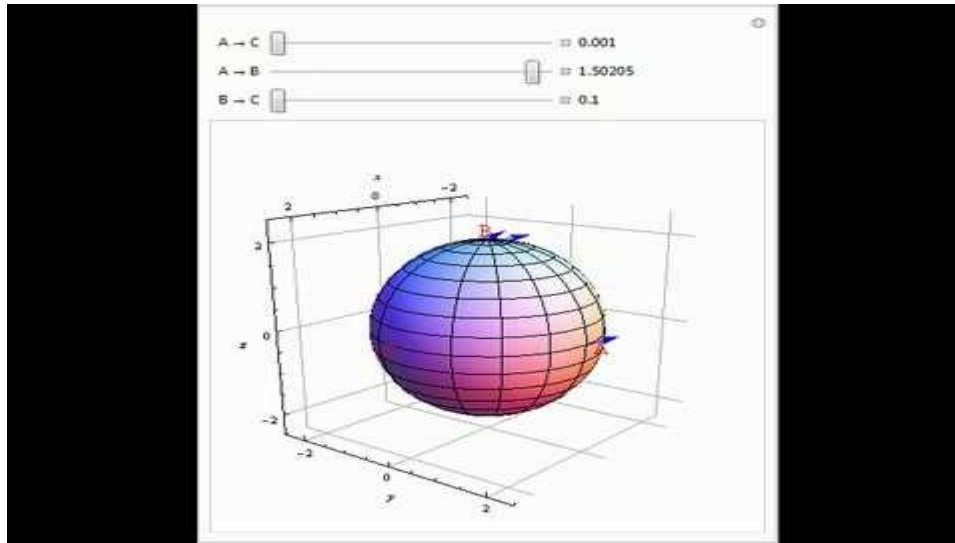
- ***No stepping on the accelerator***
  - ***No steering wheel!***



# Parallel Transport

Only path-independent if domain is flat.

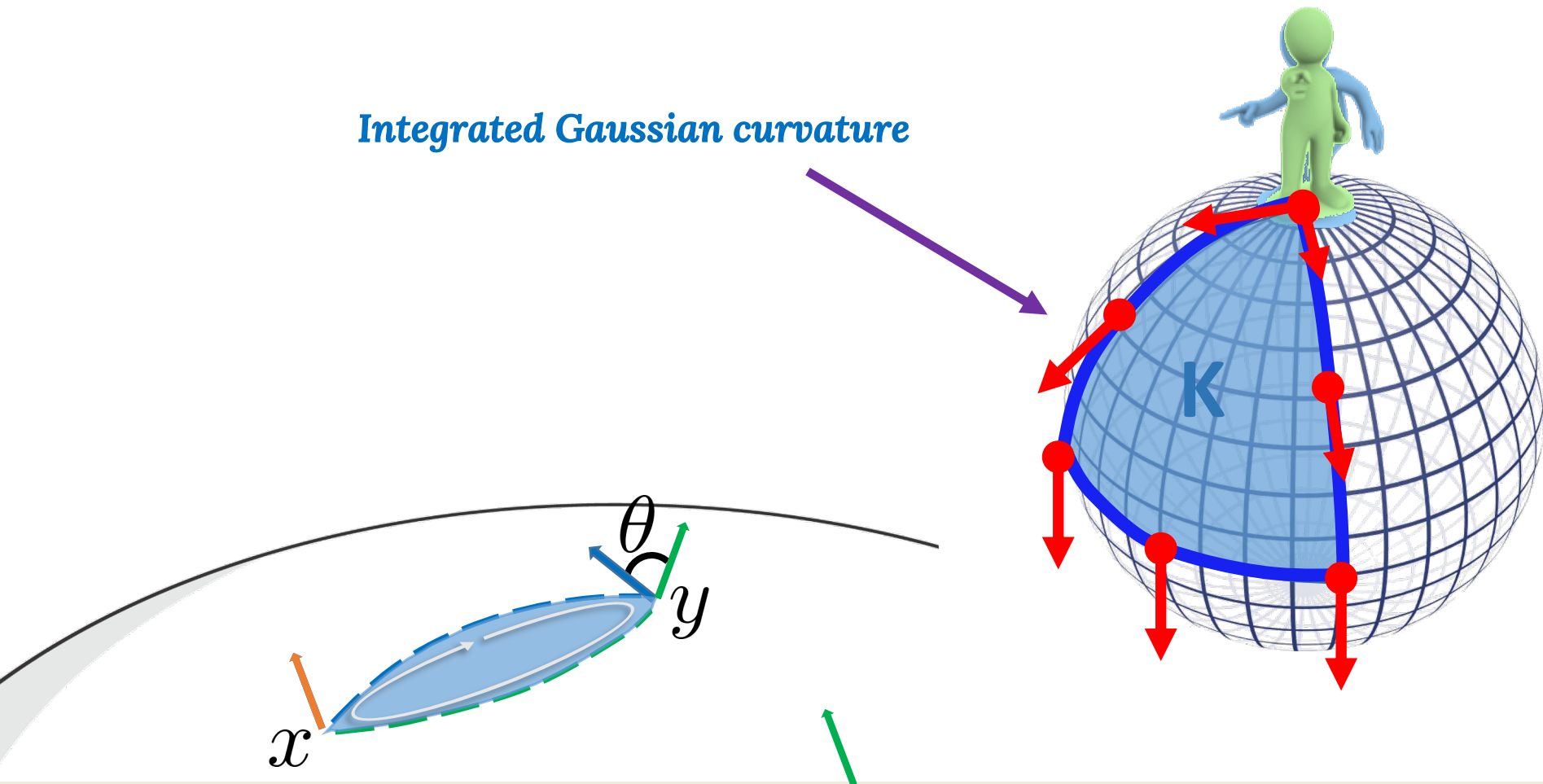
$$0 = \nabla_{\dot{\gamma}(t)} V$$



Preserves length, inner product  
(can be used to define covariant derivative)

# Holonomy

*Integrated Gaussian curvature*

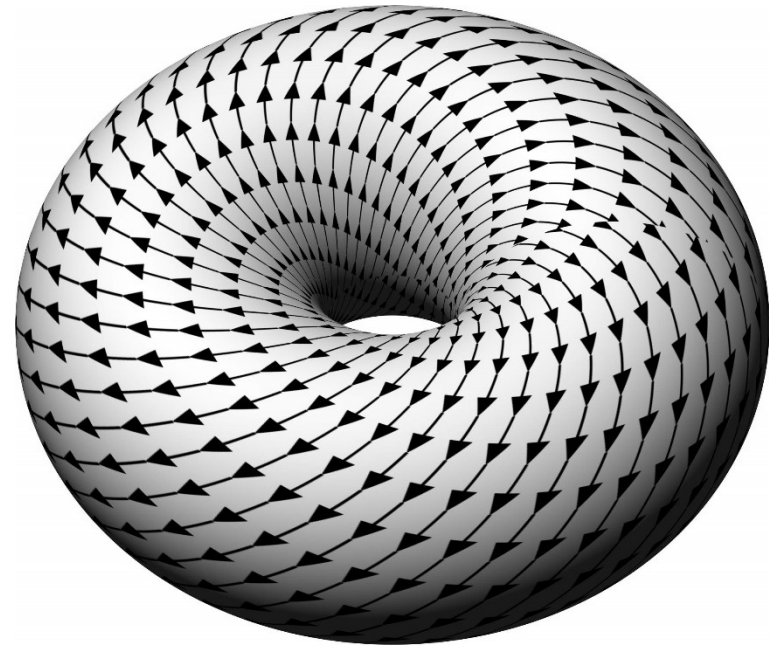


**Path dependence of parallel transport**

# Studying Vector fields

How to

- define a VF on a surface?
- Differentiate it?
- Integrate?
- ***Define its topology?***



# Vector Field Topology

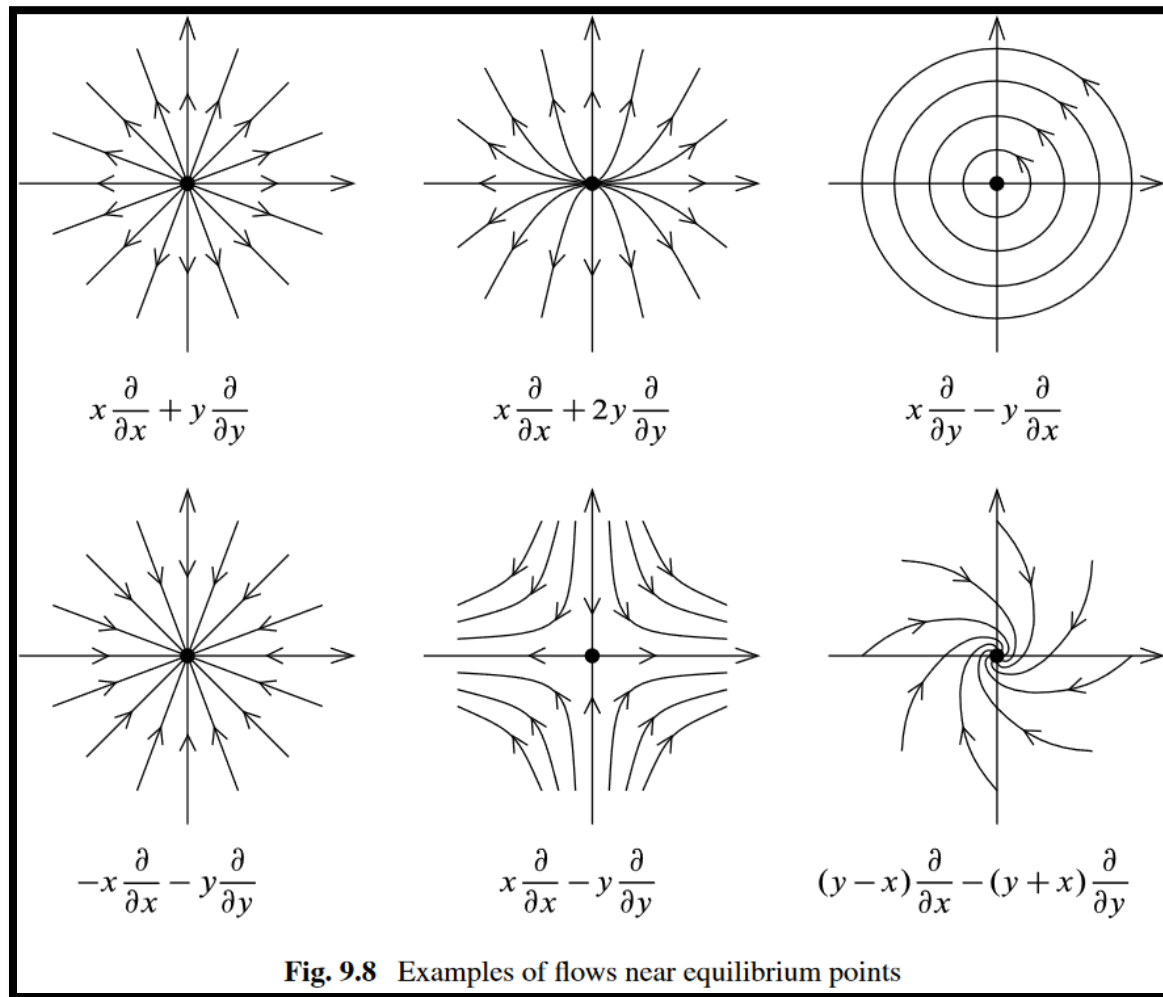
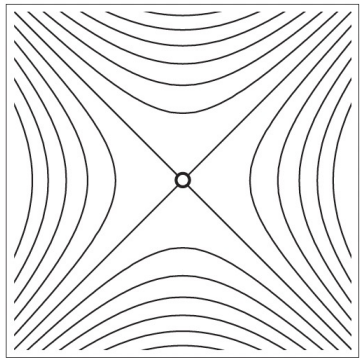


Fig. 9.8 Examples of flows near equilibrium points

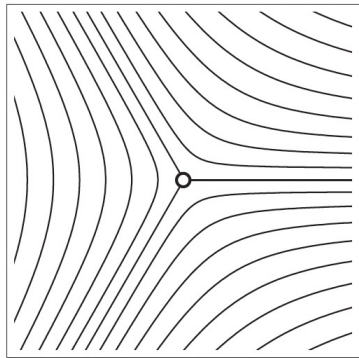
Image from **Smooth Manifolds**, Lee



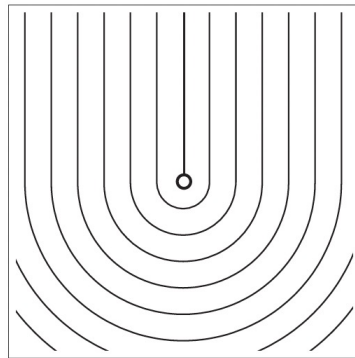
# Singularities



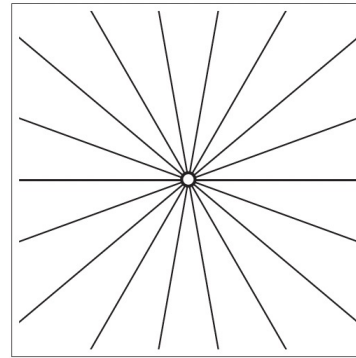
-1



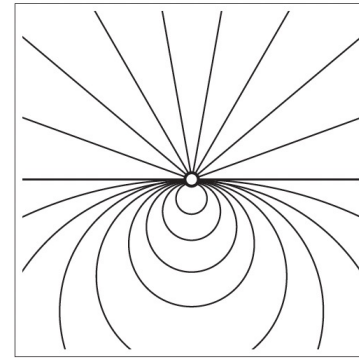
-1/2



1/2



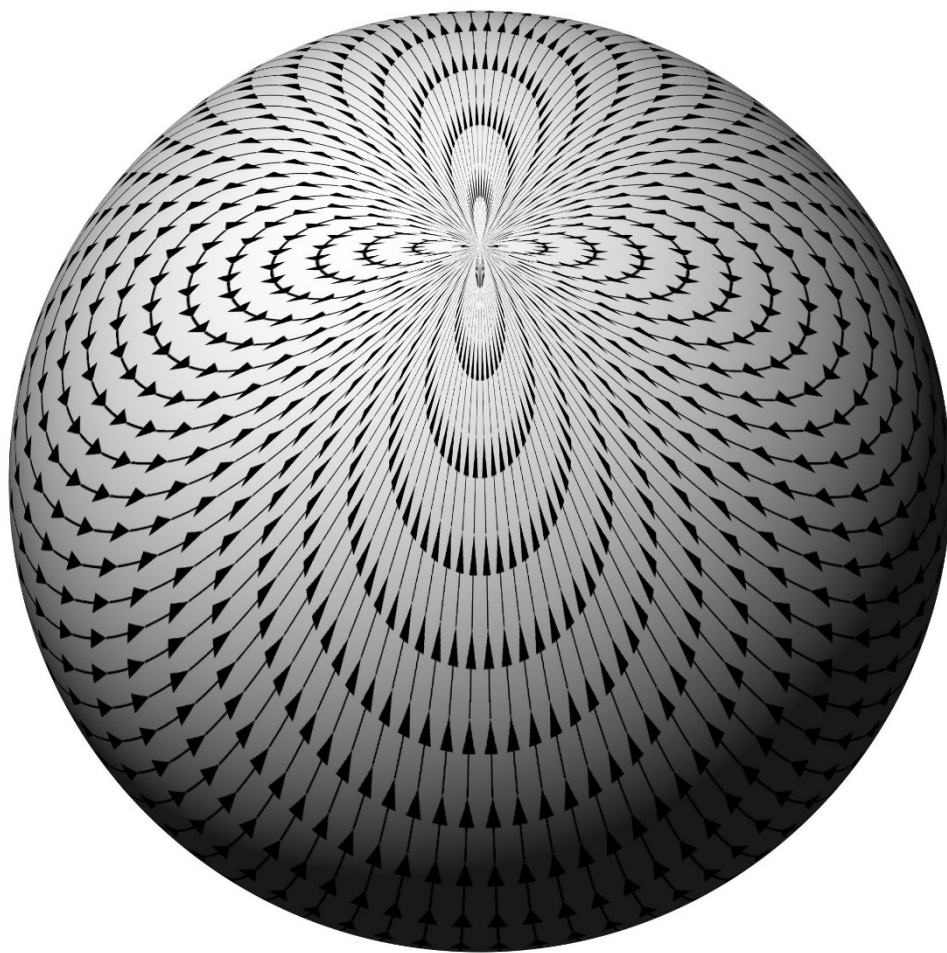
1



3/2

Image © Keenan Crane

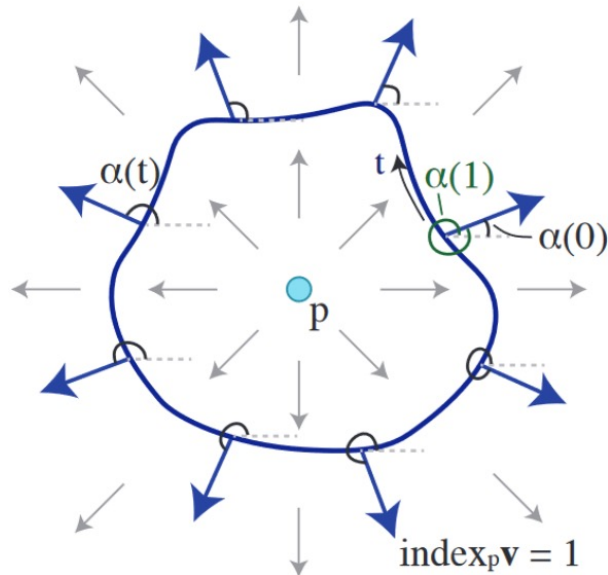
2



# Poincaré-Hopf Theorem

$$\sum_i \text{index}_{x_i}(v) = \chi(M)$$

where vector field  $v$  has isolated singularities  $\{x_i\}$ .



$$v(c(t)) = \|v(c(t))\| \begin{pmatrix} \cos \alpha(t) \\ \sin \alpha(t) \end{pmatrix}$$

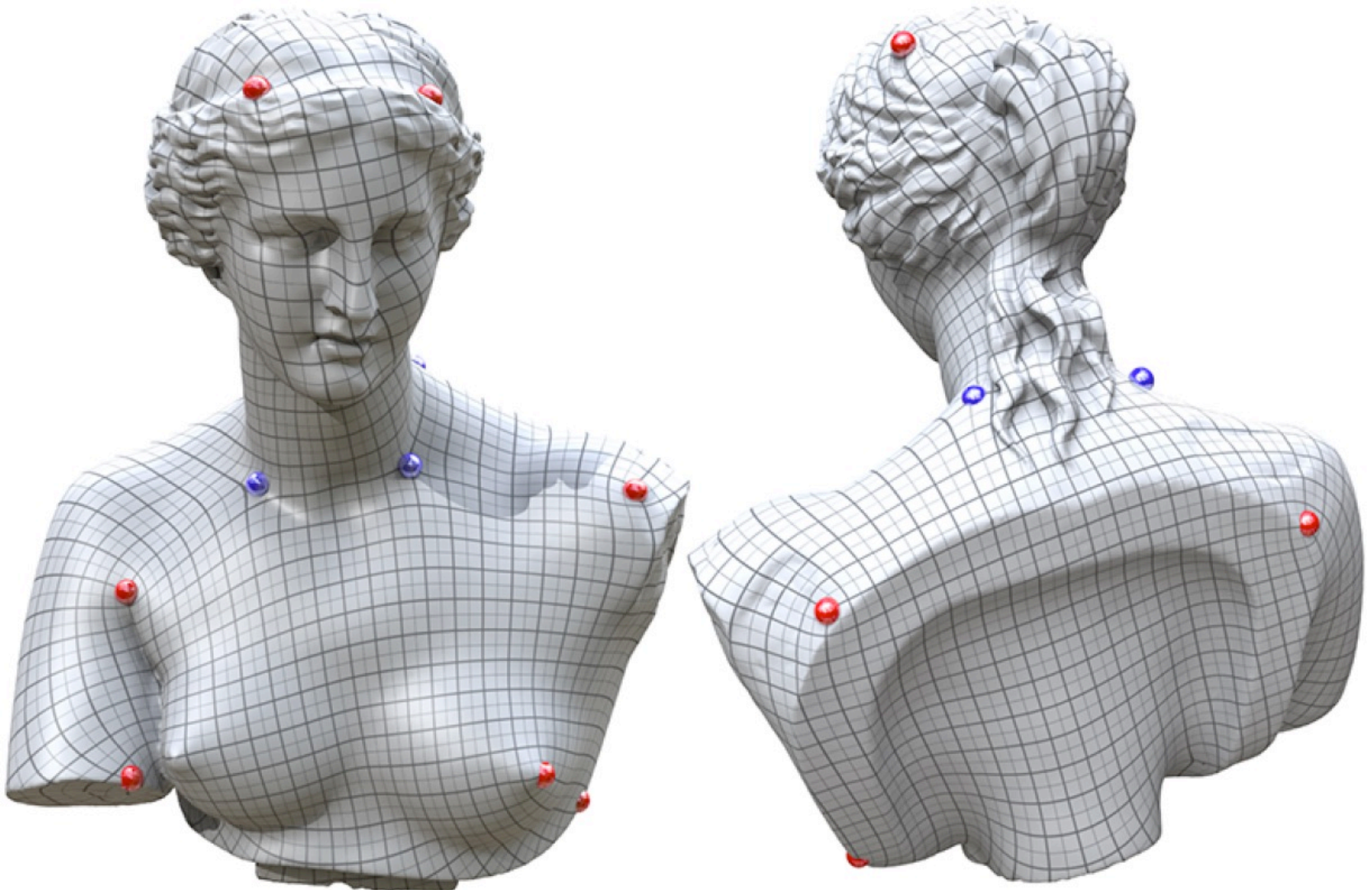
# Famous Corollary



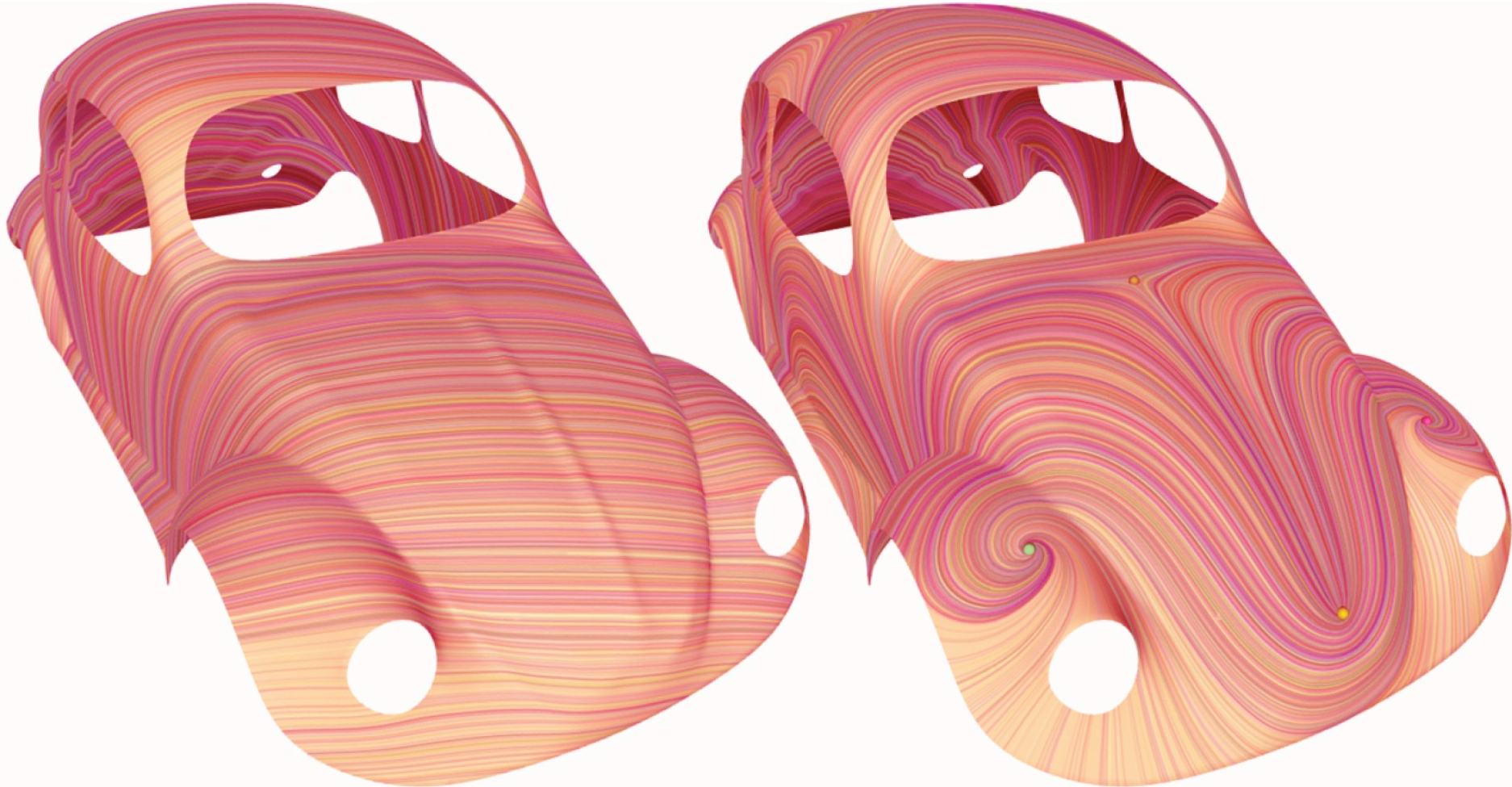
© Keenan Crane

***Hairy ball theorem***

# Singularities in wild



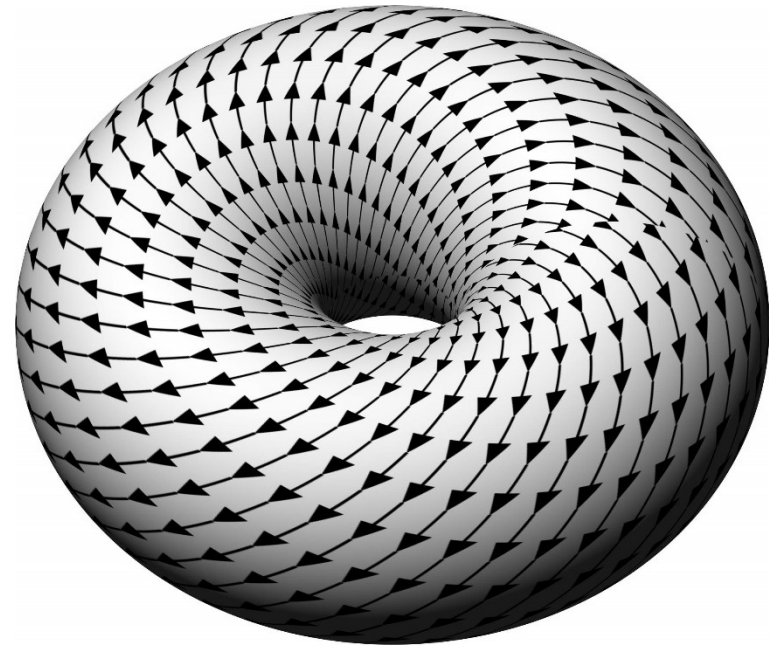
# Singularities in wild



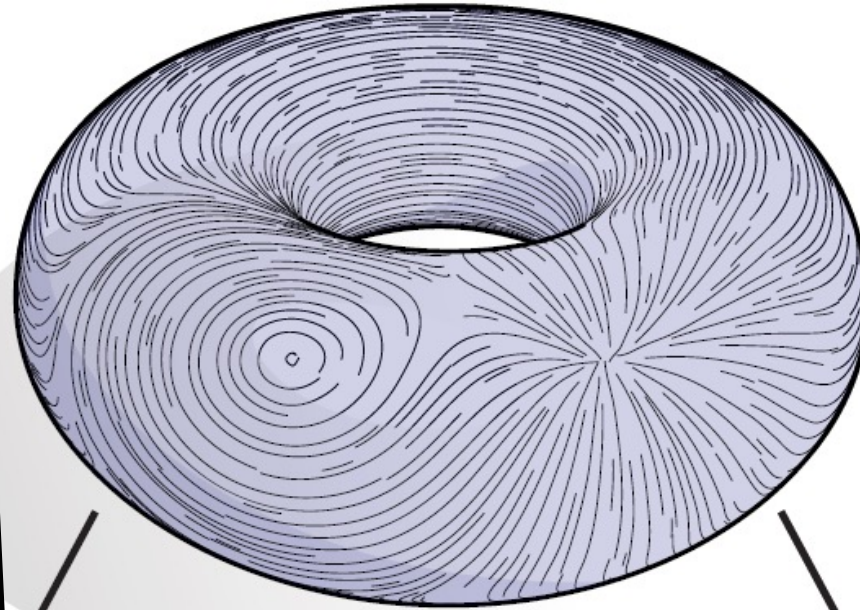
# Studying Vector fields

How to

- define a VF on a surface?
- Differentiate it?
- Integrate?
- Define its topology?
- **Characterize all possible VF?**

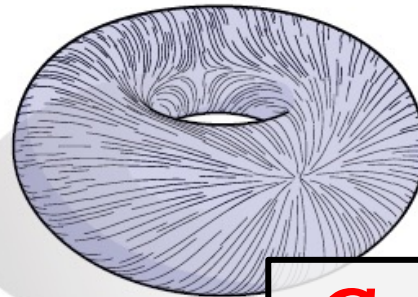


# Helmholtz-Hodge Decomposition



**Divergence  
free**

**Harmonic**



**Curl free**

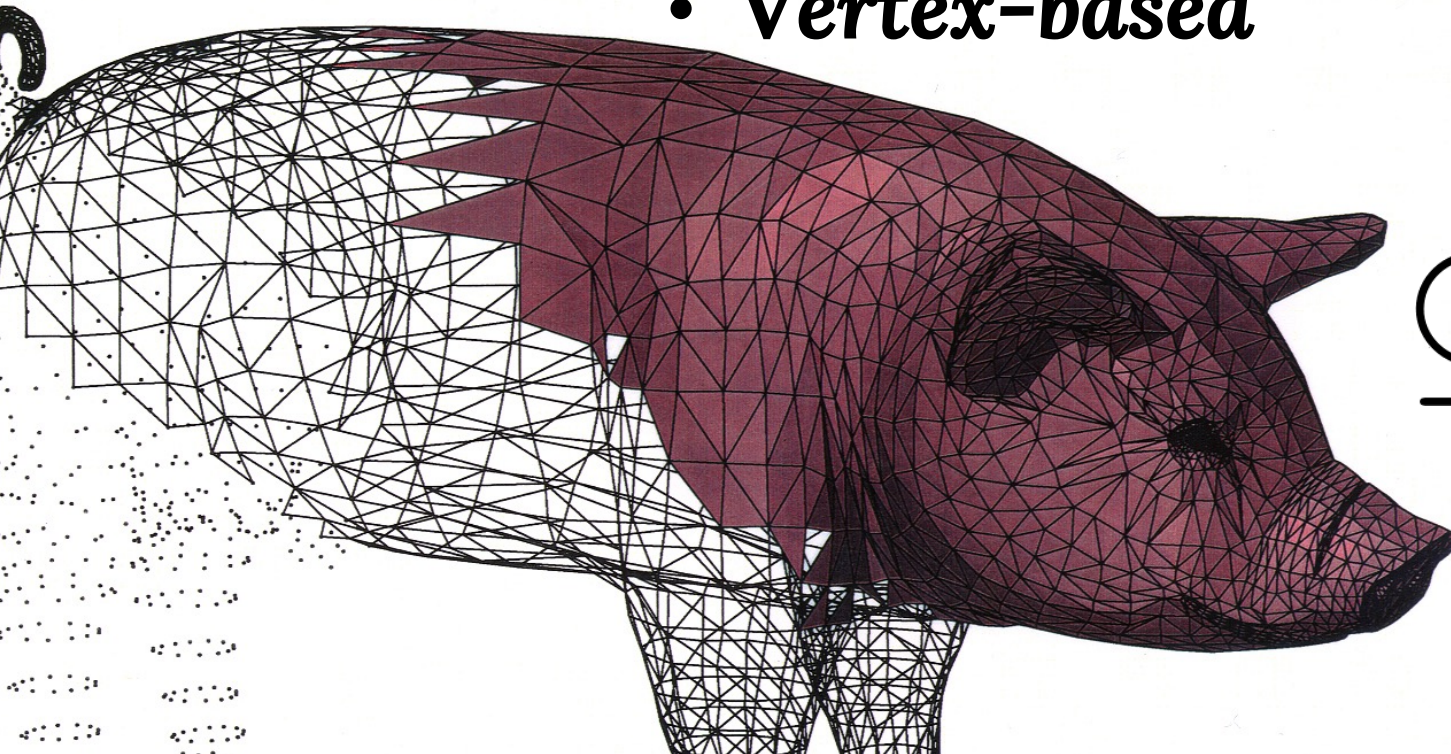


# **DISCRETE VECTOR FIELDS**

# Vector Fields on Triangle Meshes

No consensus:

- **Triangle-based**
- **Edge-based**
- **Vertex-based**

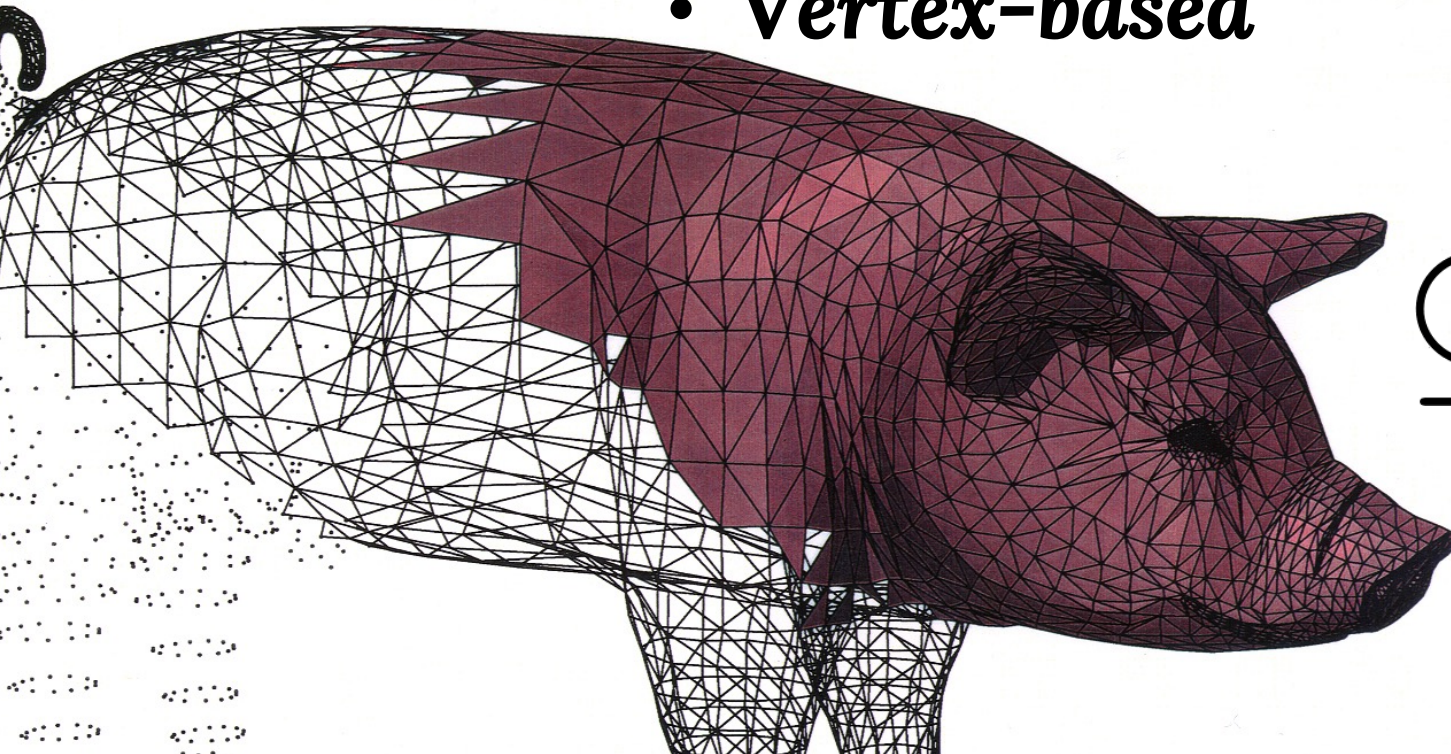


$\subseteq \mathbb{R}^3$

# Vector Fields on Triangle Meshes

No consensus:

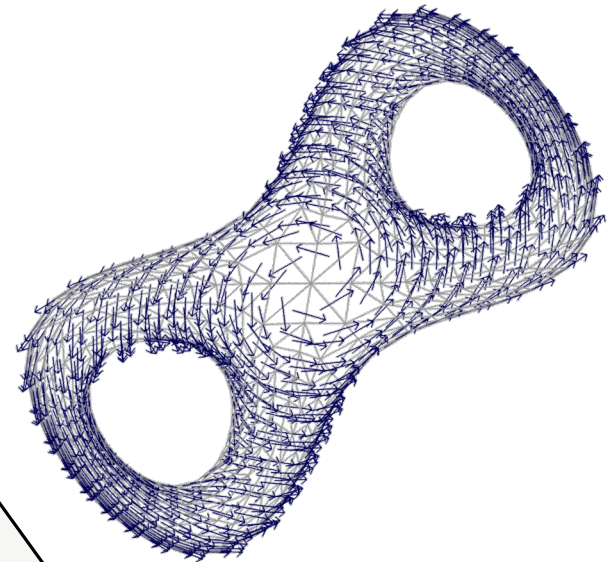
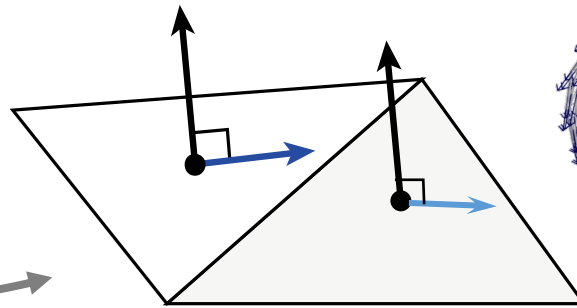
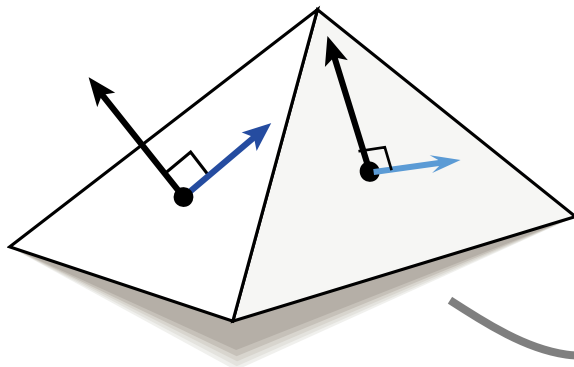
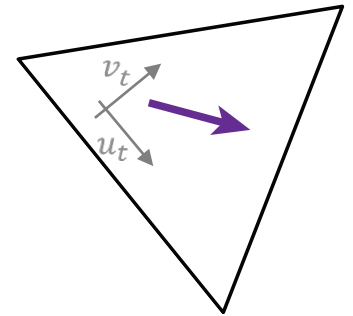
- **Triangle-based**
- **Edge-based**
- **Vertex-based**



$$\subseteq \mathbb{R}^3$$

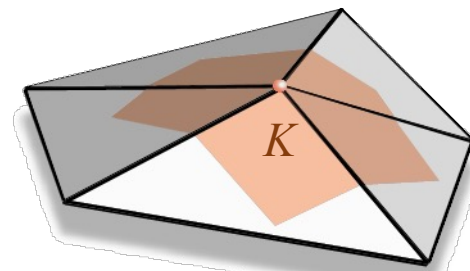
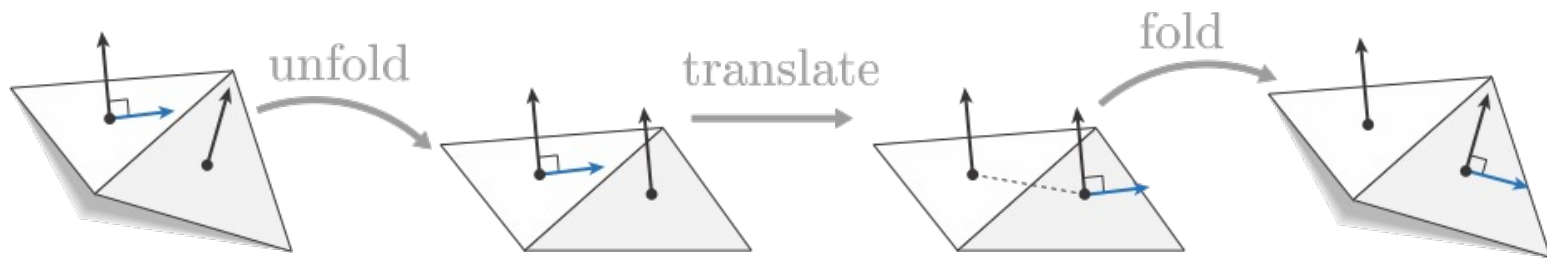
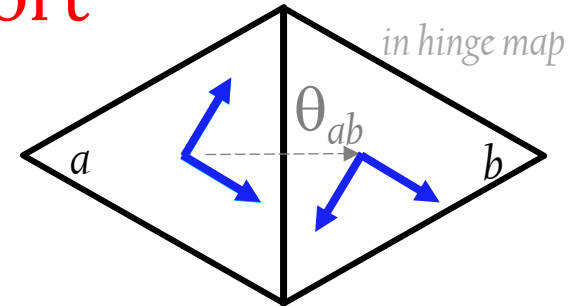
# Triangle-Based

- Triangle as its **own tangent plane**
- One vector per triangle
  - “Piecewise constant”
  - Discontinuous at edges/vertices
- Easy to “unfold”/“hinge”



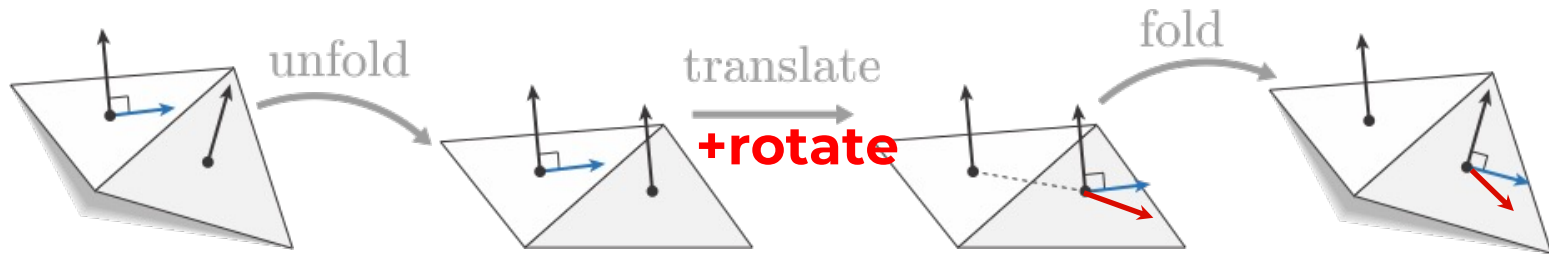
# Discrete Levi-Civita Connection

- Simple notion of **parallel transport**
- Transport around vertex:  
Excess angle is (integrated)  
**Gaussian curvature** (holonomy!)



$$K = 2\pi - \sum_a \gamma_a$$

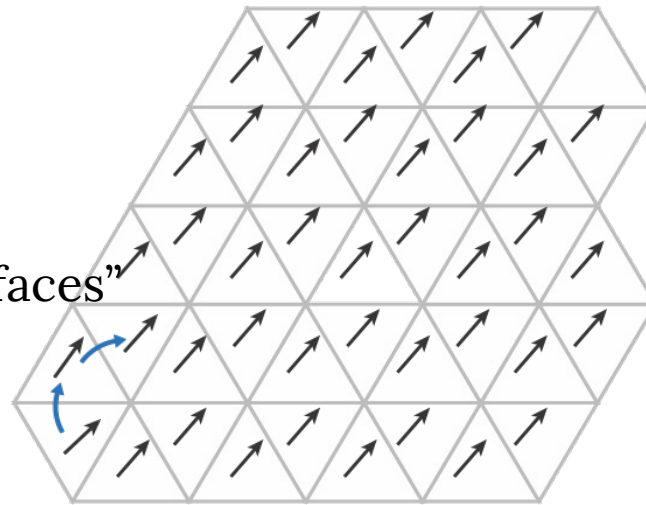
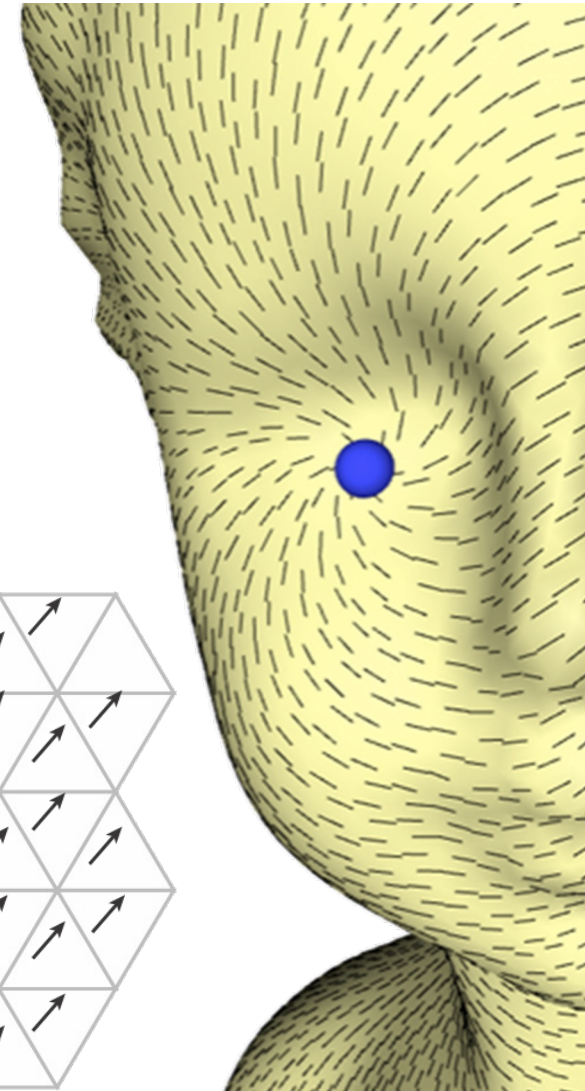
# Arbitrary Connection



Represent using angle  $\theta_{edge}$  of **extra rotation**.

# Trivial Connections

- Vector field design
- **Zero holonomy** on discrete cycles
  - Except for a few singularities
- Path-independent away from singularities



“Trivial Connections on Discrete Surfaces”

Crane et al., SGP 2010

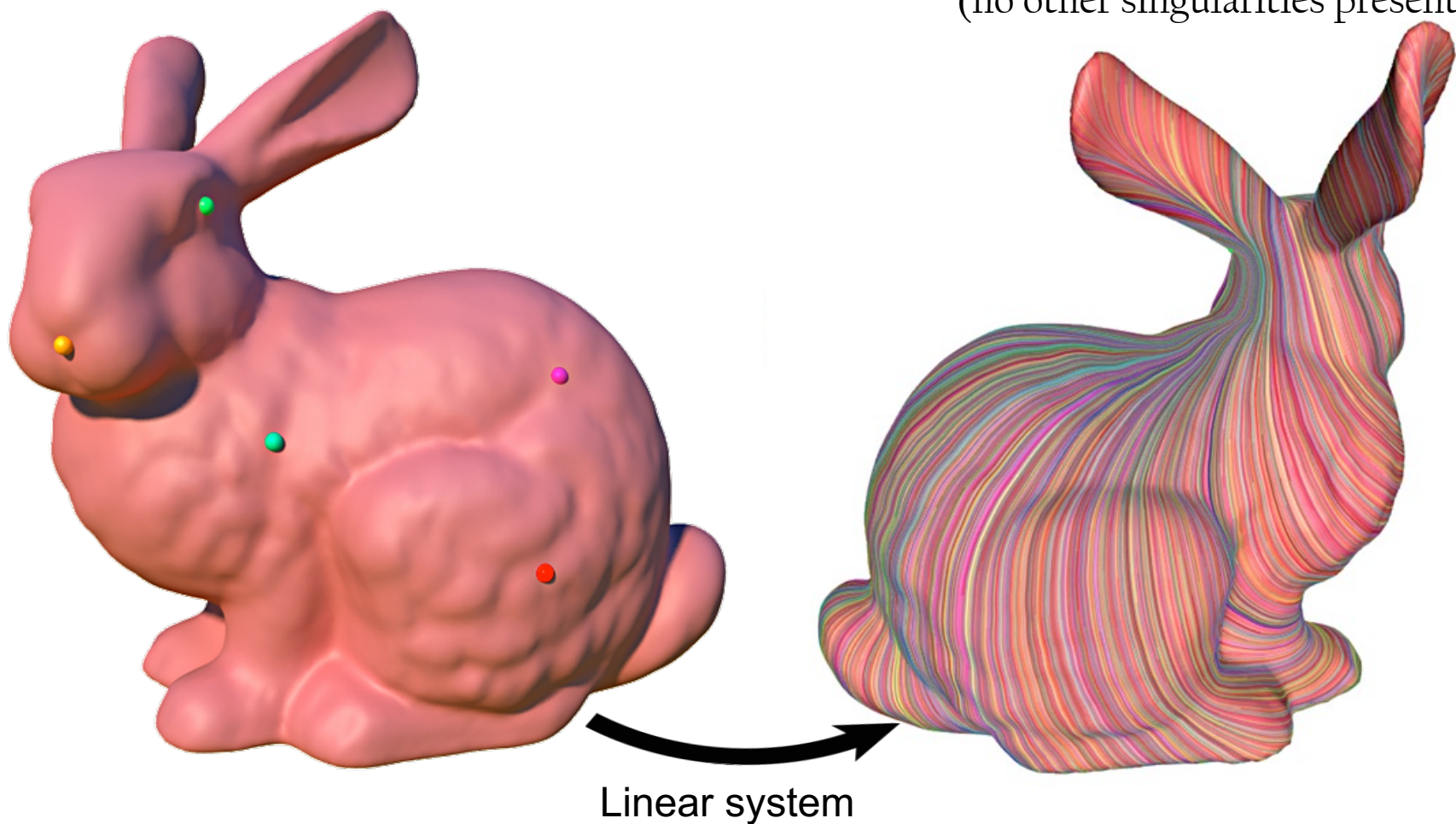
# Trivial Connections: Details

- Solve  $\theta_{edge}$  of **extra rotation** per edge
- Linear constraint:  
**Zero holonomy** on basis cycles
  - **$V+2g$  constraints**: Vertex cycles plus harmonic
  - Fix curvature at chosen singularities
- Underconstrained: **Minimize  $\|\vec{\theta}\|$** 
  - Best approximation of Levi-Civita



# Result

Resulting trivial connection  
(no other singularities present)

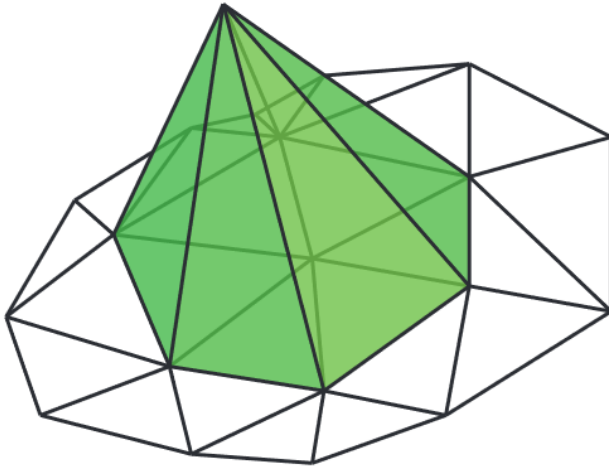


*Aside:*

# Nice 2D Identification

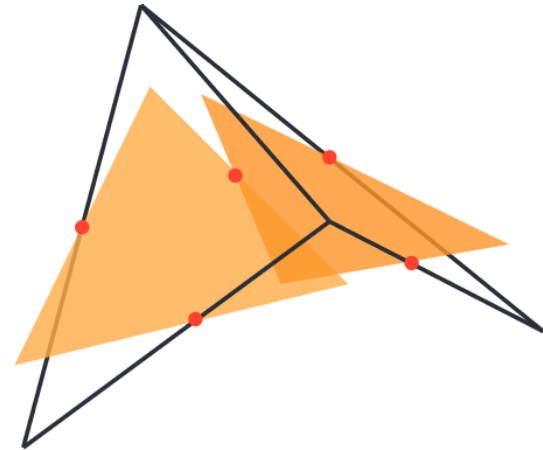
$$\vec{v} \leftrightarrow ae^{i\theta}$$

# Face-Based Calculus



**Vertex-based**

*“Conforming”  
Already familiar*



**Edge-based**

*“Nonconforming”  
[Wardetzky 2006]*

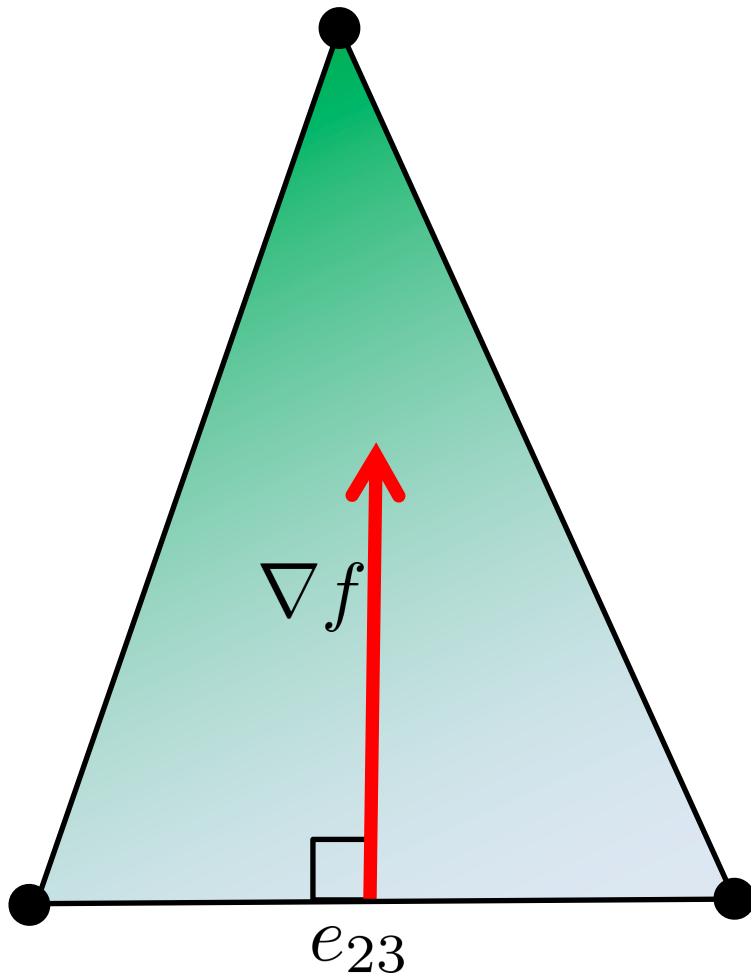
Relationship:  $\psi_{ij} = \phi_i + \phi_j - \phi_k$

**Gradient Vector Field**

*Recall:*

# Gradient of a Hat Function

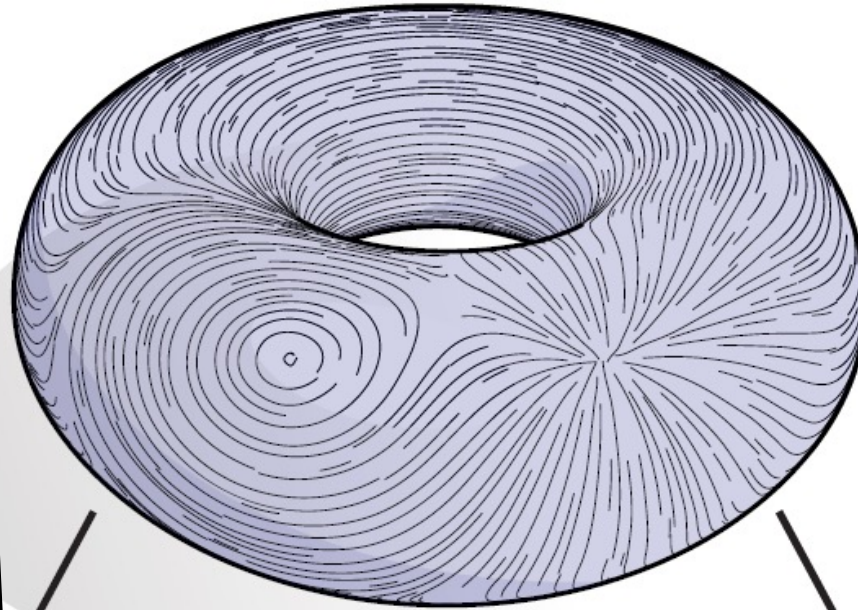
$$\|\nabla f\| = \frac{1}{l_3 \sin \theta_3} = \frac{1}{h}$$



$$\nabla f = \frac{e_{23}^\perp}{2A}$$

*Length of  $e_{23}$  cancels "base" in  $A$*

# Helmholtz-Hodge Decomposition



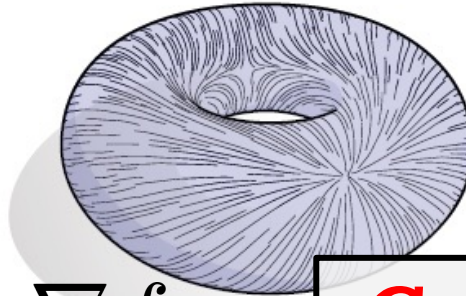
**Divergence free**

$2g$ -dimensional  
**Harmonic**

$\mathcal{R}\nabla f$



$\nabla f$



**Curl free**



*Recall:*

# Euler Characteristic

$$V - E + F := \chi$$

$$\chi = 2 - 2g$$



$$g = 0$$



$$g = 1$$



$$g = 2$$

# Discrete Helmholtz-Hodge

$$2 - 2g = V - E + F$$

$$\implies 2F = (V - 1) + (E - 1) + 2g$$

*Either*

- **Vertex-based** gradients
- **Edge-based** rotated gradients

*Dimensionality  
works out  
perfectly!*

*or*

- **Edge-based** gradients
- **Vertex-based** rotated gradients

*Can work out  
div/grad/curl*

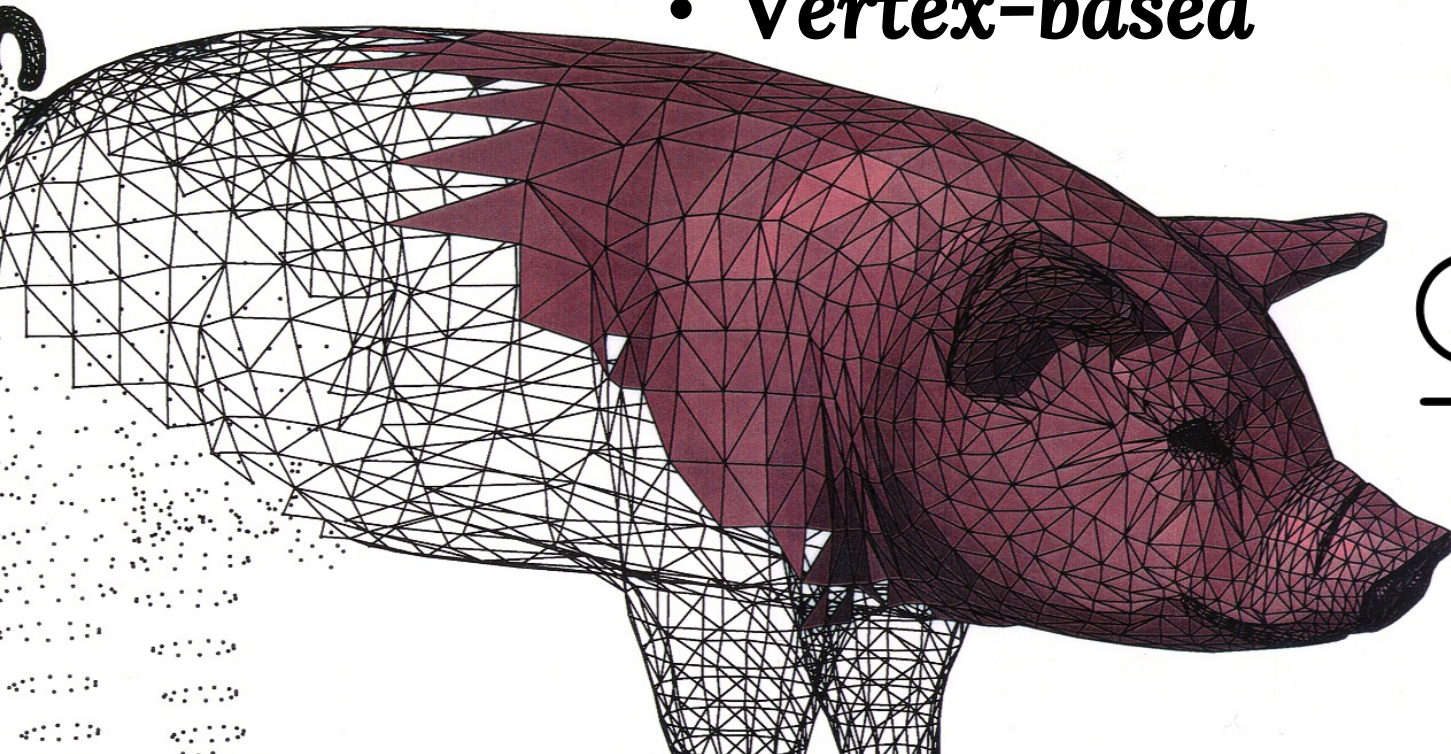
**“Mixed” finite elements**

# Vector Fields on Triangle Meshes

No consensus:

- **Triangle-based**
- **Edge-based**
- **Vertex-based**

*Not covered here!  
Google 'Discrete  
Exterior Calculus'*



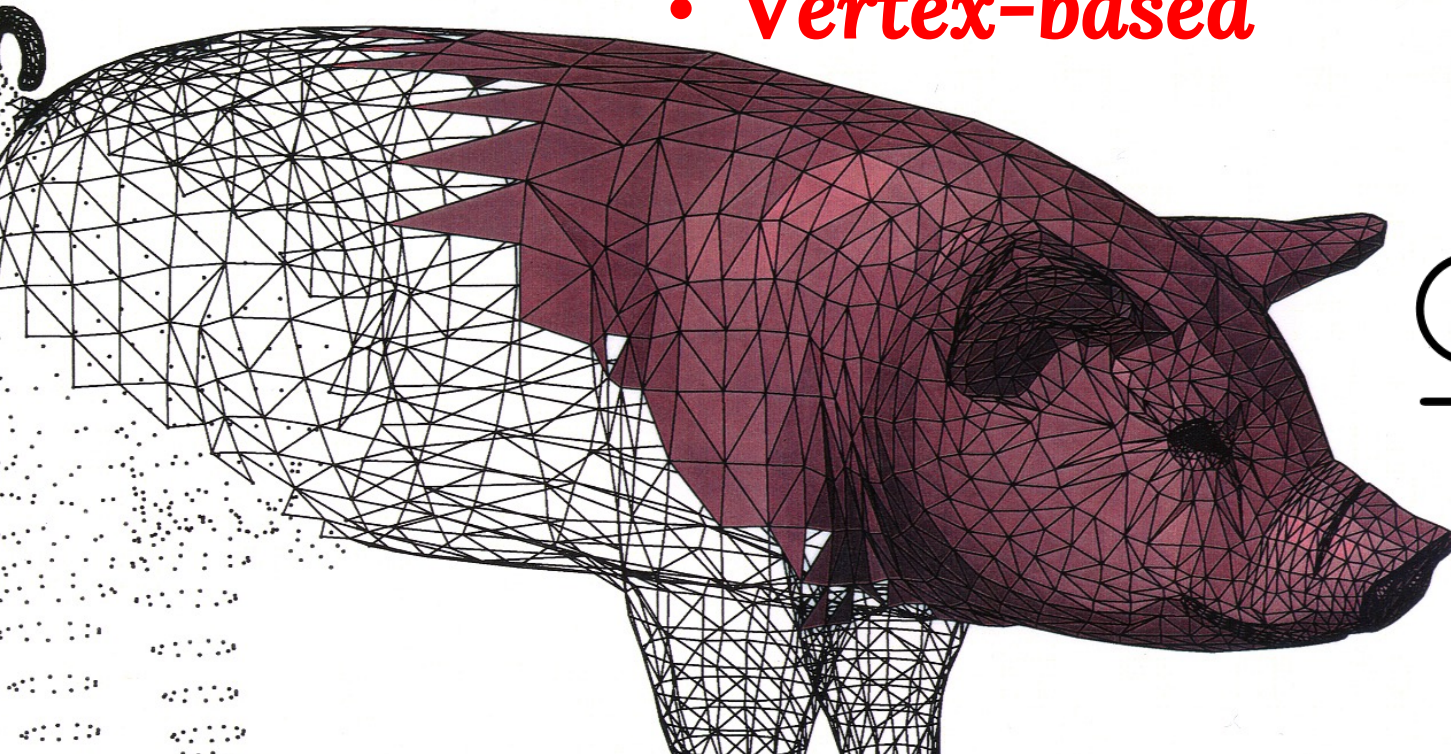
$$\subseteq \mathbb{R}^3$$



# Vector Fields on Triangle Meshes

No consensus:

- **Triangle-based**
- **Edge-based**
- **Vertex-based**



$\subseteq \mathbb{R}^3$

# Vertex-Based Fields

- **Pros**

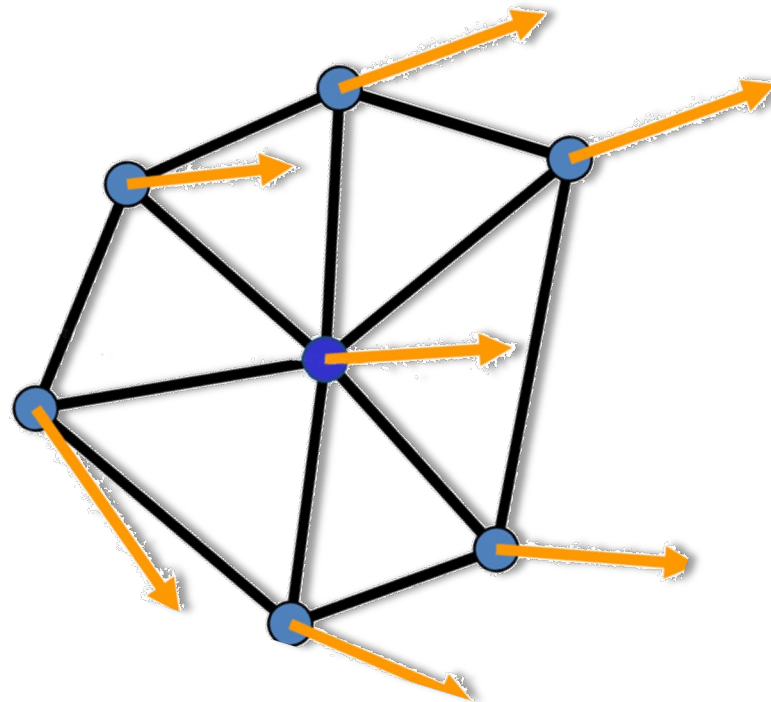
- *Possibility of higher-order differentiation*

- **Cons**

- *Vertices don't have natural tangent spaces*

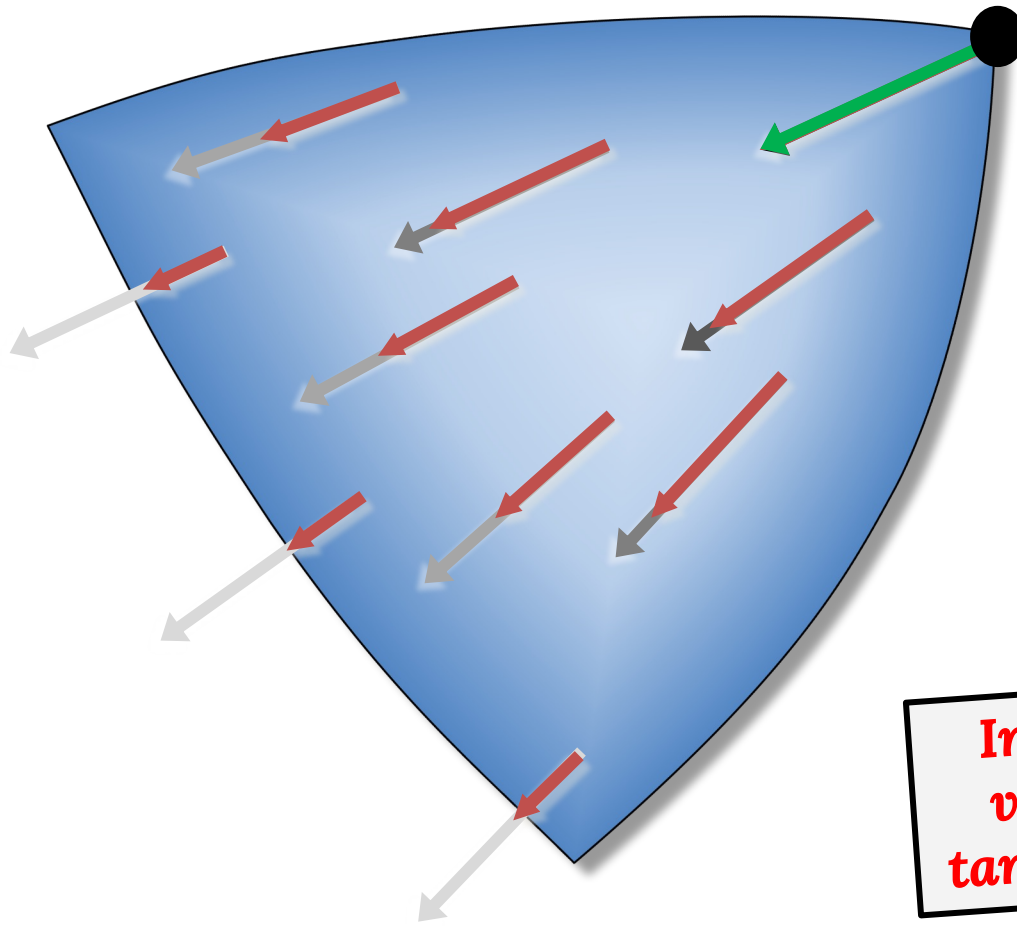
- *Gaussian curvature concentrated*

# 2D (Planar) Case: Easy



***Piecewise-linear  $(x,y)$  components***

# 3D Case: Ambiguous



**Interpolate  
vector and  
tangent space!**

# Recent Method for Continuous Fields

## Discrete Connection and Covariant Derivative for Vector Field Analysis and Design

Beibei Liu and Yiying Tong  
Michigan State University

and

Fernando de Goes and Mathieu Desbrun  
California Institute of Technology

In this paper, we introduce a discrete definition of connection on simplicial manifolds, involving closed-form continuous expressions within simplices and finite rotations across simplices. The finite-dimensional parameters of this connection are optimally computed by minimizing a quadratic measure of the deviation to the (discontinuous) Levi-Civita connection induced by the embedding of the input triangle mesh, or to any metric connection with arbitrary cone singularities at vertices. From this discrete connection, a covariant derivative is constructed through exact differentiation, leading to explicit expressions for local integrals of first-order derivatives (such as divergence, curl and the Cauchy-Riemann operator), and for  $L_2$ -based energies (such as the Dirichlet energy). We finally demonstrate the utility, flexibility, and accuracy of our discrete formulations for the design and analysis of vector,  $n$ -vector, and  $n$ -direction fields.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry & Object Modeling—*Curve & surface representations.*

CCS Concepts: •Computing methodologies → Mesh models;

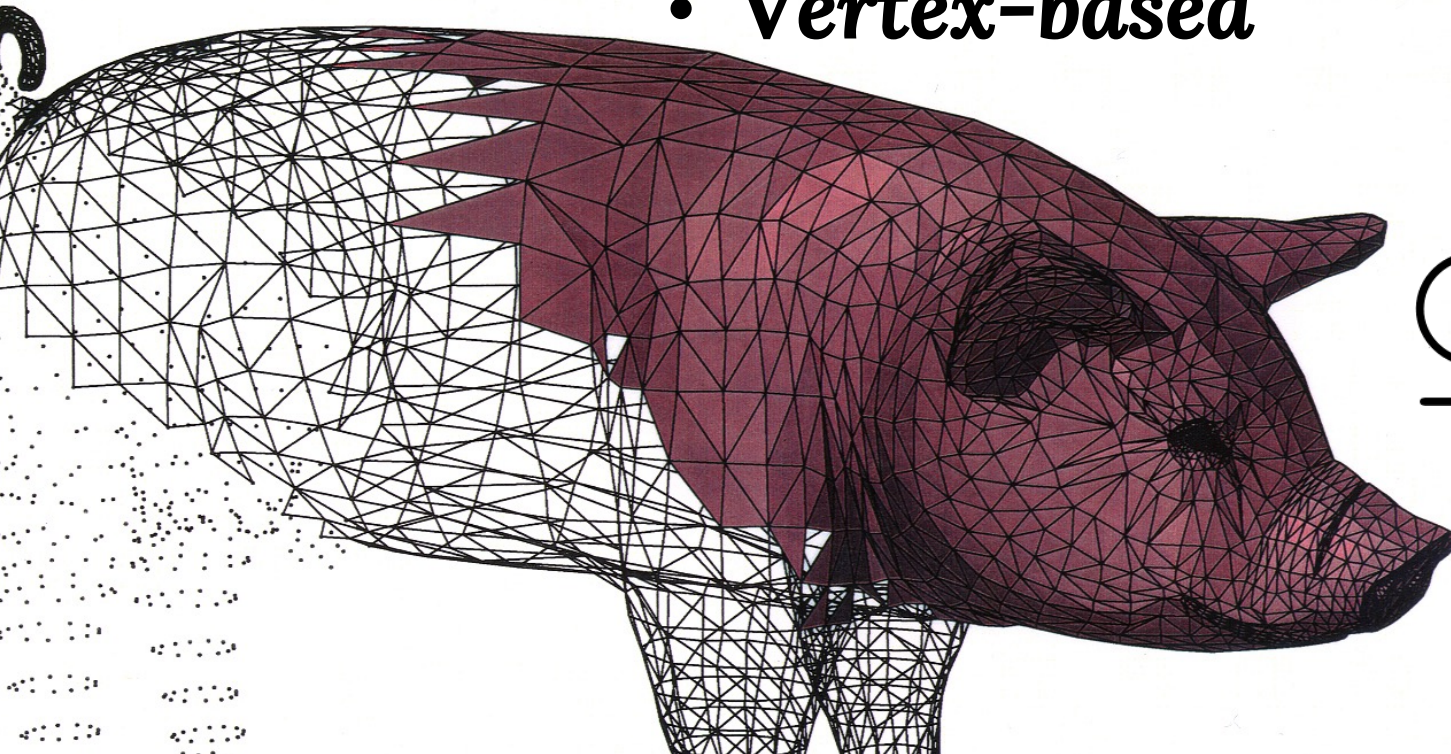
digital geometry processing, with applications ranging from texture synthesis to shape analysis, meshing, and simulation. However, existing discrete counterparts of such a differential operator acting on simplicial manifolds can either approximate local derivatives (such as divergence and curl) or estimate global integrals (such as the Dirichlet energy), but not both simultaneously.

In this paper, we present a unified discretization of the covariant derivative that offers closed-form expressions for both local and global first-order derivatives of vertex-based tangent vector fields on triangulations. Our approach is based on a new construction of discrete connections that provides consistent interpolation of tangent vectors within and across mesh simplices, while minimizing the deviation to the Levi-Civita connection induced by the 3D embedding of the input mesh—or more generally, to any metric connection with arbitrary cone singularities at vertices. We demonstrate the relevance of our contributions by providing new computational tools to design and edit vector and  $n$ -direction fields.

# Vector Fields on Triangle Meshes

No consensus:

- **Triangle-based**
- **Edge-based**
- **Vertex-based**

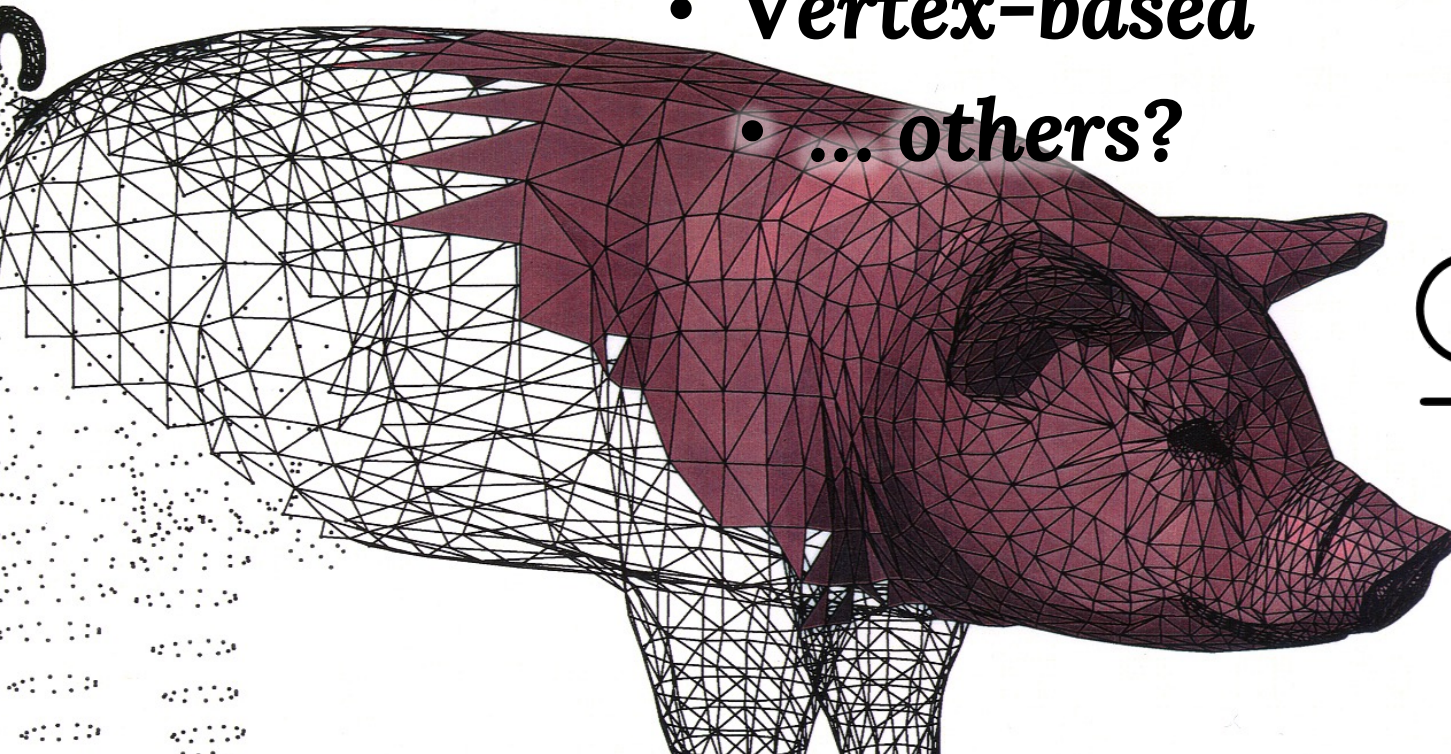


$$\subseteq \mathbb{R}^3$$

# Vector Fields on Triangle Meshes

No consensus:

- **Triangle-based**
- **Edge-based**
- **Vertex-based**
- **... others?**



$$\subseteq \mathbb{R}^3$$

# More Exotic Choice

Eurographics Symposium on Geometry Processing 2013  
Yaron Lipman and Richard Hao Zhang  
(Guest Editors)

Volume 32 (2013), Number 5

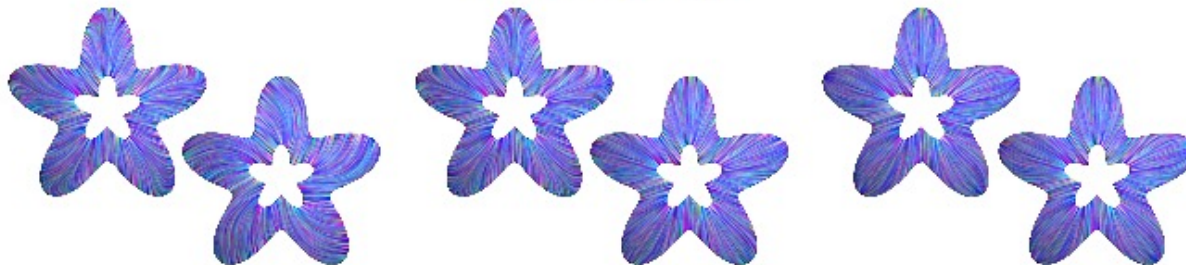
## An Operator Approach to Tangent Vector Field Processing

Omri Azencot<sup>1</sup> and Mirela Ben-Chen<sup>1</sup> and Frédéric Chazal<sup>2</sup> and Maks Ovsjanikov<sup>3</sup>

<sup>1</sup>Technion - Israel Institute of Technology

<sup>2</sup>Geometrica, INRIA

<sup>3</sup>LIX, École Polytechnique



**Figure 1:** Using our framework various vector field design goals can be easily posed as linear constraints. Here, given three symmetry maps: rotational ( $S1$ ), bilateral ( $S2$ ) and front/back ( $S3$ ), we can generate a symmetric vector field using only  $S1$  (left),  $S1 + S2$  (center) and  $S1 + S2 + S3$  (right). The top row shows the front of the 3D model, and the bottom row its back.



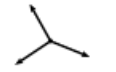






### Abstract

In this paper, we introduce a novel coordinate-free method for manipulating and analyzing vector fields on surfaces. Unlike the commonly used representations of a vector field as an assignment of vectors to the mesh, or as real values on edges, we argue that vector fields can also be naturally viewed as domain and range are functions defined on the mesh. Although this point of view is common in differential geometry it has so far not been adopted in geometry processing applications. We recall the theoretical properties of vector

**Vector fields as  
derivative  
operators**



# Extension: Direction Fields

	1-vector field	One vector, classical “vector field”
	2-direction field	Two directions with $\pi$ symmetry, “line field”, “2-RoSy field”
	$1^3$ -vector field	Three independent vectors, “3-polyvector field”
	4-vector field	Four vectors with $\pi/2$ symmetry, “non-unit cross field”
	4-direction field	Four directions with $\pi/2$ symmetry, “unit cross field”, “4-RoSy field”
	$2^2$ -vector field	Two pairs of vectors with $\pi$ symmetry each, “frame field”
	$2^2$ -direction field	Two pairs of directions with $\pi$ symmetry each, “non-ortho. cross field”
	6-direction field	Six directions with $\pi/3$ symmetry, “6-RoSy”
	$2^3$ -vector field	Three pairs of vectors with $\pi$ symmetry each

# Polyvector Fields

Eurographics Symposium on Geometry Processing 2014  
Thomas Funkhouser and Shi-Min Hu  
(Guest Editors)

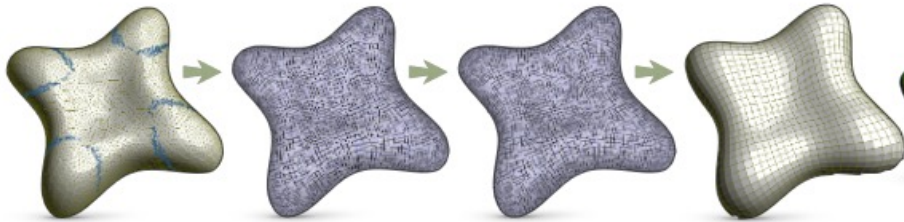
Volume 33 (2014), Number 5

## Designing $N$ -PolyVector Fields with Complex Polynomials

Olga Diamanti<sup>1</sup> Amir Vaxman<sup>2</sup> Daniele Panozzo<sup>1</sup> Olga Sorkine-Hornung<sup>1</sup>

<sup>1</sup>ETH Zurich, Switzerland

<sup>2</sup>Vienna Institute of Technology, Austria



$$\begin{aligned} & \{u_0, u_1, \dots, u_k\} \\ \mapsto & f(z) := (z - u_0) \cdots (z - u_k) \\ \mapsto & f(z) = z^{k+1} + a_k z^k + \cdots + a_1 z + a_0 \\ \mapsto & \{a_0, \dots, a_k\} \end{aligned}$$

**Figure 1:** A smooth 4-PolyVector field is generated from a sparse set of principal direction constraints (faces in light blue). We optimize the field for conjugacy and use it to guide the generation of a planar-quad mesh. Pseudocolor represents planarity.

### Abstract

We introduce  $N$ -PolyVector fields, a generalization of  $N$ -RoSy fields for which the vectors are neither necessarily orthogonal nor rotationally symmetric. We formally define a novel representation for  $N$ -PolyVectors as the root sets of complex polynomials and analyze their topological and geometric properties. A smooth  $N$ -PolyVector field can be efficiently generated by solving a sparse linear system without integer variables. We exploit the flexibility of  $N$ -PolyVector fields to design conjugate vector fields, offering an intuitive tool to generate planar quadrilateral meshes.

### 1. Introduction

ing  $N$  unit-length vectors related by a rotation of an integer

# One encoding of direction fields