

Operator-Based Representations of Discrete Tangent Vector Fields

Mirela Ben-Chen^a, Omri Azencot^b

^a*Computer Science Department, Technion - Israel Institute of Technology, Haifa, Israel*

^b*Math Department, University of California, Los Angeles, USA.*

Abstract

Tangent vector fields are prominent objects in geometry processing, and are used in a myriad of applications, e.g., texture synthesis, fluid simulation and polygonal remeshing. It is, however, challenging to represent, analyze and synthesize tangent vector fields on discrete surfaces such as triangle meshes. The goal of this chapter is to review the use of *linear operators* for these tasks. We introduce the formulation, show the advantages and demonstrate the unique applications such a representation makes possible. We additionally discuss the relation to functional maps that were introduced in previous chapters.

Keywords: Geometry processing, tangent vector fields, linear operators, discrete differential geometry

List of Key Symbols

Smooth

\mathcal{M} A smooth, orientable and compact surface embedded in \mathbb{R}^3

\mathcal{X} Surface embedding $\mathcal{M} \rightarrow \mathbb{R}^3$,

f, g, h Real valued functions $\mathcal{M} \rightarrow \mathbb{R}$

u, v Tangent vector fields $\mathcal{M} \rightarrow T\mathcal{M}$

$\langle u, v \rangle$ Pointwise inner product of two tangent vector fields, given by the Riemannian metric on \mathcal{M}

∇f Gradient vector field of f

$\nabla \cdot v$ Divergence function of v

Δ Laplace-Beltrami operator

\mathcal{D}_v Functional vector field (FVF) corresponding to v

$\overline{\mathcal{D}}_f$ Dual FVF corresponding to f

$\tilde{\mathcal{D}}_v$ Divergence-based FVF corresponding to v

$\varphi_{v,t}$ The flow map of the vector field v for time $t \in \mathbb{R}$, $\varphi_{v,t} : \mathcal{M} \rightarrow \mathcal{M}$

$\phi_{v,t}$ The functional flow map of the vector field v for time $t \in \mathbb{R}$

Discrete

M A triangle mesh, where the vertices, edges and faces are denoted by $(\mathcal{V}, \mathcal{E}, \mathcal{F})$, respectively.

n Number of vertices $|\mathcal{V}|$

m Number of faces $|\mathcal{F}|$

X Mesh embedding $X \in \mathbb{R}^{n \times 3}$

f, g, h Piecewise linear functions $f, g, h \in \mathbb{R}^n$

u, v Piecewise constant tangent vector fields $u, v \in \mathbb{R}^{3m}$

grad Gradient operator $\text{grad} \in \mathbb{R}^{3m \times n}$

div Divergence operator $\text{div} \in \mathbb{R}^{n \times 3m}$

$A_{\mathcal{V}}$ Vector of vertex areas $A_{\mathcal{V}} \in \mathbb{R}^n$

$A_{\mathcal{F}}$ Vector of face areas $A_{\mathcal{F}} \in \mathbb{R}^m$

$I_{\mathcal{V}}^{\mathcal{F}}$ Interpolation matrix from faces to vertices $I_{\mathcal{V}}^{\mathcal{F}} \in \mathbb{R}^{n \times m}$

$I_{\mathcal{F}}^{\mathcal{V}}$ Interpolation matrix from vertices to faces $I_{\mathcal{F}}^{\mathcal{V}} \in \mathbb{R}^{m \times n}$

L Laplace-Beltrami operator $L \in \mathbb{R}^{n \times n}$

$D_v^{\mathcal{F}}$ Rectangular FVF corresponding to v , $D_v^{\mathcal{F}} \in \mathbb{R}^{m \times n}$

$D_v^{\mathcal{V}}$ Square FVF corresponding to v , $D_v^{\mathcal{V}} \in \mathbb{R}^{n \times n}$

$\overline{D}_f^{\mathcal{F}}$ Dual to rectangular FVF corresponding to f , $\overline{D}_f^{\mathcal{F}} \in \mathbb{R}^{m \times 3m}$

$\overline{D}_f^{\mathcal{V}}$ Dual to square FVF corresponding to f , $\overline{D}_f^{\mathcal{V}} \in \mathbb{R}^{n \times 3m}$

$\tilde{D}_v^{\mathcal{F}}$ Rectangular divergence-based FVF corresponding to v , $\tilde{D}_v^{\mathcal{F}} \in \mathbb{R}^{n \times m}$

$\tilde{D}_v^{\mathcal{V}}$ Square divergence-based FVF corresponding to v , $\tilde{D}_v^{\mathcal{V}} \in \mathbb{R}^{n \times n}$

$\Phi_{v,t}$ The functional flow map of the vector field v for time $t \in \mathbb{R}$, $\Phi_{v,t} \in \mathbb{R}^{n \times n}$



Figure 1: A nearly symmetric quadrangular mesh generated using the operator based representation of vector fields (Azencot et al., 2017) (left and right). The symmetry enables the generation of a very low resolution model, which can be fabricated using additive manufacturing, by thickening the quad mesh edges (left center and right center). The center image is a photograph of the final printed model.

1. Introduction

Tangent vector fields are geometric objects that represent *directional* information tangent to a surface. Examples include: velocity fields, curvature directions and tangents to the parameter lines of a surface parameterization.

On smooth surfaces, a tangent vector field is often defined as a *smooth* assignment of a tangent vector to each point on the surface. In geometry processing and computer graphics, surfaces are often represented using triangle meshes. Thus, on a triangle mesh it is natural to represent a discrete tangent vector field using a *piecewise constant* (PC) representation, namely, as a constant tangent vector per triangle. However, formulating and enforcing smoothness of PC vector fields is challenging, since they are piecewise constant and therefore non-differentiable. Some approaches to tackle this problem include using finite differences (Crane et al., 2010), using a higher order polynomial discretization (Zhang et al., 2006; Knöppel et al., 2013), and using discrete 1-forms (Desbrun et al., 2005).

In this chapter, we advocate a different approach, first suggested by Azencot et al. (2013). Specifically, we consider the *action* of tangent vector fields on scalar functions, by treating tangent vector fields as *directional derivative* operators. These operators are linear, and, in the smooth case, completely encode the corresponding vector fields. The smooth theory involving these operators, also known as *derivations* (Morita, 2001, Chapter 1.4), is quite vast, as this operator-based point of view is classical in differential geometry.

In the discrete setting, however, this discretization is novel, and considerably simplifies many vector field related tasks, for both synthesis and analysis. This is due to two main properties of this scheme: (1) the only objects that are differentiated are *scalar functions*, and (2) the linear operator can be represented as a compact matrix in a reduced spectral basis. Consider, for example, the problem of joint design of vector fields on two corresponding surfaces, or on an intrinsically symmetric surface \mathcal{M} with a self-correspondence $\varphi : \mathcal{M} \rightarrow \mathcal{M}$. A tangent vector field v is said to be *consistent* with φ if $d\varphi(v(q)) = v(\varphi(q))$ for any $q \in \mathcal{M}$. This definition requires the *differential* of the map, which in a discrete setting on a triangle mesh is difficult to obtain. However, if we use the *operator* point of view, we can formulate the consistency by $\langle v, \nabla(\ell \circ \varphi) \rangle = \langle v, \nabla \ell \rangle \circ \varphi$ for any scalar function $\ell : \mathcal{M} \rightarrow \mathbb{R}$. With this formulation, there is no need for the map differential, which considerably simplifies the implementation on discrete surfaces. Figure 1 shows an example of a nearly symmetric quadrangular mesh generated using this approach (Azencot et al., 2017).

Due to these two properties, the operator-based approach to tangent vector fields enables many additional applications, such as: vector field design (Azencot et al., 2013), function transport (Azencot et al., 2013, 2014a, 2015b, 2016) and vector field transport (Azencot et al., 2015a, 2018).

1.1. Organization

The rest of the chapter is organized as follows. In Section 2 we present the formulation of operator-based tangent vector fields in the smooth setting. We then proceed to describe the discrete representation, its properties and some applications in Section 3, which is based on the paper that first introduced the topic (Azencot et al., 2013). Then, in Section 4 we present a new, divergence based discretization of the smooth setting, which allows for a more accurate representation of composite operators, such as the *Lie bracket* operator, and we present a few applications. We conclude with a brief outlook and some open questions in section 5.

2. Smooth Functional Vector Fields

In this section we provide the definition of *functional vector fields*, or the operator representation of tangent vector fields, in the smooth setting. For clarity of the exposition we avoid fully rigorous definitions, and refer the interested reader to the excellent book by Morita (2001) for a full treatment.

2.1. Notation

In the following, \mathcal{M} is a smooth, orientable, compact and embedded surface, and $\ell, g, h : \mathcal{M} \rightarrow \mathbb{R}$ are C^∞ real valued functions on it. In addition, u, v

are tangent vector fields on \mathcal{M} , and $\langle v, u \rangle$ is a function $\mathcal{M} \rightarrow \mathbb{R}$ such that $\langle v, u \rangle(p) = \langle v(p), u(p) \rangle_p$, for any point $p \in \mathcal{M}$. Here, the inner product $\langle \cdot, \cdot \rangle_p$ is given by the Riemannian metric of \mathcal{M} at p .

2.2. Directional Derivative of Functions

A tangent vector field v on \mathcal{M} is given by a smooth assignment of a tangent vector $v(p)$ to every point $p \in \mathcal{M}$. Such a vector field can be used to compute the *directional derivative* of a smooth real valued function $f : \mathcal{M} \rightarrow \mathbb{R}$. The directional derivative function $g : \mathcal{M} \rightarrow \mathbb{R}$ is given by:

$$g = \langle v, \nabla f \rangle, \quad (1)$$

where ∇f is the gradient vector field of f .

Thus, given v , we can compute the directional derivative of any function f . Intriguingly, *the opposite is also true*. If two vector fields agree on the directional derivative of *any* arbitrary function f , then the two vector fields are identical (Morita, 2001, Prop. 1.39). Hence, we can *represent* a vector field uniquely by its directional derivative action, as is formalized in the next definition.

2.3. Functional Vector Fields

Definition 1. Given a tangent vector field v on \mathcal{M} , the corresponding functional vector field operator \mathcal{D}_v is given by:

$$\mathcal{D}_v(f) = \langle v, \nabla f \rangle. \quad (2)$$

The functional vector field operator \mathcal{D}_v is linear in the function it acts on, due to the linearity of the gradient. Hence,

$$\mathcal{D}_v(af + bg) = a\mathcal{D}_v(f) + b\mathcal{D}_v(g), \quad (3)$$

for any two functions $f, g : \mathcal{M} \rightarrow \mathbb{R}$ and any two constants $a, b \in \mathbb{R}$.

We further define the *dual* functional vector field operator, which switches the role of the function and the vector field.

Definition 2. Given a real valued function $f : \mathcal{M} \rightarrow \mathbb{R}$, the corresponding dual functional vector field operator $\bar{\mathcal{D}}_f$ is given by:

$$\bar{\mathcal{D}}_f(v) = \mathcal{D}_v(f) = \langle v, \nabla f \rangle. \quad (4)$$

Due to the linearity of the inner product, the dual operator is linear in its vector field argument, with respect to both functions and constants. Thus,

$$\bar{\mathcal{D}}_h(fv + gu) = f\bar{\mathcal{D}}_h(v) + g\bar{\mathcal{D}}_h(u), \quad (5)$$

for any two tangent vector fields u, v , and functions $h, f, g : \mathcal{M} \rightarrow \mathbb{R}$.

As discussed previously, a functional vector field operator uniquely encodes its corresponding vector field. If the surface is *embedded* in \mathbb{R}^3 , the vector field can be easily extracted from the corresponding operator using the following Lemma.

Lemma 1. Let ν be a tangent vector field on \mathcal{M} , and let \mathcal{D}_ν be its corresponding functional vector field operator. Further, denote the embedding of \mathcal{M} by $\mathcal{X} : \mathcal{M} \rightarrow \mathbb{R}^3$, where \mathcal{X} is given by the three coordinate functions $\mathcal{X} = (x_1, x_2, x_3)$. Then:

$$\mathcal{D}_\nu(x_i) = \nu_i, \quad i \in (1, \dots, 3), \quad (6)$$

where ν_i are the coordinate functions of ν .

Proof. One approach to compute the gradient of a function $\mathcal{M} \rightarrow \mathbb{R}$ is to extend it to a function over \mathbb{R}^3 , take its Euclidean gradient and project the result on $T_p\mathcal{M}$, the tangent plane to \mathcal{M} at p (Absil et al., 2009, Eq.(3.37)). Let (x_1, x_2, x_3) denote the canonical coordinates of \mathbb{R}^3 . By the definition of \mathcal{X} we have that $x_i(p) = \mathcal{X}_i(p)$, for all the points $p \in \mathcal{M}$, and therefore x_i is the extension to \mathbb{R}^3 of \mathcal{X}_i . Hence, we have that

$$(\nabla x_i)(p) = [(\nabla x_i)(p)]_{T_p\mathcal{M}} = \left[\sum_{j=1}^3 \frac{\partial x_i}{\partial x_j}(p) \hat{x}_j \right]_{T_p\mathcal{M}},$$

where $[\cdot]_{T_p\mathcal{M}}$ denotes projection on $T_p\mathcal{M}$ and $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ denote the canonical directions in \mathbb{R}^3 .

Note that for all $p \in \mathcal{M}$, $(\frac{\partial x_i}{\partial x_j})(p) = \delta_{ij}$, i.e., it equals 1 when $i = j$ and 0 otherwise. Thus, we have:

$$\langle \nu, \nabla x_i \rangle(p) = \left\langle \nu(p), \left[\sum_{j=1}^3 \delta_{ij} \hat{x}_j \right]_{T_p\mathcal{M}} \right\rangle_p = \left\langle \nu(p), \sum_{j=1}^3 \delta_{ij} \hat{x}_j \right\rangle, \quad (7)$$

where the second equality holds since ν is tangent to \mathcal{M} , and we assumed that the Riemannian metric is inherited from the standard inner product in \mathbb{R}^3 .

Finally, by definition, ν is given in coordinates by $\nu(p) = \sum_{j=1}^3 \nu_j(p) \hat{x}_j$. Therefore:

$$(\mathcal{D}_\nu(x_i))(p) = \langle \nu, \nabla x_i \rangle(p) = \sum_{j=1}^3 \delta_{ij} \nu_j(p) = \nu_i(p), \quad (8)$$

which completes the proof. \square

2.4. Flow Maps

Tangent vector fields are closely related to *self-maps* of \mathcal{M} . Consider a particle at a point $p \in \mathcal{M}$ (see Figure 2). Given a tangent vector field ν , we can ask where will this particle be after time t , if its velocity is given by ν . Such a particle will trace a *flow line* $\gamma_\nu(p)(t)$, $t \in \mathbb{R}$ of ν on \mathcal{M} , that starts at p for $t = 0$. This flow line is also known as an *integral curve*, and it fulfills:

$$\frac{d}{dt} \gamma_\nu(p)(t) = \nu(\gamma_\nu(p)(t)), \quad \gamma_\nu(p)(0) = p, \quad \forall t \in \mathbb{R}. \quad (9)$$

Similarly, the paths that *all* the points on \mathcal{M} trace, for every $t \in \mathbb{R}$, define a map, in fact a *diffeomorphism*, from \mathcal{M} to itself, known as the *flow-map*.

Definition 3. The flow-map of ν is denoted by $\varphi_{\nu,t} : \mathcal{M} \rightarrow \mathcal{M}$, and is given by $\varphi_{\nu,t}(p) = \gamma_\nu(p)(t)$, for all $p \in \mathcal{M}$, $t \in \mathbb{R}$.

For compact surfaces, the flow map is guaranteed to exist for any vector field ν and any $t \in \mathbb{R}$ (Morita, 2001, Thm 1.45). Furthermore, the flow maps have additional structure, since (Morita, 2001, Prop. 1.42):

$$\varphi_{\nu,t} \circ \varphi_{\nu,s} = \varphi_{\nu,t+s}, \quad \forall t, s \in \mathbb{R}. \quad (10)$$

The group of diffeomorphisms $\{\varphi_{\nu,t} \mid t \in \mathbb{R}\}$, with the group operation given by composition, is known as the *one parameter group of transformations* generated by ν . By applying Equation (9) to all points $p \in \mathcal{M}$, we get that the flow-maps fulfill:

$$\frac{d}{dt} \varphi_{\nu,t} = \nu(\varphi_{\nu,t}), \quad \varphi_{\nu,0} = Id. \quad (11)$$

Finally, combining Equations (10) and (11), it is straightforward to see that the *inverse flow map* is given by $\varphi_{\nu,t}^{-1} = \varphi_{\nu,-t}$.

2.5. Functional Flow-maps

In previous chapters, *Functional maps* (Ovsjanikov et al., 2012) have played an important role. Given a diffeomorphism $\mathcal{T}_{12} : \mathcal{M}_1 \rightarrow \mathcal{M}_2$, the corresponding functional map takes functions on \mathcal{M}_1 to functions on \mathcal{M}_2 . The functional map is given by composition with the inverse of \mathcal{T}_{12} , and can be used to transport functions between surfaces, even without the knowledge of the corresponding pointwise map.

In the context of flow-maps, we can think of $\varphi_{\nu,t}$ as the transporter of quantities, and define the corresponding *functional flow map*.

Definition 4. The functional flow-map of ν , denoted by $\phi_{\nu,t}$, takes functions on \mathcal{M} to functions on \mathcal{M} , and is given by:

$$\phi_{\nu,t}(\mathcal{f})(p) = \mathcal{f}(\varphi_{\nu,-t}(p)), \quad \forall \mathcal{f} : \mathcal{M} \rightarrow \mathbb{R}, p \in \mathcal{M}, t \in \mathbb{R}. \quad (12)$$

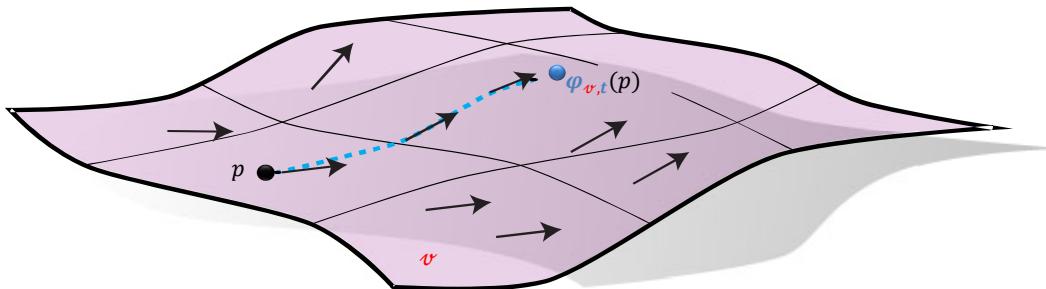


Figure 2: Illustration of a tangent vector field ν and its flow $\varphi_{\nu,t}$.

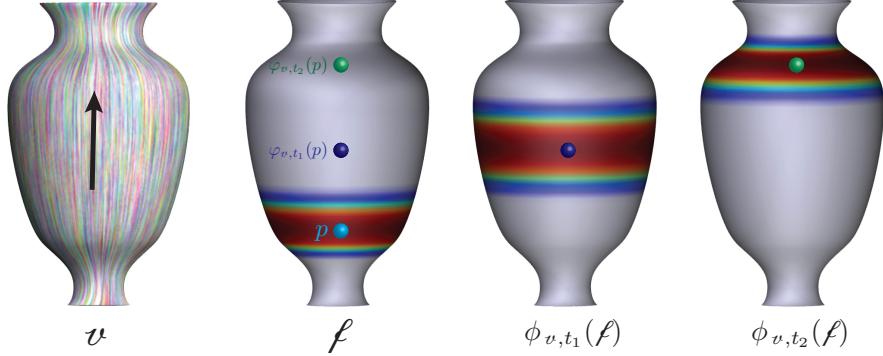


Figure 3: Illustration of the transportation of a function f on the flow of a vector field v (visualized using Line Integral Convolution), for two values of $t = t_1, t_2$.

See Figure 3 for an example of a function f transported on the flow of a vector field v for two values of t , denoted by t_1, t_2 . Consider, for example, the value of the function $\phi_{v,t_1}(f)$ at the dark blue point $\varphi_{v,t_1}(p)$ (Figure 3, center right). By Definition 4 this value is $\phi_{v,t_1}(f)(\varphi_{v,t_1}(p)) = f(\varphi_{v,-t_1}(\varphi_{v,t_1}(p))) = f(p)$. Hence, this value is equal to the value of the function f (Figure 3, center left) at the light blue point p .

Functional flow-maps have some interesting properties. First, note that the functional flow-maps fulfill the same composition rules as the flow-maps.

Lemma 2. *Let $\{\phi_{v,t} \mid t \in \mathbb{R}\}$ be the group of functional flow-maps corresponding to the one parameter group of transformations $\varphi_{v,t}$ generated by the tangent vector field v . Then the following holds for any function $f : \mathcal{M} \rightarrow \mathbb{R}$:*

$$\phi_{v,t}(\phi_{v,s}f) = \phi_{v,t+s}(f), \quad \forall t, s \in \mathbb{R}, \quad (13)$$

and in addition, $\phi_{v,0}f = f$.

Proof. By definition we have for any point $p \in \mathcal{M}$:

$$\phi_{v,t}(\phi_{v,s}f)(p) = (\phi_{v,s}f)(\varphi_{v,-t}(p)) = f(\varphi_{v,-s}(\varphi_{v,-t}(p))). \quad (14)$$

By Equation (10) we get:

$$f(\varphi_{v,-s}(\varphi_{v,-t}(p))) = f(\varphi_{v,-(t+s)}(p)) = \phi_{v,t+s}(f)(p). \quad (15)$$

Finally, $\phi_{v,0}(f)(p) = f(\varphi_{v,0}(p)) = f(p)$, due to Equation (11), which completes the proof. \square

Furthermore, the functional flow-map is closely related to the functional vector field operator by differentiation with respect to t , as follows.

Lemma 3. Let $\{\phi_{v,t} \mid t \in \mathbb{R}\}$ be the group of functional flow-maps corresponding to the one parameter group of transformations $\varphi_{v,t}$ generated by the tangent vector field v . Then the following holds for any function $f : \mathcal{M} \rightarrow \mathbb{R}$ (Morita, 2001, Eq. (2.28)):

$$\frac{d}{dt}\phi_{v,t}(f)|_{t=0} = -\mathcal{D}_v(f). \quad (16)$$

We repeat the proof here, for completeness, and using our notations and definitions so far.

Proof. By the definition of $\phi_{v,t}$ and $\varphi_{v,t}$ we have:

$$\begin{aligned} \frac{d}{dt}\phi_{v,t}(f)(p) &= \frac{d}{dt}f(\varphi_{v,-t}(p)) \\ &= \langle (\nabla f)(\varphi_{v,-t}(p)), \frac{d}{dt}\varphi_{v,-t}(p) \rangle && \text{Multivariate chain rule} \\ &= \langle (\nabla f)(\varphi_{v,-t}(p)), -v(\varphi_{v,-t}(p)) \rangle && \text{Eq. (11)} \\ &= -(\mathcal{D}_v f)(\varphi_{v,-t}(p)) && \text{Def. 1,} \end{aligned} \quad (17)$$

and since $\varphi_{v,0} = Id$, the result follows. \square

Finally, when both are considered as operators on functions, the functional flow-map and the functional vector field operators *commute*, as formalized in the following Lemma.

Lemma 4. Let $\{\phi_{v,t} \mid t \in \mathbb{R}\}$ be the group of functional flow-maps corresponding to the one parameter group of transformations $\varphi_{v,t}$ generated by the tangent vector field v , and let \mathcal{D}_v be the corresponding functional vector field operator. Then, for any function $f : \mathcal{M} \rightarrow \mathbb{R}$, and for any $t \in \mathbb{R}$ we have:

$$\mathcal{D}_v(\phi_{v,t}f) = \phi_{v,t}(\mathcal{D}_v f). \quad (18)$$

Proof.

$$\begin{aligned} \mathcal{D}_v(\phi_{v,t}f) &= -\frac{d}{ds}\phi_{v,s}(\phi_{v,t}f)|_{s=0} && \text{Lemma 3} \\ &= -\frac{d}{ds}\phi_{v,s+t}f|_{s=0} && \text{Lemma 2} \\ &= -\frac{d}{ds}\phi_{v,t}(\phi_{v,s}f)|_{s=0} && \text{Lemma 2} \\ &= \phi_{v,t}(-\frac{d}{ds}\phi_{v,s}f)|_{s=0} && v_{v,t} \text{ independent of } s \\ &= \phi_{v,t}(\mathcal{D}_v f) && \text{Lemma 3.} \end{aligned} \quad (19)$$

\square

These properties of the functional flow-maps allow us to derive a useful PDE for the family of functions generated by the functional flow-maps.

Lemma 5. *Let $\{\phi_{v,t} \mid t \in \mathbb{R}\}$ be the group of functional flow-maps corresponding to the one parameter group of transformations $\varphi_{v,t}$ generated by the tangent vector field v . Let $f_0 : \mathcal{M} \rightarrow \mathbb{R}$, and define the time-varying function $f : \mathbb{R} \times \mathcal{M} \rightarrow \mathbb{R}$ by $f(t) = \phi_{v,t}(f_0)$. Then the following holds:*

$$\frac{d}{dt} f(t) = -\mathcal{D}_v f(t), \quad f(0) = f_0, \quad \forall t \in \mathbb{R}. \quad (20)$$

Proof.

$$\begin{aligned} \frac{d}{dt} f(t) &= \frac{d}{dt} \phi_{v,t}(f_0) = && \text{definition of } f(t) \\ &= \frac{d}{ds}|_{s=0} \phi_{v,t+s}(f_0) = \\ &= \frac{d}{ds}|_{s=0} \phi_{v,s} \phi_{v,t}(f_0) = && \text{Lemma 2} \\ &= \frac{d}{ds}|_{s=0} \phi_{v,s} f(t) = && \text{definition of } f(t) \\ &= -\mathcal{D}_v f(t) && \text{Lemma 3.} \end{aligned} \quad (21)$$

□

Functional flow-maps enable us to work with *functions* and *linear operators* as the main object of interest instead of *points* and *maps*. On smooth surfaces, the two approaches are similar. However, when considering optimization problems on discretized surfaces, represented by triangle meshes, there are inherent advantages to using functions.

First, the discretization of the space of functions is well studied, and methods such as finite elements are very effective at this task. Second, the space of functions on a surface forms a *linear vector space*, which can be effectively represented using a basis. Maps and points, on the other hand, are more difficult to deal with, due to the non-linear and non-convex constraint which requires that each point is mapped to a point that lies *on the surface*.

2.6. Lie bracket

Tangent vector fields have an additional important structure, denoted by the *Lie bracket*. Given two tangent vector fields v, w , their Lie bracket $[v, w]$ is a tangent vector field, given by (Morita, 2001, Eq.(1.9)):

$$\mathcal{D}_{[v,w]} f = \mathcal{D}_v(\mathcal{D}_w f) - \mathcal{D}_w(\mathcal{D}_v f). \quad (22)$$

When \mathcal{M} is an embedded surface we can use Lemma 1 to get a coordinate expression for the Lie bracket.

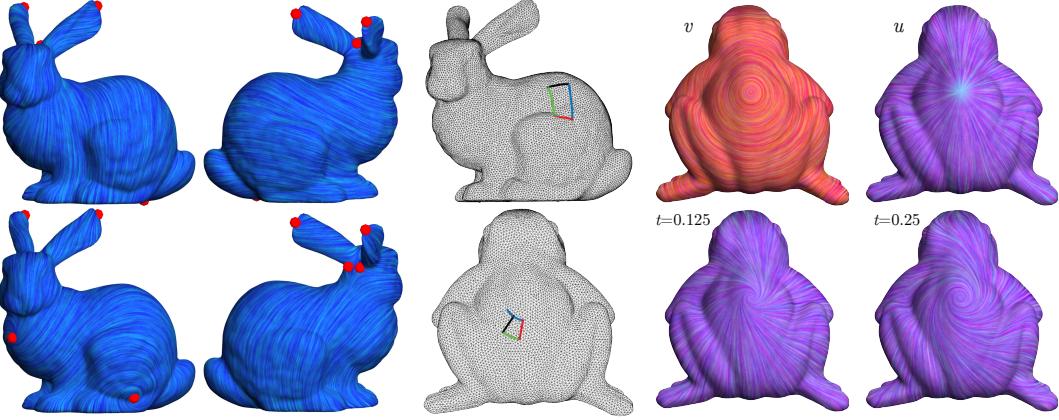


Figure 4: (left) The vector field shown at the bottom row yields a zero bracket with respect to the vector field shown at the top row. (center) The Lie bracket of two vector fields u, v encodes the failure of their flows to commute. We use the vector fields shown on the left and on the right for each shape, correspondingly. (right) Two non-commuting vector fields u, v (top) and the transport of u over the differential of the flow of v for time t (bottom). Images on the left and right are reused with permission from (Azencot et al., 2016).

Lemma 6. *Let \mathcal{M} be an embedded surface. If v, w are two tangent vector fields, then the coordinate functions of $[v, w]$ are:*

$$[v, w]_i = \mathcal{D}_v(w_i) - \mathcal{D}_w(v_i), \quad i \in [1, \dots, 3], \quad (23)$$

where v_i, w_i are the coordinate functions of v, w , respectively: $v = \sum_{i=1}^3 v_i \hat{x}_i$, and $w = \sum_{i=1}^3 w_i \hat{x}_i$.

Proof. Let $\mathcal{X} : \mathcal{M} \rightarrow \mathbb{R}^3$ be the embedding functions of \mathcal{M} , $\mathcal{X} = (x_1, x_2, x_3)$. Then, according to Lemma 1, we have:

$$[v, w]_i = \mathcal{D}_{[v, w]}(x_i) = \mathcal{D}_v(\mathcal{D}_w x_i) - \mathcal{D}_w(\mathcal{D}_v x_i) = \mathcal{D}_v(w_i) - \mathcal{D}_w(v_i). \quad (24)$$

□

The Lie bracket defines a special relation between vector fields, known as *commutation*. Two vector fields v, w commute if $[v, w] = 0$, and this holds if and only if their flow-maps commute (Lee, 2012, Thm. 9.44), namely:

$$\varphi_{v,s} \circ \varphi_{w,t} = \varphi_{w,t} \circ \varphi_{v,s}, \quad \forall s, t \in \mathbb{R}. \quad (25)$$

Figure 4 demonstrates the relationship between the bracket of two vector fields u, v and the resulting flows. We consider the two vector fields on the bunny (left), and the frog (right). The center figures show the flow line of v starting from a point p for time t (blue) followed by the flow line of u (red), and then backward

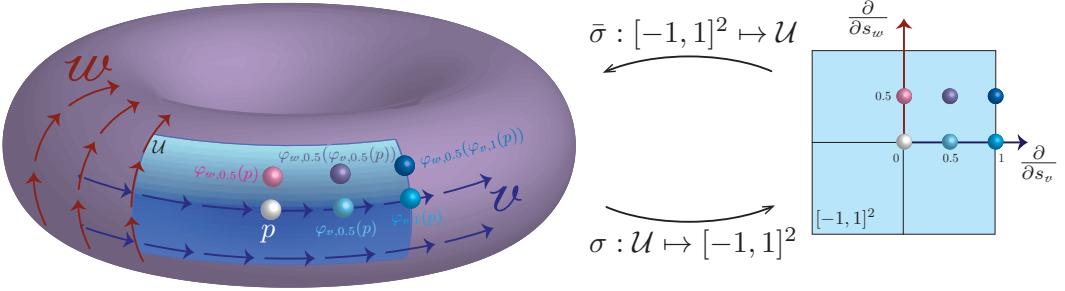


Figure 5: Illustration of the construction of $\bar{\sigma} : [-1, 1]^2 \mapsto \mathcal{U}$, the inverse of a local chart σ , using the flows of two commuting vector fields v, w in the neighborhood of a point p .

on v (green) and backward on u (black), all for time t . For the bunny, the two vector fields commute and so do their flows, “closing” the rectangle. For the frog, the vector fields do not commute, resulting in a mismatch of the original point p and the final point, leading to interesting flow behavior on the right.

In the context of applications, the Lie bracket is important because of its relation to finding local coordinates, or a *local parameterization*, on \mathcal{M} . In applications such as curvature-aligned quadrangular remeshing (Bommes et al., 2009), two tangent vector fields v, w are given, and we would like to find a local coordinate chart $\sigma : \mathcal{U} \rightarrow \mathbb{R}^2$ for a sub-domain $\mathcal{U} \subset \mathcal{M}$, such that if σ is given by the two functions (s_v, s_w) , then the vector fields v, w are mapped by $d\sigma$ to the coordinate directions $\frac{\partial}{\partial s_v}, \frac{\partial}{\partial s_w}$, respectively.

An important result (Lee, 2012, Thm. 9.46) is that if the vector fields v, w commute, then such a parameterization indeed exists. Given two linearly independent commuting tangent vector fields v, w on an open subset $\mathcal{U} \subset \mathcal{M}$, we can construct the local coordinate chart as follows (see Figure 5). Given a point $p \in \mathcal{U}$, there exists an $\epsilon > 0$ such that the map $\bar{\sigma} : (-\epsilon, \epsilon)^2 \rightarrow \mathcal{U}$, given by $\bar{\sigma}(s_v, s_w) = (\varphi_{w, s_w}(\varphi_{v, s_v}(p)))$, is a diffeomorphism, and its differential maps the coordinate directions $\frac{\partial}{\partial s_v}, \frac{\partial}{\partial s_w}$ to the tangent vector fields v, w , respectively. Then, the required coordinate chart is given by $\sigma = \bar{\sigma}^{-1}$.

3. Discrete Functional Vector Fields

In this section, we describe a discretization of the concepts that were presented in the previous Section. The discretization approach follows the original presentation of the method (Azencot et al., 2013). In this case, some (but not all) of the properties from the smooth setting carry over to the discrete setting.

3.1. Notation

In the following, M is an orientable, closed and embedded triangle mesh, where $M = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ are the sets of vertices, edges and faces, respectively, and we denote by $n = |\mathcal{V}|$ and $m = |\mathcal{F}|$ the number of vertices and faces. The embedding of the mesh in \mathbb{R}^3 is denoted by X .

Functions and vector fields. We use finite elements to represent discrete scalar functions and tangent vector fields. Specifically, we use conforming piecewise linear (PL) elements for functions, and piecewise constant (PC) elements for vector fields. Thus, $f, g, h : \mathcal{V} \rightarrow \mathbb{R}$ are column vectors of length n that represent the degrees of freedom of PL functions, and $u, v : \mathcal{F} \rightarrow \mathbb{R}^3$ are column vectors of length $3m$ that represent PC vector fields. We also occasionally use piecewise constant functions, which we denote by $\mathbf{f}, \mathbf{g}, \mathbf{h} : \mathcal{F} \rightarrow \mathbb{R}$, and they are column vectors of length m .

Rotations. We denote by $\mathcal{J}_t \in \mathbb{R}^{3 \times 3}$ the counter-clockwise rotation by $\pi/2$ in the plane of the face $t \in \mathcal{F}$, and by $\mathcal{J} \in \mathbb{R}^{3m \times 3m}$ the block diagonal matrix that encodes the rotations for all the faces.

Vectors to matrices. We use the bracket $[.]$ operator to convert vectors in \mathbb{R}^n and \mathbb{R}^m to diagonal matrices in $\mathbb{R}^{n \times n}$, $\mathbb{R}^{m \times m}$ and $\mathbb{R}^{3m \times 3m}$ respectively (replicating each entry 3 times for the latter). The dimensions of the resulting matrix is inferred from the context.

We additionally require an operator that encodes the face-wise multiplication of a vector field by a piecewise constant function, which we denote by $[.]_\bullet \in \mathbb{R}^{3m \times m}$. Specifically, if $w = [v]_\bullet \mathbf{f}$, then $w(t) = v(t)\mathbf{f}(t) \in \mathbb{R}^3, \forall t \in \mathcal{F}$. The transpose of this operator, $[.]_\bullet^T \in \mathbb{R}^{m \times 3m}$ evaluates face-wise inner products, i.e. if $\mathbf{f} = [v]_\bullet^T w$, then $\mathbf{f}(t) = \langle v(t), w(t) \rangle \in \mathbb{R}$.

Note that these operators have the following properties:

1. $[.]$ is symmetric w.r.t its input and its argument, i.e. $[f]g = [g]f$.
2. $[.]_\bullet$ is linear in its argument, i.e. $[v + w]_\bullet = [v]_\bullet + [w]_\bullet$.
3. $[.]_\bullet^T$ is symmetric w.r.t its input and its argument, i.e. $[v]_\bullet^T w = [w]_\bullet^T v$.
4. $[v]_\bullet^T [w]_\bullet = [[v]_\bullet^T w] \in \mathbb{R}^{m \times m}$.
5. $[v]_\bullet [w]_\bullet^T \in \mathbb{R}^{3m \times 3m}$ is a block diagonal matrix, with blocks of size 3×3 , that encodes for every face t the outer product $v(t)w(t)^T \in \mathbb{R}^{3 \times 3}$.
6. $[w]_\bullet \mathbf{f} = [\mathbf{f}]w \in \mathbb{R}^{3m}$.

Areas and inner products. The vector of triangle areas and vertex areas are denoted by $A_{\mathcal{F}} \in \mathbb{R}^m$ and $A_{\mathcal{V}} \in \mathbb{R}^n$, respectively. We take the vertex area as $1/3$ of the total area of its adjacent triangles. The L_2 inner product between two PL functions f, g , two PC functions \mathbf{f}, \mathbf{g} and two vector fields u, v are given by:

$$\langle f, g \rangle_{L_2} = f^T [A_{\mathcal{V}}] g, \quad \langle \mathbf{f}, \mathbf{g} \rangle_{L_2} = \mathbf{f}^T [A_{\mathcal{F}}] \mathbf{g}, \quad \langle u, v \rangle_{L_2} = u^T [A_{\mathcal{F}}] v. \quad (26)$$

Interpolation. The interpolation matrix $I_{\mathcal{V}}^{\mathcal{F}} \in \mathbb{R}^{n \times m}$ interpolates quantities from faces to the vertices, i.e., $I_{\mathcal{V}}^{\mathcal{F}}(i, j) = \frac{A_{\mathcal{F}}(j)}{3A_{\mathcal{V}}(i)}$, iff vertex i belongs to face j and 0 otherwise. Similarly, $I_{\mathcal{F}}^{\mathcal{V}}$ interpolates data from vertices to faces and is defined by $I_{\mathcal{F}}^{\mathcal{V}} = [A_{\mathcal{F}}]^{-1}(I_{\mathcal{V}}^{\mathcal{F}})^T[A_{\mathcal{V}}]$ such that $\langle f, I_{\mathcal{F}}^{\mathcal{V}}g \rangle_{L_2} = \langle I_{\mathcal{V}}^{\mathcal{F}}f, g \rangle_{L_2}$ holds.

Differential operators. We additionally require discrete gradient $\text{grad} \in \mathbb{R}^{3m \times n}$ and divergence $\text{div} \in \mathbb{R}^{n \times 3m}$ operators. Specifically we use:

$$(\text{grad } f)(t) = \frac{1}{2A_{\mathcal{F}}(t)} \sum_{v_i \in t} f_{v_i} \mathcal{J}_t e_{tv_i}, \quad t \in \mathcal{F}, \quad (27)$$

where e_{tv_i} is the directed edge opposite to the vertex v_i in the triangle $t = (v_1, v_2, v_3) \in \mathcal{F}, v_i \in \mathcal{V}$ (see Figure 6). Alternatively, this operator can be written as:

$$\text{grad} = \frac{1}{2}[A_{\mathcal{F}}]^{-1} \mathcal{J} X_{\mathcal{E}}, \quad (28)$$

where $X_{\mathcal{E}} \in \mathbb{R}^{3m \times n}$ is a matrix that encodes the edge vectors of M . Specifically, $X_{\mathcal{E}}(t, v_i) = e_{tv_i} = X(v_{i+2}) - X(v_{i+1}) \in \mathbb{R}^3, i \in 1, \dots, 3$.

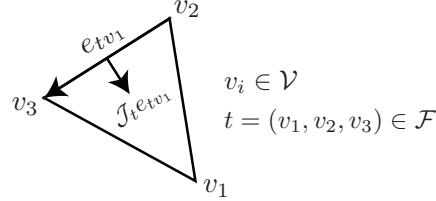


Figure 6: Illustration of the computation of the gradient operator in Equation (27).

The discrete divergence is given in terms of the gradient operator:

$$\text{div} = -[A_{\mathcal{V}}]^{-1} \text{grad}^T [A_{\mathcal{F}}]. \quad (29)$$

Note, that these operators are identical to the standard ones in geometry processing, see e.g. (Botsch et al., 2010, Chapter 3). We would like to note two important properties that follow directly from the discrete definitions of functions, vector fields, L_2 inner products and differential operators that we use.

3.1.1. Divergence free vector fields

In the smooth setting, for all $\nu = \mathcal{J} \nabla \ell$ we have that $\nabla \cdot \nu = 0$. The analogous property holds when using the discrete operators, as was shown by Polthier and Preuss (2003), i.e., let $v = \mathcal{J} \text{grad } f$ then $\text{div } v = 0$.

3.1.2. Discrete Integration by Parts

In the smooth setting, for a closed surface, the following holds (Petersen et al., 2006, pp. 382):

$$\langle \boldsymbol{v}, \nabla f \rangle_{L_2} + \langle \nabla \cdot \boldsymbol{v}, f \rangle_{L_2} = \int_M \langle \boldsymbol{v}, \nabla f \rangle da + \int_M f(\nabla \cdot \boldsymbol{v}) da = \int_M \nabla \cdot (\boldsymbol{f} \boldsymbol{v}) = 0. \quad (30)$$

Analogously, using our definitions, we have the following Lemma. It was proved by Azencot et al. (2013), and we repeat the proof here using our notation.

Lemma 7. *For a closed triangle mesh M , a function f and a vector field \boldsymbol{v} , the following holds:*

$$\langle \boldsymbol{v}, \operatorname{grad} f \rangle_{L_2} + \langle \operatorname{div} \boldsymbol{v}, f \rangle_{L_2} = \boldsymbol{v}^T [A_{\mathcal{F}}] \operatorname{grad} f + f^T [A_{\mathcal{V}}] \operatorname{div} \boldsymbol{v} = 0. \quad (31)$$

Proof. The proof follows directly from the definition of grad in Equation (29):

$$\begin{aligned} f^T [A_{\mathcal{V}}] \operatorname{div} \boldsymbol{v} &= -f^T [A_{\mathcal{V}}] [A_{\mathcal{V}}]^{-1} \operatorname{grad}^T [A_{\mathcal{F}}] \boldsymbol{v} = \\ &= -(\operatorname{grad} f)^T [A_{\mathcal{F}}] \boldsymbol{v} = -\boldsymbol{v}^T [A_{\mathcal{F}}] \operatorname{grad} f, \end{aligned} \quad (32)$$

where in the last step we used the fact that a scalar is the transpose of itself, and that $[A_{\mathcal{F}}]$ is diagonal and thus symmetric. \square

Note that we did *not* use the product rule $\nabla \cdot (\boldsymbol{f} \boldsymbol{v}) = \langle \boldsymbol{v}, \nabla \boldsymbol{f} \rangle + \boldsymbol{f}(\nabla \cdot \boldsymbol{v})$, which was used to show integration by parts in the smooth case. This is a much stronger condition, since it is defined pointwise, and in fact it *does not* hold for our definition of discrete differential operators.

3.2. Directional Derivative of Functions

Analogous to the smooth case, we can define the directional derivative of a PL function $f \in \mathbb{R}^n$ in the direction of a PC vector field $\boldsymbol{v} \in \mathbb{R}^{3m}$. Unlike the smooth case, the resulting function is *not* as smooth as f , but is instead *piecewise constant*. The directional derivative $g : \mathcal{F} \rightarrow \mathbb{R}$ is given by:

$$g(t) = \langle \boldsymbol{v}(t), (\operatorname{grad} f)(t) \rangle, \quad \forall t \in \mathcal{F}, \quad (33)$$

or equivalently:

$$g = [\boldsymbol{v}]^T_{\bullet} \operatorname{grad} f \in \mathbb{R}^m. \quad (34)$$

In the discrete case, the directional derivative function g is *not* in the same space as the input function f , and therefore the linear operator that encodes directional derivatives is *rectangular*. In the following, we use an interpolation approach to address this issue, whereas in the next Section we consider a different discretization.

3.3. Functional Vector Fields

Definition 5. Given a tangent vector field v on M , we define two corresponding functional vector field operators:

$$D_v^{\mathcal{F}} = [v]_{\bullet}^T \text{grad} \in \mathbb{R}^{m \times n}, \quad (35)$$

$$D_v^{\mathcal{V}} = I_{\mathcal{V}}^{\mathcal{F}} D_v^{\mathcal{F}} \in \mathbb{R}^{n \times n}. \quad (36)$$

The operator $D_v^{\mathcal{F}}$ does not use interpolation, and thus retains all the information about v . However, this operator is *rectangular*, and therefore cannot be used as is in settings which require *square* matrices, e.g. for flow-maps. In such settings, we will use $D_v^{\mathcal{V}}$.

In the discrete case, the functional vector field operators are simply *matrices*, and therefore linear in their function argument with respect to constants.

Lemma 8. The sparsity structure of $D_v^{\mathcal{F}}$ and $D_v^{\mathcal{V}}$ are given by the following.

1. If the entry (t, i) , $t \in \mathcal{F}$, $i \in \mathcal{V}$ in $D_v^{\mathcal{F}}$ is non-zero, then $i \in t$, i.e. the vertex i belongs to the triangle t .
2. If the entry (i, j) , $i, j \in \mathcal{V}$ in $D_v^{\mathcal{V}}$ is non-zero, then either $i = j$ or $(i, j) \in \mathcal{E}$, i.e. (i, j) is an edge in M .

Proof.

1. Let $f_i \in \mathbb{R}^n$ be a PL function such that $f_i(j) = 1$ if and only if $i = j$ and 0 otherwise, and let $g_i = D_v^{\mathcal{F}} f_i$. If the entry (t, i) is non-zero, then $g_i(t)$ is non-zero. By the definition of $D_v^{\mathcal{F}}$, we have that $g_i(t) = \langle v(t), (\text{grad } f_i)(t) \rangle$. Since $\text{grad } f_i$ is non zero only for triangles t which are neighbors of the vertex i , the result follows.
2. If the entry (i, j) is not zero, then there should exist an entry (i, t) in $I_{\mathcal{V}}^{\mathcal{F}}$ and an entry (t, j) in $D_v^{\mathcal{F}}$ which are not zero, for some triangle $t \in \mathcal{F}$. By the definition of $I_{\mathcal{V}}^{\mathcal{F}}$ and the first part of the proof, this implies that both i and j belong to the triangle t . This is possible only if $i = j$ or $(i, j) \in \mathcal{E}$.

□

Definition 6. Given a function $f : \mathcal{V} \rightarrow \mathbb{R}$, the corresponding dual functional vector field operators are given by:

$$\overline{D}_f^{\mathcal{F}} = [\text{grad } f]_{\bullet}^T \in \mathbb{R}^{m \times 3m}, \quad (37)$$

$$\overline{D}_f^{\mathcal{V}} = I_{\mathcal{V}}^{\mathcal{F}} \overline{D}_f^{\mathcal{F}} \in \mathbb{R}^{n \times 3m}. \quad (38)$$

Duality. As in the smooth case, we have that $\overline{D}_f^{\mathcal{F}}(v) = D_v^{\mathcal{F}}(f)$, since $[\text{grad } f]_{\bullet}^T v = [v]_{\bullet}^T \text{grad } f$, and similarly for $\overline{D}_f^{\mathcal{V}}$.

Linearity. Linearity with respect to constants follows since the operators are matrices. Linearity with respect to functions, however, is more subtle in the discrete case, and holds only for $\overline{D}_f^{\mathcal{F}}$. We have that:

$$\overline{D}_h^{\mathcal{F}}([v]_{\bullet} f + [u]_{\bullet} g) = [\text{grad } h]_{\bullet}^T [v]_{\bullet} f + [\text{grad } h]_{\bullet}^T [u]_{\bullet} g = [\overline{D}_h^{\mathcal{F}} v] f + [\overline{D}_h^{\mathcal{F}} u] g, \quad (39)$$

for any two tangent vector fields u, v , and functions $h : \mathcal{V} \rightarrow \mathbb{R}$, $f, g : \mathcal{F} \rightarrow \mathbb{R}$.

Reconstruction. In the discrete case, in general, full reconstruction given the embedding X is possible only for $D_v^{\mathcal{F}}$. We have:

Lemma 9. *Let v be a tangent vector field on M , and let $D_v^{\mathcal{F}}$ be its corresponding functional vector field operator. Further, denote the embedding of M by $X : \mathcal{V} \rightarrow \mathbb{R}^3$, where X is given by the three coordinate functions $X = (x_1, x_2, x_3)$. Then:*

$$D_v^{\mathcal{F}}(x_i) = v_i, \quad i \in (1, \dots, 3), \quad (40)$$

where $v_i \in \mathbb{R}^m$ are the coordinate functions of v .

Proof. Let \mathcal{M}_t denote the plane of the triangle t , and extend $v(t)$ to a constant vector field v_t on that plane. Since \mathcal{M}_t is a smooth surface, and v_t a smooth vector field, we can apply Lemma 1, and deduce that $\mathcal{D}_{v_t}(x_i) = v_{ti}$. Since v_t is constant, $\mathcal{D}_{v_t}(x_i)(p) = D_v^{\mathcal{F}}(x_i)(t) = v_{ti}(p) = v_i(t)$, $\forall t \in \mathcal{F}, p \in \mathcal{M}_t$. \square

3.4. Flow-Maps

Carrying the flow-map definition presented in Definition 3 to the discrete case introduces some challenges, due to the discontinuity of piecewise constant vector fields at mesh edges. In fact, tracing vector fields robustly on a triangle mesh is a separate and important research topic, see e.g. the work by Ray et al. (Ray and Sokolov, 2014) and citations within. We therefore avoid defining discrete flow maps, and define directly discrete *functional* flow-maps.

3.5. Functional Flow-maps

In Lemma 5, we have seen that given a one parameter group of transformations $\{\varphi_{v,t} \mid t \in \mathbb{R}\}$ generated by the tangent vector field v , its corresponding group of functional flow-maps $\phi_{v,t}$, and a function \mathcal{f}_0 , then the following holds:

$$\frac{d}{dt} \mathcal{f}(t) = -\mathcal{D}_v \mathcal{f}(t), \quad \mathcal{f}(0) = \mathcal{f}_0, \quad \forall t \in \mathbb{R}, \quad (41)$$

where $\mathcal{f}(t) = \phi_{v,t}(\mathcal{f}_0)$.

In the discrete setting, we mimic functional flow maps, by defining operators that generate solutions to the analogous discrete equation.

Definition 7. Let v be a tangent vector field on M , and $D_v^{\mathcal{V}}$ its corresponding functional vector field. The discrete functional flow map of v is given by:

$$\Phi_{v,t} = \exp(-tD_v^{\mathcal{V}}) \in \mathbb{R}^{n \times n}, \quad \forall t \in \mathbb{R}, \quad (42)$$

where \exp is the matrix exponential.

Lemma 10. Let $\{\Phi_{v,t} | t \in \mathbb{R}\}$ be as in Equation (42), for the tangent vector field v . Let $f_0 : \mathcal{V} \rightarrow \mathbb{R}$, and define the time-varying function $f : \mathbb{R} \times \mathcal{V} \rightarrow \mathbb{R}$ by $f(t) = \Phi_{v,t} f_0$. Then the following holds:

$$\frac{d}{dt} f(t) = -D_v^{\mathcal{V}} f(t), \quad f(0) = f_0, \quad \forall t \in \mathbb{R}. \quad (43)$$

Proof. The system of linear differential equations in Equation (43) is a matrix ODE, which is known to have a unique solution (Godunov, 1997, Thm 1), given by (Hochbruck and Ostermann, 2010, Eq. (1.4)) $f(t) = \exp(-tD_v^{\mathcal{V}}) f_0 = \Phi_{v,t} f_0$. \square

Discrete functional flow-maps allow us to use flow-related properties *without explicitly computing the discrete flow-map*. Furthermore, all the operations are not geometric, but algebraic, thus we can leverage various algebraic tools such as spectral analysis, representation in a reduced basis, etc. While computing the matrix exponential is challenging numerically (Moler and Van Loan, 2003), there exist efficient methods for computing *the product of the matrix exponential with a vector* (Al-Mohy and Higham, 2011), which is the only computation that we need in order to compute the functional flow of a given function f_0 . The flow in Figure 3 for example, was generated using this approach.

Using this definition, additional properties of functional flow maps carry over to the discrete case, due to the properties of the matrix exponential.

Lemma 11. Let $\{\Phi_{v,t} | t \in \mathbb{R}\}$ be the group of functional flow-maps generated by the tangent vector field v . Then the following hold:

1. Identity at 0: $\Phi_{v,0} = Id$.
2. Composition: $\Phi_{v,t} \Phi_{v,s} = \Phi_{v,t+s}$, $\forall t, s \in \mathbb{R}$.
3. Derivative at 0: $\frac{d}{dt} \Phi_{v,t} \Big|_{t=0} = -D_v^{\mathcal{V}}$.
4. Commutativity: $D_v^{\mathcal{V}} \Phi_{v,t} = \Phi_{v,t} D_v^{\mathcal{V}}$.

Proof. All the properties follow directly from $\Phi_{v,t} = e^{-tD_v^{\mathcal{V}}}$.

1. Trivial since $e^0 = Id$.
2. Follows from $e^{-tD_v^{\mathcal{V}}} e^{-sD_v^{\mathcal{V}}} = e^{-(t+s)D_v^{\mathcal{V}}}$.
3. We have that $e^{-tA} = Id - tA + \frac{1}{2}t^2 A^2 + O(t^3)$, thus $\frac{d}{dt} e^{-tA} \Big|_{t=0} = -A$.

4. We have that $e^{-tA} = \sum_{k=0}^{\infty} \frac{1}{k!} (-tA)^k$, therefore $Ae^{-tA} = \sum_{k=0}^{\infty} \frac{1}{k!} (-t)^k A^{k+1} = e^{-tA}A$.

□

In addition to the properties we have shown in the smooth case, additional properties are easy to show in the discrete finite dimensional case. For example, we have the following:

Lemma 12. *Let $\{\Phi_{v,t} | t \in \mathbb{R}\}$ be the group of functional flow-maps generated by the tangent vector field v , and let $O \in \mathbb{R}^{n \times n}$ be a real matrix. Then the following holds:*

$$D_v^{\mathcal{V}} O = O D_v^{\mathcal{V}} \quad \Leftrightarrow \quad \Phi_{v,t} O = O \Phi_{v,t} \quad \forall t \in \mathbb{R}. \quad (44)$$

Proof.

⇒ If $D_v^{\mathcal{V}} O = O D_v^{\mathcal{V}}$ then it is easy to see by induction on k that for all $k \geq 0$ we have $(D_v^{\mathcal{V}})^k O = O (D_v^{\mathcal{V}})^k$. Hence, O also commutes with polynomials in $D_v^{\mathcal{V}}$ and the result follows.

⇐ Taking the derivative with respect to t on both sides at $t = 0$, we have:

$$\begin{aligned} \frac{d}{dt} \Phi_{v,t} O|_{t=0} &= \frac{d}{dt} O \Phi_{v,t}|_{t=0} & \forall t \in \mathbb{R} \\ \frac{d}{dt} \Phi_{v,t}|_{t=0} O &= O \frac{d}{dt} \Phi_{v,t}|_{t=0} & O \text{ does not depend on } t \\ -D_v^{\mathcal{V}} O &= -O D_v^{\mathcal{V}} & \text{Lemma 11(3)} \end{aligned} \quad (45)$$

□

3.6. Lie bracket

Analogously to the smooth case, we define the Lie bracket operator of two tangent vector fields u, v on M as the commutator of their *square* functional vector field operators:

$$D_{[v,w]}^{\mathcal{V}} = D_v^{\mathcal{V}} D_w^{\mathcal{V}} - D_w^{\mathcal{V}} D_v^{\mathcal{V}}. \quad (46)$$

As in the smooth case, we have the relationship between the functional flow maps and the Lie bracket operator.

Lemma 13. *The functional flow maps of two vector fields commute if and only if their Lie bracket vanishes:*

$$\Phi_{v,t} \Phi_{w,s} = \Phi_{w,s} \Phi_{v,t}, \quad \forall s, t \in \mathbb{R} \quad \Leftrightarrow \quad D_{[v,w]}^{\mathcal{V}} = 0. \quad (47)$$

Proof. First, set $O = \Phi_{w,s}$, and apply Lemma 12, to get that:

$$\Phi_{v,t}\Phi_{w,s} = \Phi_{w,s}\Phi_{v,t}, \quad \forall s, t \in \mathbb{R} \quad \Leftrightarrow \quad D_v^{\mathcal{V}}\Phi_{w,s} = \Phi_{w,s}D_v^{\mathcal{V}}, \quad \forall s \in \mathbb{R}. \quad (48)$$

Now, set $O = D_v^{\mathcal{V}}$, and apply Lemma 12 again, yielding:

$$D_v^{\mathcal{V}}\Phi_{w,s} = \Phi_{w,s}D_v^{\mathcal{V}}, \quad \forall s \in \mathbb{R} \quad \Leftrightarrow \quad D_v^{\mathcal{V}}D_w^{\mathcal{V}} = D_w^{\mathcal{V}}D_v^{\mathcal{V}}. \quad (49)$$

□

Unfortunately, unlike in the smooth case, the operator $D_{[v,w]}^{\mathcal{V}}$ is *not* a functional vector field. This is easy to see when considering the sparsity structure of $D_v^{\mathcal{V}}$. As shown in Lemma 8, $D_v^{\mathcal{V}}$ has a 1-ring neighborhood sparsity structure. When two such matrices are multiplied, the resulting matrix has a 2-ring neighborhood sparsity structure, and therefore, cannot in general be a functional vector field. In the smooth case, these second order dependencies cancel out, however in the discrete case they do not.

To extract a vector field from $D_{[v,w]}^{\mathcal{V}}$ it is still possible to use the embedding and compute $u = D_{[v,w]}^{\mathcal{V}}X$, where $u \in \mathbb{R}^{n \times 3}$. However, this yields a vector field on the *vertices*, which is only approximately tangent to the input surface. This approach was used by Azencot et al. (2018) to numerically simulate the EPDiff flow on a triangulated mesh.

4. Divergence-based Functional Vector Fields

As we have seen, discrete functional vector fields are quite powerful, allowing us to represent complicated geometric objects such as the flow map, using simple algebraic operations on matrices, such as computing the matrix exponential.

There are however some properties of the smooth case that do not carry over nicely to the discrete case. One such example is the Lie bracket described in Section 3.6, which fails to yield a discrete functional vector field as in the smooth case. In this section, we propose an *additional discretization* of the functional vector field operator, which, combined with the previous one, allows us to alleviate some of the inconsistencies in the discrete case.

4.1. Smooth DFVF

A well-known result from Riemannian geometry (see e.g., (Petersen et al., 2006, pg. 58)), the product rule for the product of functions and vector fields, is given by:

$$\mathcal{f} \nabla \cdot \mathbf{v} + \langle \nabla \mathcal{f}, \mathbf{v} \rangle = \nabla \cdot (\mathcal{f} \mathbf{v}), \quad (50)$$

where this equation holds for any point $p \in \mathcal{M}$. Thus, instead of directly working with gradients of functions, we define the following.

Definition 8. Given a tangent vector field v on \mathcal{M} , the corresponding divergence-based functional vector field (DFVF) operator $\tilde{\mathcal{D}}_v$ is given by:

$$\tilde{\mathcal{D}}_v(\mathcal{f}) = \nabla \cdot (\mathcal{f}v) - \mathcal{f}\nabla \cdot v. \quad (51)$$

In the smooth case, the definitions in equations (2) and (51) are equivalent, i.e. $\mathcal{D}_v(\mathcal{f}) = \tilde{\mathcal{D}}_v(\mathcal{f})$ for all functions \mathcal{f} . In the discrete case, however, this choice leads to different discrete operators.

4.2. Discrete DFVF

Definition 9. Given a tangent vector field v on M , the corresponding divergence-based functional vector field (DFVF) operators are given by:

$$\begin{aligned} \tilde{D}_v^{\mathcal{F}} &= \text{div}[v]_{\bullet} - [\text{div } v]I_{\mathcal{V}}^{\mathcal{F}} \in \mathbb{R}^{n \times m}, \\ \tilde{D}_v^{\mathcal{V}} &= \text{div}[v]_{\bullet}I_{\mathcal{F}}^{\mathcal{V}} - [\text{div } v] \in \mathbb{R}^{n \times n}. \end{aligned} \quad (52)$$

The corresponding dual operators are given by:

$$\begin{aligned} \overline{\tilde{D}_f^{\mathcal{F}}} &= \text{div}[f] - [I_{\mathcal{V}}^{\mathcal{F}}f] \text{div} \in \mathbb{R}^{n \times 3m}, \\ \overline{\tilde{D}_f^{\mathcal{V}}} &= \text{div}[I_{\mathcal{F}}^{\mathcal{V}}f] - [f] \text{div} \in \mathbb{R}^{n \times 3m}, \end{aligned} \quad (53)$$

and we have: $\tilde{D}_v^{\mathcal{F}}f = \overline{\tilde{D}_f^{\mathcal{F}}}v$ and $\tilde{D}_v^{\mathcal{V}}f = \overline{\tilde{D}_f^{\mathcal{V}}}v$.

The main advantage of the divergence-based rectangular operator, is that it can be multiplied with the rectangular functional vector field operator that we have introduced earlier, to yield a better discretization of the Lie bracket operator.

4.3. Mixed Lie bracket operator

Definition 10. The mixed Lie bracket operator of two tangent vector fields v, w is given by:

$$D_{[v,w]}^{\mathcal{F}} = \tilde{D}_v^{\mathcal{F}}D_w^{\mathcal{F}} - \tilde{D}_w^{\mathcal{F}}D_v^{\mathcal{F}} \in \mathbb{R}^{n \times n}. \quad (54)$$

The mixed lie bracket operator uses both the discrete divergence and discrete gradient operators, to yield a better discretization, that has some interesting properties.

Lemma 14. The operator $D_{[v,w]}^{\mathcal{F}}$ can be written as:

$$D_{[v,w]}^{\mathcal{F}} = \text{div } B(v, w) \text{grad} - [\text{div } v]D_w^{\mathcal{V}} + [\text{div } w]D_v^{\mathcal{V}}, \quad (55)$$

where $B(v, w) = [v]_{\bullet}[w]_{\bullet}^T - [w]_{\bullet}[v]_{\bullet}^T$ is a $3m \times 3m$ block diagonal matrix, encoding the outer products $v(t)w(t)^T - w(t)v(t)^T$ per face $t \in \mathcal{F}$.

Proof.

$$\begin{aligned}
D_{[v,w]}^{\mathcal{F}} &= (\operatorname{div}[v]_{\bullet} - [\operatorname{div} v]I_{\mathcal{V}}^{\mathcal{F}})[w]_{\bullet}^T \operatorname{grad} - (\operatorname{div}[w]_{\bullet} - [\operatorname{div} w]I_{\mathcal{V}}^{\mathcal{F}})[v]_{\bullet}^T \operatorname{grad} \\
&= \operatorname{div}[v]_{\bullet}[w]_{\bullet}^T \operatorname{grad} - [\operatorname{div} v]I_{\mathcal{V}}^{\mathcal{F}}[w]_{\bullet}^T \operatorname{grad} - \operatorname{div}[w]_{\bullet}[v]_{\bullet}^T \operatorname{grad} + [\operatorname{div} w]I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \operatorname{grad} \\
&= \operatorname{div}([v]_{\bullet}[w]_{\bullet}^T - [w]_{\bullet}[v]_{\bullet}^T) \operatorname{grad} - [\operatorname{div} v]I_{\mathcal{V}}^{\mathcal{F}}[w]_{\bullet}^T \operatorname{grad} + [\operatorname{div} w]I_{\mathcal{V}}^{\mathcal{F}}[v]_{\bullet}^T \operatorname{grad} \\
&= \operatorname{div}([v]_{\bullet}[w]_{\bullet}^T - [w]_{\bullet}[v]_{\bullet}^T) \operatorname{grad} - [\operatorname{div} v]D_w^{\mathcal{V}} + [\operatorname{div} w]D_v^{\mathcal{V}} \\
&= \operatorname{div} B(v, w) \operatorname{grad} - [\operatorname{div} v]D_w^{\mathcal{V}} + [\operatorname{div} w]D_v^{\mathcal{V}}.
\end{aligned} \tag{56}$$

□

We can further simplify the operator B as follows.

Lemma 15. *Let $B(v, w)$ be the operator defined in Lemma 14. Then the following holds:*

$$B(v, w) = [[v]_{\bullet}^T \mathcal{J} w] \mathcal{J}. \tag{57}$$

Proof. By the definition in Lemma 14 we have that $B(v, w)$ at the triangle $t \in \mathcal{F}$ depends only on the values of v, w at t . Let $\xi_i \in \mathbb{R}^3, i \in 1..3$ be an orthonormal frame at the triangle t , such that ξ_3 is in the normal direction, and set $\mathcal{B} = (\xi_1 \ \xi_2 \ \xi_3)$. Note, that the rotation by $\pi/2$ in the triangle t is given by $\mathcal{J}_t = \mathcal{B} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathcal{B}^T$. Now, let $v(t) = \mathcal{B} \begin{pmatrix} a \\ b \\ 0 \end{pmatrix}, w(t) = \mathcal{B} \begin{pmatrix} c \\ d \\ 0 \end{pmatrix}$ be the representations of $v(t), w(t)$ in this basis, respectively. Then $B(v, w)(t)$ given by

$$\begin{aligned}
B(v, w)(t) &= \mathcal{B} \left(\begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \begin{pmatrix} c & d & 0 \end{pmatrix} - \begin{pmatrix} c \\ d \\ 0 \end{pmatrix} \begin{pmatrix} a & b & 0 \end{pmatrix} \right) \mathcal{B}^T \\
&= \mathcal{B} \begin{pmatrix} 0 & ad-bc & 0 \\ bc-ad & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathcal{B}^T \\
&= (bc - ad) \mathcal{B} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathcal{B}^T \\
&= (a \ b \ 0) \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} c \\ d \\ 0 \end{pmatrix} \mathcal{J}_t \\
&= (a \ b \ 0) \mathcal{B}^T \mathcal{B} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathcal{B}^T \mathcal{B} \begin{pmatrix} c \\ d \\ 0 \end{pmatrix} \mathcal{J}_t \\
&= v(t)^T \mathcal{J}_t w(t) \mathcal{J}_t \\
&= \langle v(t), \mathcal{J}_t w(t) \rangle \mathcal{J}_t.
\end{aligned} \tag{58}$$

Since this holds for every triangle t , the result follows by the definitions of the $[\cdot]_{\bullet}^T$ and $[\cdot]$ operators. □

4.3.1. Sparsity structure

Using Lemma 14, we can deduce the sparsity structure of the mixed Lie bracket operator.

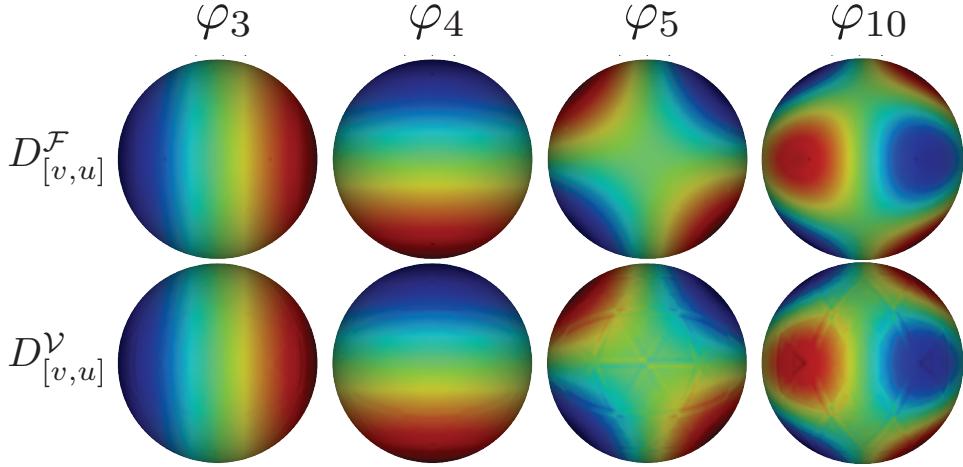


Figure 7: Applying the Lie bracket operator $D_{[v,u]}^V$ to the i -th eigenfunction φ_i of the Laplace-Beltrami operator, leads to visible oscillations (bottom), whereas when using the mixed Lie bracket $D_{[v,u]}^F$ the results are smoother (top).

Lemma 16. *The mixed Lie bracket operator $D_{[v,w]}^F$ has a 1-ring sparsity structure. I.e, if the entry (i,j) is non-zero, then either $i = j$ or $(i,j) \in \mathcal{E}$.*

Proof. The first component, up to diagonal mass matrices and constants, has the structure of $\text{grad}^T B(v,w) \text{grad}$. As we have shown earlier, grad has a non-zero entry (t,i) only if $i \in t$. The matrix B has no effect on the sparsity structure, since it is block diagonal, and does not have mixed entries for two different triangles. Hence, if (i,j) is non-zero in the first component, then there exists a triangle t such that $i \in t$ and $j \in t$, therefore either $i = j$ or $(i,j) \in \mathcal{E}$. Therefore, the first component has a 1-ring sparsity structure. The last two components are formed by the multiplication of a diagonal matrix with a 1-ring sparse matrix, according to Lemma 8, and therefore are also 1-ring sparse. Hence, $D_{[v,w]}^F$ is the sum of 1-ring sparse matrices, and the result follows. \square

The sparsity structure of the Lie bracket operator is of great importance in practice. For example, when applying the operator to a smooth function (e.g. the first few eigenfunctions of the Laplace-Beltrami operator), the mixed Lie bracket operator $D_{[v,u]}^F$ yields a smooth function, whereas the operator from Azencot et al. (2013), $D_{[v,u]}^V$, exhibits visible oscillations (see Figure 7).

Furthermore, one of the most useful applications of the functional vector field operator is transporting functions on flows of vector fields using the functional flow map, e.g., as was done in Azencot et al. (2014b, 2015b) for fluid simulation. This computation is done, as described in Definition 7, by evaluating $f(t) = e^{-tD_v} f_0$. Thus, powers of D_v are involved in the computation, and the wrong sparsity

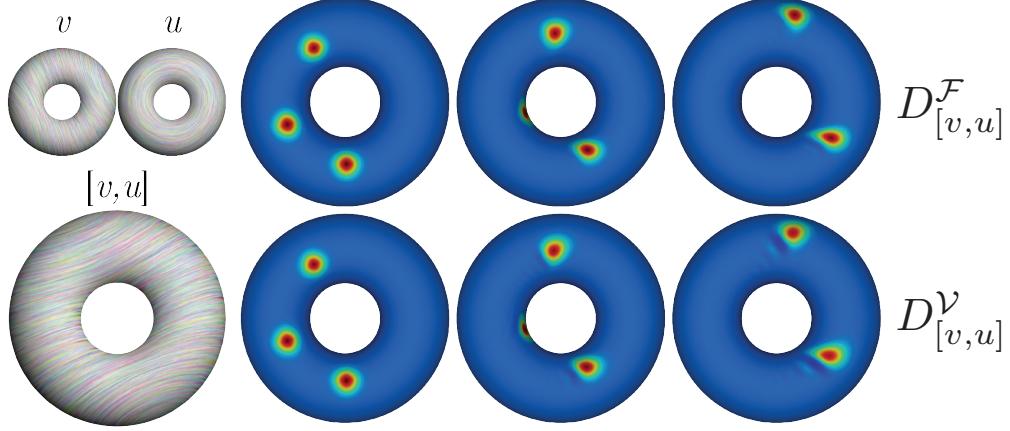


Figure 8: Transporting a function using the bracket operator $D_{[v,u]}^V$ leads to more discretization error (bottom) compared to computing transport using the mixed operator $D_{[v,u]}^F$ (top).

structure can generate considerable discretization noise during function transport, as can be seen in Figure 8.

4.3.2. The Lie bracket of divergence-free vector fields

Consider the special case when the smooth vector fields v, u are gradients of functions rotated in the tangent plane by $\pi/2$, namely $v = \mathcal{J} \nabla f, u = \mathcal{J} \nabla g$. Then, it is well known (Arnold, 1989, Appendix 2) that $[v, u] = J \nabla \{f, g\}$, where $\{f, g\} = \langle \mathcal{J} \nabla f, \nabla g \rangle$ is known as the *Poisson bracket* of the functions f, g . Hence in the smooth case, the following holds:

$$\mathcal{D}_{[v,u]} h = \langle \nabla h, \mathcal{J} \nabla \{f, g\} \rangle = -\tilde{\mathcal{D}}_{\mathcal{J} \nabla h} \{f, g\}, \quad (59)$$

where the first equality is by the definition of \mathcal{D} , and the second by multiplying both sides of the inner product by \mathcal{J} and using the definition again.

In the discrete setting using our operators, we have the following result.

Theorem 1. *Let $v = \mathcal{J} \operatorname{grad} f, u = \mathcal{J} \operatorname{grad} g$, and define $\{f, g\} = \langle \mathcal{J} \operatorname{grad} f, \operatorname{grad} g \rangle$, analogously to the smooth case. Then the following holds:*

$$D_{[v,u]}^F h = -\tilde{\mathcal{D}}_{\mathcal{J} \operatorname{grad} h}^F \{f, g\}. \quad (60)$$

Proof. First, note that $\operatorname{div} v = \operatorname{div} u = 0$, as was discussed in section 3.1.1. Hence, the last two addends in Equation (55) are zero. Thus, from Lemma 15, for any

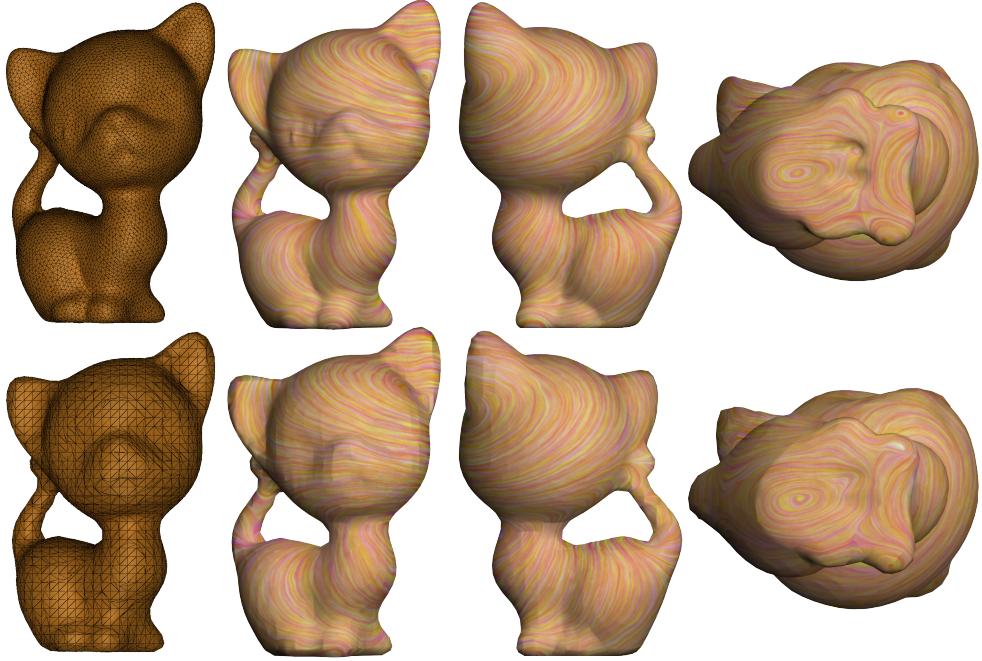


Figure 9: The extrinsic extraction of the commutator vector field yields similar results on a uniformly tessellated mesh (top) and a highly non uniform tessellation of the same model (bottom).

function h we have,

$$\begin{aligned}
 D_{[v,u]}^{\mathcal{F}} h &= \operatorname{div}([v]^T \mathcal{J} u] \mathcal{J}) \operatorname{grad} h \\
 &= \operatorname{div}([\mathcal{J} \operatorname{grad} f]^T \mathcal{J} \mathcal{J} \operatorname{grad} g] \mathcal{J}) \operatorname{grad} h \\
 &= -\operatorname{div}[\mathcal{J} \operatorname{grad} f]^T \operatorname{grad} g] \mathcal{J} \operatorname{grad} h \\
 &= -\operatorname{div}[\{f, g\}] \mathcal{J} \operatorname{grad} h \\
 &= -\operatorname{div}[\mathcal{J} \operatorname{grad} h]_{\bullet} \{f, g\} \\
 &= -\tilde{D}_{\mathcal{J} \operatorname{grad} h}^{\mathcal{F}} \{f, g\},
 \end{aligned} \tag{61}$$

where in the step before last we used Property 6 of the bracket operators (see Section 3.1), and in the last step we used again the fact that $\operatorname{div} \mathcal{J} \operatorname{grad} h = 0$. \square

4.3.3. Extraction of the commutator vector field

Given the operator $D_{[v,w]}^{\mathcal{F}}$, it is possible to extract a corresponding vector field by taking $u = D_{[v,w]}^{\mathcal{F}} X \in \mathbb{R}^{n \times 3}$. While this procedure results in a vector field on the *vertices*, which is not guaranteed to be tangent to the surface, in practice this procedure seems to yield a satisfactory result. We show in Figure 9 the robustness this extraction to the triangulation of the mesh. Given two fields on

a smooth mesh, we compute their bracket and show its reconstruction in the top row. The bottom row shows the same computation, but on a low resolution uniform sampling of the original mesh, which contains many skinny triangles. Note that the resulting vector fields are similar in both cases.

4.3.4. Limitation: loss of group structure

While the sparsity structure of $D_{[v,u]}^{\mathcal{F}}$ is the correct one, we do not have a result linking it back to a corresponding vector field w , such that $D_w^{\mathcal{V}} = D_{[v,u]}^{\mathcal{F}}$. Furthermore, the result from Lemma 13 that links a vanishing Lie bracket to the commutation of the functional flows also does not hold, since the separate flows are no longer well defined. It remains an open question whether one can design discrete functional vector fields, and a corresponding Lie bracket operator, such that both the group structure, and the closure under bracket properties hold.

4.4. The Lie bracket as a linear transformation on vector fields

In some applications, instead of computing $D_{[v,w]}^{\mathcal{F}}$ for two given vector fields, we wish to compute a vector field w whose bracket with v fulfills some property, e.g. its norm is minimal. We therefore define the following operator.

Definition 11. Given a vector field v , the operator $\text{ad}_v : \mathbb{R}^{3m} \rightarrow \mathbb{R}^{n \times 3}$ is given by:

$$\text{ad}_v(w) = D_{[v,w]}^{\mathcal{F}} X. \quad (62)$$

Note that ad_v is linear, and can be written in the following concise form.

Lemma 17. The operator ad_v can be written as:

$$\text{ad}_v = \begin{pmatrix} \tilde{D}_v^{\mathcal{F}} & 0 & 0 \\ 0 & \tilde{D}_v^{\mathcal{F}} & 0 \\ 0 & 0 & \tilde{D}_v^{\mathcal{F}} \end{pmatrix} - \begin{pmatrix} \overline{\tilde{D}_{v_1}^{\mathcal{F}}} \\ \overline{\tilde{D}_{v_2}^{\mathcal{F}}} \\ \overline{\tilde{D}_{v_3}^{\mathcal{F}}} \end{pmatrix} \in \mathbb{R}^{3n \times 3m}, \quad (63)$$

where $v_i \in \mathbb{R}^m$ are the coordinate functions of v .

Proof. From Lemma 9, we have that $D_v^{\mathcal{F}} x_i = v_i$, $i \in 1, \dots, 3$. Hence, from Definitions 9 and 10 we have:

$$D_{[v,w]}^{\mathcal{F}} x_i = \tilde{D}_v^{\mathcal{F}} w_i - \tilde{D}_w^{\mathcal{F}} v_i = \tilde{D}_v^{\mathcal{F}} w_i - \overline{\tilde{D}_{v_i}^{\mathcal{F}}} w. \quad (64)$$

The result follows, by considering all three components, and taking $w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$. \square

Note, that one could potentially apply interpolation and projection on ad_v to obtain a vector field on the faces.

4.4.1. Application: commutation-guided rescaling

In parameterization applications it is often required that the parameterization aligns with two direction fields, which are often computed from the curvature directions of the surface. However, given two arbitrary vector fields, their flows generate a parameterization only if they commute. Hence, we can formulate the following optimization problem for the smooth case. Given two vector fields v, u find a smooth scalar function φ such that $\|[v, \varphi u]\|_{L^2}$ is minimal:

$$\min_{\varphi} \left\{ \frac{1}{2} \int_{\mathcal{M}} \| [v, \varphi u] \|^2 \, d\alpha + \frac{\alpha}{2} \int_{\mathcal{M}} |\nabla \varphi|^2 \, d\alpha \right\} \quad (65)$$

$$\text{subject to } \int_{\mathcal{M}} |\varphi|^2 \, d\alpha = 1. \quad (66)$$

Our discrete framework allows for an efficient evaluation of the above expression by computing the eigenvectors of the matrix:

$$E = F(v, u)^T F(v, u) + \alpha L ,$$

where $F(v, u)f$ discretizes the smooth action on functions $[v, \varphi u]$, and is given by $F(v, u) = \text{ad}_v[u] \bullet I_{\mathcal{F}}^{\mathcal{V}}$, and L is the standard Laplace–Beltrami operator, given by $L = -\text{div grad}$.

Figure 10 illustrates the result this procedure. Here, v is a smooth vector field with unit norm, and u is its rotation by $\pi/2$. The two fields have non-zero bracket, as the color coding of the norm of $[v, u]$ shows, where the bracket’s norm is concentrated around the singularity locations. Using the optimization problem mentioned previously, we find a function φ which minimizes $\|[v, \varphi u]\|_{L^2}^2$. The resulting norm is smaller and more evenly distributed, as can be seen in the rightmost image (see e.g. the tips of the fingers).

4.4.2. Application: As-Commuting-As-Possible vector fields

Instead of optimizing for a scalar when two directions are given as above, we are also interested in finding a set of vector fields which commute with respect to another vector field v . Since the linear operator ad_v encodes the bracket operation on vector fields, we can easily extract commuting fields by computing its kernel. Notice that v itself lies in the discrete kernel, by construction. Unfortunately, this is also true for any vector field which is parallel to v . As the set of discrete commuting fields contains many non-smooth vector fields (e.g., any subset of vectors in v can be rotated by π), we add a regularizer term to the energy expression. We

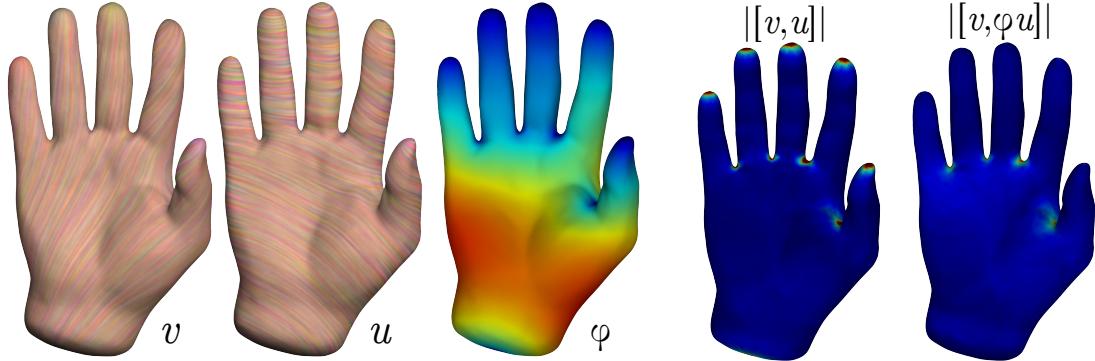


Figure 10: Two smooth vector fields with non zero bracket $[v, u]$ and a smooth rescaling function φ which minimizes $\|[v, \varphi u]\|_{L^2}^2$. Note that after rescaling the bracket's norm is smaller and more evenly distributed on the surface (see e.g. the tips of the fingers).

formulate the following optimization problem for the smooth case:

$$\begin{aligned} \min_u & \left\{ \frac{1}{2} \int_{\mathcal{M}} \|\text{ad}_v u\|^2 da + \frac{\lambda}{2} \int_{\mathcal{M}} \|\nabla u\|^2 da \right\} \\ \text{subject to } & \int_{\mathcal{M}} \|u\|^2 da = 1, \end{aligned}$$

where ∇ denotes here the covariant derivative of vector fields. In the discrete setup, the above equation is expressed as an eigenvalue problem of the following sparse matrix:

$$E = \text{ad}_v^T \text{ad}_v + \lambda L,$$

where L is the vector Laplacian. We used the discrete connection Laplacian with a 1-ring neighborhood of faces given by Singer and Wu (2012), however, any other vector Laplacian can be used instead, e.g., the one suggested by Knöppel et al. (2013). Since we are interested in tangent vector fields, we further modify the matrix E which is of size $3m \times 3m$ by $E_B = B^T E B$, where $B \in \mathbb{R}^{3m \times 2m}$ encodes an arbitrary orthonormal basis per face. Figure 11 shows a collection of vector fields which are approximately commuting with a given one, computed using this approach.

4.5. Integrating the Lie bracket operator

The functional flow-map defines a discrete flow-map of a vector field v , by using the matrix exponential of D_v^V . Applying the functional flow-map allows us to transport *functions* on the flow of v . Similarly, we use the matrix exponential of ad_v , to transport *vector fields* on the flow of v . To that end, we define the following operator:

$$\text{Ad}_{\Phi_{v,t}} = \exp(t[I_F^V] \text{ad}_v) \in \mathbb{R}^{3m \times 3m}, \quad (67)$$

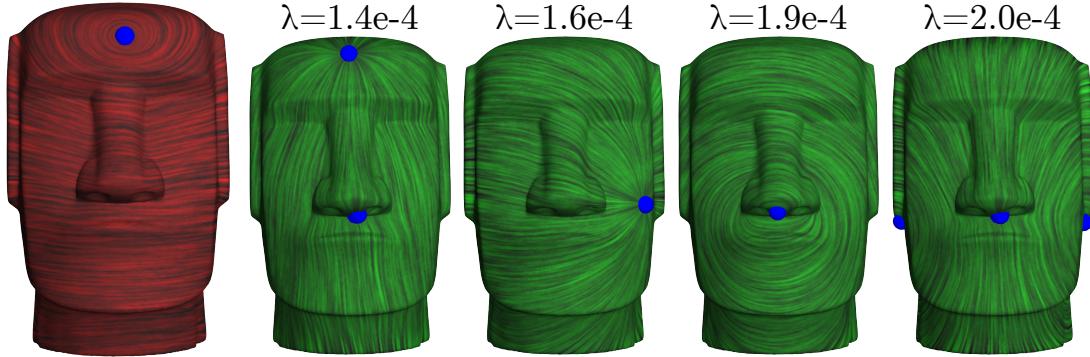


Figure 11: As-commuting-as-possible vector fields, with respect to the vector field on the left, the blue points are the singularities.

where $[.]$, replicates the input matrix 3 times along the diagonal.

The discrete approximation given in Eq. (67) is potentially useful in the context of fluid simulation. Indeed, while ideal fluids on surfaces can be simulated using function transport as was shown in Azencot et al. (2014b), the situation for ideal fluids in volume is more involved as it includes vectorial transport. For instance, Elcott et al. (2007) simulated fluids on tetrahedral meshes by advecting the vorticity field over the flow lines of the velocity field. Since our operators can be easily adapted to tetrahedral meshes, an operator related to $\text{Ad}_{\Phi_{v,t}}$ can be similarly used to integrate the volumetric Euler equations. This is an interesting direction for future work.

4.5.1. Application: Interpolation of vector fields

The flow transportation machinery allows us to interpolate between two vector fields in a more intuitive way than pointwise interpolation. Specifically, consider the vector fields u_0 and u_1 shown in Figure 12. They both have two singularities: one visible on the back, and another one on the left leg. If we simply interpolate the vector fields pointwise, we would get two singularities on the back (Figure 12 (left)).

However, if there exists another vector field v , with a corresponding flow $\Phi_{v,t}$, such that the singularity of u_0 is mapped to the singularity of u_1 under the map $\Phi_{v,1}$, then we can transport both vector fields on the flow of v and combine the result to get the interpolated vector field u_t . Specifically, for $t \in [0, 1]$ we use:

$$u_t = (1 - t) \text{Ad}_{\Phi_{v,t}} u_0 + t \text{Ad}_{\Phi_{v,t-1}} u_1. \quad (68)$$

A few frames from the result are shown in Figure 12. Note, that for $t = 0$ and $t = 1$ we get u_0 and u_1 , respectively, as expected, and for intermediate t values we get a smooth interpolation of both the location of the singularity and the behavior of the vector field.

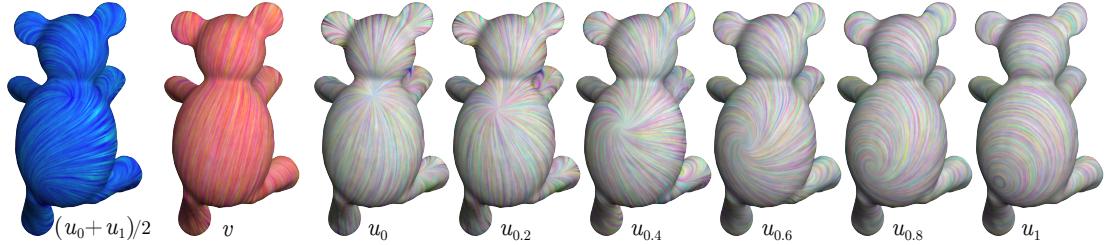


Figure 12: Interpolating between the vector fields u_0 and u_1 by transporting both on the flow of v . See the text for details.

It is worth noting that given two arbitrary vector fields, it is in general a difficult problem to find a corresponding vector field v which will transport u_0 to u_1 in a meaningful way. However, we believe that such a problem could perhaps be treated using optimal transportation machinery, as has previously been done for function flow in Solomon et al. (2014). Hence, our vector field transport could potentially be a key ingredient in extending such methods to vector fields.

4.6. Outlook

The mixed operators we have defined could potentially find multiple applications in geometry processing. Commuting flows can be used for local parameterization, and vector field transport could be used for numerically simulating partial differential equations, or as a building block in more complex setups for computing optimal transportation flows. Furthermore, the ad operator can be used to discretize the directional derivative of the matrix exponential, potentially allowing to compute a time-varying vector field whose flow corresponds to a given map.

5. Conclusion and Future Work

The algebraic approach presented in this chapter, in which tangent vector fields are represented globally instead of locally, as linear operators instead of pointwise directions, has proven to be beneficial in many applications, and provides ample opportunity for future work.

From the theoretical perspective, the most interesting open question is whether the space of tangent vector fields on a triangle mesh, and the corresponding flows, can be discretized such that the properties of the smooth case carry over to the discrete setting. Specifically, in the smooth case, the flows and tangent vector fields form a Lie group and a Lie algebra, respectively. In the discrete case, it would be most natural to describe these using a matrix Lie group and algebra. The answer to this question hinges on finding a definition of directional derivative operators, whose commutator is again a tangent vector field, as in the smooth

case. The mixed Lie bracket that we have presented in Definition 10 has some of the required properties, e.g. the 1-ring sparsity structure, but not all. Finding such a definition, or proving it cannot exist, remains an open question, to the best of our knowledge.

From the practical perspective, the operators described in this chapter are ripe for use in geometry processing applications. The most obvious use case is parameterization using commuting tangent vector fields. Other potential applications include vector field tracing, vector field visualization, and numerical simulation of PDEs.

References

- Absil, P.A., Mahony, R., Sepulchre, R., 2009. Optimization algorithms on matrix manifolds. Princeton University Press.
- Al-Mohy, A.H., Higham, N.J., 2011. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM J. Scientific Computing* 33, 488–511.
- Arnold, V.I., 1989. Mathematical methods of classical mechanics. volume 60. Springer Science & Business Media.
- Azencot, O., Ben-Chen, M., Chazal, F., Ovsjanikov, M., 2013. An operator approach to tangent vector field processing. *Comput. Graph. Forum* 32.
- Azencot, O., Corman, E., Ben-Chen, M., Ovsjanikov, M., 2017. Consistent functional cross field design for mesh quadrangulation. *ACM Trans. Graph.* 36, 92:1–92:13.
- Azencot, O., Ovsjanikov, M., Chazal, F., Ben-Chen, M., 2015a. Discrete derivatives of vector fields on surfaces – an operator approach. *ACM Trans. Graph.* 34.
- Azencot, O., Vantzos, O., Ben-Chen, M., 2016. Advection-Based Function Matching on Surfaces. *Computer Graphics Forum* .
- Azencot, O., Vantzos, O., Ben-Chen, M., 2018. An explicit structure-preserving numerical scheme for epdiff, in: *Computer Graphics Forum*, Wiley Online Library. pp. 107–119.
- Azencot, O., Vantzos, O., Wardetzky, M., Rumpf, M., Ben-Chen, M., 2015b. Functional thin films on surfaces, in: *Proc. Symp. Computer Animation*.
- Azencot, O., Weißmann, S., Ovsjanikov, M., Wardetzky, M., Ben-Chen, M., 2014a. Functional fluids on surfaces. *Computer Graphics Forum* 33.

- Azencot, O., Weißmann, S., Ovsjanikov, M., Wardetzky, M., Ben-Chen, M., 2014b. Functional fluids on surfaces. *Computer Graphics Forum* 33, 237–246.
- Bommes, D., Zimmer, H., Kobbelt, L., 2009. Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG)* 28, 77.
- Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., Lévy, B., 2010. *Polygon mesh processing*. CRC press.
- Crane, K., Desbrun, M., Schröder, P., 2010. Trivial connections on discrete surfaces. *Computer Graphics Forum (SGP)* 29.
- Desbrun, M., Hirani, A., Leok, M., Marsden, J., 2005. Discrete exterior calculus. Preprint, arXiv:math.DG/0508341.
- Elcott, S., Tong, Y., Kanso, E., Schröder, P., Desbrun, M., 2007. Stable, circulation-preserving, simplicial fluids. *ACM Transactions on Graphics (TOG)* 26, 4.
- Godunov, S.K., 1997. Ordinary differential equations with constant coefficient. volume 169. American Mathematical Soc.
- Hochbruck, M., Ostermann, A., 2010. Exponential integrators. *Acta Numerica* 19, 209–286.
- Knöppel, F., Crane, K., Pinkall, U., Schröder, P., 2013. Globally optimal direction fields. *ACM Trans. Graph.* 32.
- Knöppel, F., Crane, K., Pinkall, U., Schröder, P., 2013. Globally optimal direction fields. *ACM Transactions on Graphics (TOG)* 32, 59.
- Lee, J., 2012. *Introduction to Smooth Manifolds* (Graduate Texts in Mathematics, Vol. 218). Springer.
- Moler, C., Van Loan, C., 2003. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review* 45, 3–49.
- Morita, S., 2001. *Geometry of differential forms, translations of mathematical monographs*, vol. 201. American Mathematical Society, Providence, RI .
- Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L., 2012. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 30.
- Petersen, P., Axler, S., Ribet, K., 2006. *Riemannian geometry*. volume 171. Springer.

- Polthier, K., Preuss, E., 2003. Identifying vector field singularities using a discrete Hodge decomposition, in: Visualization and Mathematics III. Springer, pp. 113–134.
- Ray, N., Sokolov, D., 2014. Robust polylines tracing for n-symmetry direction field on triangulated surfaces. ACM Transactions on Graphics (TOG) 33, 30.
- Singer, A., Wu, H.T., 2012. Vector diffusion maps and the connection Laplacian. Communications on pure and applied mathematics 65.
- Solomon, J., Rustamov, R., Guibas, L., Butscher, A., 2014. Earth mover’s distances on discrete surfaces. ACM Transactions on Graphics (TOG) 33, 67.
- Zhang, E., Mischaikow, K., Turk, G., 2006. Vector field design on surfaces. ACM Trans. Graphic. 25.