Objective Observer-Relative Flow Visualization in Curved Spaces for Unsteady 2D Geophysical Flows

Peter Rautek, Matej Mlejnek, Johanna Beyer, Jakob Troidl, Hanspeter Pfister, Thomas Theußl, and Markus Hadwiger

1 INTRODUCTION

Flow visualization is concerned with velocity vector fields that describe fluid motion. Since the definition of velocity is the infinitesimal change of spatial position with respect to time, this description depends on the notion of what is considered to be “the same position” over time. Mathematically, this is the notion of a frame of reference, or an observer. A frame of reference is a way of referencing positions in space over time, i.e., a time-dependent transformation of space, with respect to which derivatives are to be taken. (Not an observer as an actual person.)

For this reason, flow visualization as well as the computation of flow features, such as vortices, in general inherently depend on the chosen frame of reference. An important related concept therefore is invariance with respect to certain changes of reference frame. Galilean invariance [28] refers to frame changes with constant relative velocity, which is inherently related to inertial frames in physics [63]. Objectivity [31] considers frame changes with time-dependent translations and rotations. The standard definition used in visualization and continuum mechanics is the one of Truesdell and Noll [65], which has received significant attention for flow fields since the work of Haller [31]. A major basic limitation of this definition, however, is that it is given only for Euclidean space and cannot be used in general curved spaces.

One area where objectivity is particularly relevant is that of geophysical flows, such as in the Earth’s atmosphere or oceans. The natural reference frame, in this case, is rotating [51, p.14], i.e., non-inertial, and Galilean invariance is not well-suited. As is common in fluid visualization, we only consider motions by themselves (kinematics), not...
We develop a novel observer field framework where a geophysical flow field is defined from an intrinsic, curved 2D perspective. Our major contribution is the generalization of a prior Euclidean framework [30] showing four different minimizing properties of manifolds and vector fields. These are particularly important for computations with vector fields on surfaces, such as vector field design on surfaces. Image-based texture advection techniques [37, 67] have been extended as well, such as image-based flow visualization on surfaces [39, 68]. Image space techniques typically project vector fields from 3D to 2D screen space for visualization. We refer to the survey of Edmunds et al. [20]. Texture atlases have also been used for flow visualization, for example flow charts [44]. We similarly use multiple coordinate charts.

### 1.1 Observer Fields and Observed Time Derivatives

An observer field $\mathbf{u}(x,t)$ is a time-dependent velocity field that, instead of describing the motion of particles, describes the motion of a continuous field of observers [30]. The *path lines* of the observer field correspond to the *world lines* [3, p.8] of individual spatial locations in the space of an observer that are viewed as being the “same point” over time. If an observer field $\mathbf{u}$ describes a rigid motion, it is a *Killing vector field* [53]. Figs. 2 (a,b,c) depict three time steps of observer fields on a sphere. Fig. 3 shows one example input vector field $\mathbf{v}$ describing a simple rotating vortex, as it is seen by four different observer fields. The *observed time derivative* corresponding to a given $\mathbf{u}(x,t)$ measures the differential change of an arbitrary input field $\mathbf{v}(x,t)$ with respect to the observer motion described by $\mathbf{u}$. This concept was introduced in the context of Euclidean space [30]. We define this time derivative, on any differential manifold $M$, as the differential operator

$$\mathcal{D} = \frac{\partial}{\partial t} + \mathcal{L}_\mathbf{u}.$$  

Here, $\mathcal{L}_\mathbf{u}$ denotes the Lie derivative [22, Ch. 4] with respect to the observer field $\mathbf{u}$. In order to “steadify” an input vector field $\mathbf{v}$, e.g., to be able to compute and follow features, we compute observer fields $\mathbf{u}$ by minimizing the derivative $(\mathcal{D}/\partial t)(\mathbf{v} - \mathbf{u})$. If Fig. 3 is interpreted as showing four different input fields $\mathbf{v}$, the observer-relative visualization with respect to the optimized field $\mathbf{u}$ will always look like Fig. 3 (d).

### 1.2 Mathematical Framework for Curved Spaces

We develop a novel fully intrinsic mathematical framework for 2D flow visualization and feature computation in curved spaces. Our major contribution is the generalization of a prior Euclidean framework [30] for the objective computation of observer fields to curved spaces. We compute these fields by solving an energy minimization problem, making unsteady input flows as steady as possible. We first introduce a generalized notion of objectivity for curved spaces. While equivalent to the standard one in Euclidean space [65], ours is fully intrinsic, using the concepts of continuous symmetry groups and group actions on manifolds from mathematical physics. From general symmetry groups, we specialize to observer transformations modeled as elements of the *isometry group* of a manifold $M$. To simplify computations, we work with infinitesimal isometries, given explicitly by vector fields on $M$.

To be able to achieve this, we derive the differential geometric operators needed to compute observer fields in curved spaces: (1) Observed time derivatives; (2) Approximate Killing fields; and (3) The velocity gradient tensor $\mathbf{V}$ as a *covariant derivative* [22, Ch. 9], instead of the Jacobian that is commonly used in flow visualization.

**Methodology.** Our generalization of objectivity and observer fields to curved spaces requires mathematical machinery from Riemannian geometry and mathematical physics that is not common in the flow visualization literature. For this reason, we summarize background and our notation in the beginning, and provide extensive supplementary appendices. These techniques enable deriving a powerful generalization of previous flow visualization techniques from flat space to curved spaces in a clean and intrinsically manner. We believe that this powerful methodology can be very useful for flow visualization in general, and that it is worthwhile to introduce it to the flow visualization literature.

### 2 Related Work

The framework presented here extends the work of Hadwiger et al. [30] from Euclidean space $\mathbb{R}^n$ to curved spaces, focusing on curved two-manifolds. We also model a collection of observers by an *observer velocity field* describing their motion, which we also compute by solving a least-squares problem such that (1) it is objective, and (2) the input flow field is as steady as possible relative to the observer field. However, the generalization to curved spaces is conceptually and technically challenging. We propose a fully intrinsic generalization, working “inside” the curved space. This results in natural and efficient generalizations, and facilitates computing 2D observer fields via 2D optimization.

**Flow visualization on curved surfaces.** All major flow visualization techniques have been extended from flat space to curved surfaces. For example, LIC [14] and variants [43, 55, 60] have been extended to curved surfaces, either for parametric surfaces [21], by texturing triangle meshes [8, 59, 62], or by computing LIC in screen space [6, 36]. Image-based texture advection techniques [37, 67] have been extended as well, such as image-based flow visualization on surfaces [39, 68]. Image space techniques typically project vector fields from 3D to 2D screen space for visualization. We refer to the survey by Edmunds et al. [20]. Texture atlases have also been used for flow visualization, for example flow charts [44]. We similarly use multiple coordinate charts.

**Vector field design on surfaces.** We extensively use differential properties of manifolds and vector fields. These are particularly important for computations with vector fields on surfaces, such as vector field design on surfaces [15, 16, 71]. We refer to the survey of Peng and Lameere [52]. We would like to particularly point out the use of geodesic polar maps and parallel transport along geodesics in the work.
of Palacios and Zhang [50]. The notion of covariant derivative, which we employ in our work, is inherently related to parallel transport.

**Covariant derivatives** have been used for mesh processing [18] and computing vector [17] and direction [69] fields on surfaces. An important focus is often their discretization on triangle meshes [5], e.g., by using discrete exterior calculus [18]. We discretize on triangle vertices in 2D charts, and particularly focus on fully intrinsic 2D computations.

**Interpolation.** We use standard barycentric interpolation in triangles. However, spherical barycentric coordinates, i.e., mean value coordinates on the sphere, could also be used for better interpolation [38].

**Killing vector fields** on (Riemannian) manifolds [19] correspond to their infinitesimal isometries. Their flows correspond to the **intrinsic** isometries of a manifold. Basics of Killing fields for curved surfaces are discussed by Ben-Chen et al. [9] and by Solomon [57]; a more advanced treatment is given by Petersen [53]. Approximate Killing fields have been used to compute approximate intrinsic isometries of curved surfaces [9], or to design approximate Killing fields on meshes [4, 5]. As-rigid-as-possible shape interpolation methods [11] work directly with isometries instead of with derivatives, and are computed extrinsically.

**Observers.** The concept of an observer as well as that of objectivity in Euclidean space has been recognized to be of importance in fluid mechanics [31] as well as in flow visualization [28]. A standard reference in continuum mechanics is Truesdell and Noll [65], whose definition for objectivity in Euclidean space is used often [7, 25, 31, 32, 34, 48].

**Vortex detection and frame invariance.** Vortices are an important topic in fluid mechanics [54] and in flow visualization [28]. Many vortex detection methods were originally defined for steady flow, such as Sujudi and Haines [61], and were later extended to unsteady flow [70]. Galilean invariance of these criteria is considered to be important [28]. Several known methods are Galilean-invariant, such as Weinkauf et al. [70], or the method of Bujack et al. [13]. The latter jointly considers multiple observers for 2D flow fields. Specific vortex detectors have been developed to be objective by design [31, 32], Günther et al. [7, 25] present objective methods for Euclidean space that are generic by building on optimization, making non-objective vortex detectors objective. Other kinds of invariance have been defined, particularly for flow fields, such as rotation invariance [26], or hyper-objective vortexes [27].

**Flow decomposition** methods, such as variants of the Helmholtz-Hodge decomposition [10, 11], can also be used to remove background flow, corresponding to the harmonic component [12]. These decompositions can be computed on curved surfaces [64]. However, they apply only to time-independent fields, or alternatively they have to be applied to each time step individually, without considering time derivatives.

### 3 Background and Notation

Our intrinsic framework enables all computations, most importantly the optimization of observer fields, to be 2D for curved surfaces, instead of requiring 3D ambient space computations. We perform all computations in an atlas of 2D coordinate charts (Fig. 4). This also simplifies visualization computations on curved surfaces, by working directly in charts, for example for path line integration or Line Integral Convolution (LIC), as depicted in Fig. 3. However, already a surface as simple as a sphere cannot be covered by a single chart. All computations in charts therefore need to be independent of the choice of coordinates. We ensure this invariance by using general tensor methods [24], with the corresponding transformation rules for transitions between charts.
We note that even in flat $\mathbb{R}^2$ with polar coordinates, $\partial \rho \nu^i$ is not a tensor. This has important implications, such as that a tensor that vanishes in one coordinate system must vanish in all coordinate systems [24, p.82]. This is not true for $\partial \nu^i$, and thus the Jacobian behaves only as it should in special cases, such as by restriction to Cartesian tensors [2, Ch.2].

Our framework requires a more general definition of $\nabla \nu$ as the covariant derivative $\nabla \nu^i$, which is standard in mathematical physics. However, we have found no explicit use of this concept in the flow-visualization literature. We derive this notion for our purposes in Sec. 6.

**Approximate Observer Killing fields.** In general, an observer field is allowed to be given by an arbitrary vector field $u$. However, it makes sense to restrict the motion described by $u$ to be “as rigid as possible.” We therefore optimize observer fields such that they correspond to approximately isometric deformations, i.e., to “almost-rigid” motions. In the same vein as Hadwiger et al. [30], we minimize the rate-of-deformation tensor of $u$. However, in curved spaces $M$ this computation requires using the covariant derivative $\nabla u$, and the metric $g$ on $M$.

### 4 Observers in Curved Spaces

The first crucial notion for us to define is the meaning of an observer in a curved space. In analogy with the standard notion of observer transformations in Euclidean space [65], it is natural to consider observer transformations in general be time-dependent distance-preserving transformations of the underlying space, i.e., time-dependent isometries. In the general case, however, we have to consider the intrinsic isometries of a smooth manifold $M$ with metric $g$, i.e., corresponding to geodesic distances instead of Euclidean distances. Generally stated,

**Remark.** We employ a (non-relativistic) concept of frames of reference postulating that all observers agree on the pairwise spatial geodesic distances of events happening at the same time, and all observers also agree on the time difference between events (i.e., time is absolute).

Observer transformations are therefore time-dependent intrinsic distance-preserving transformations. Mathematically, we model these observer transformations as paths $t \mapsto g(t) \in G$, through a Lie group $G$, chosen as the isometry group $G := \text{Isom}(M)$ of the manifold $M$ with metric $g$. In our framework, we often also work directly with the derivatives of $g(t)$, forming the corresponding path $t \mapsto X(t) \in g$ through the Lie algebra $\mathfrak{g} = \text{isom}(M)$, the Lie algebra of infinitesimal isometries.

#### 4.1 Observers on the Sphere

Figs. 2 and 5 illustrate this idea for $M = S^2$, the standard two-sphere. The motion of the Earth relative to a fixed observer, such as the airplane in Fig. 2, is given by a path of isometries $t \mapsto g(t) \in SO(3)$, which here are different rotations of the Earth. At any instant in time $t$, the derivative of this path corresponds to a Lie algebra element $X(t) \in \mathfrak{so}(3)$, which defines an infinitesimal isometry of $M$. However, $X(t)$ is not necessarily an anti-symmetric matrix (Fig. 5, bottom). The specific corresponding infinitesimal isometry of the sphere is given by an isomorphism between $\mathfrak{so}(3)$ and the Lie algebra of Killing vector fields on the sphere (Fig. 5, top). This isomorphism is mathematically described by a Lie algebra action (see App. J). In this way, the infinitesimal isometry on the surface of the Earth is the Killing vector field $X(t)$ corresponding to $X(t)$. At each point $x \in M$, this gives a relative velocity vector $x \cdot X(t)$ (Fig. 5, top). We also know that $x$ describes an infinitesimal isometry of $M$, because $\nabla x$, as the covariant derivative of $x$, is anti-symmetric. In fact, any Killing field $X$ is uniquely determined by its value $X(x)$ and covariant derivative $(\nabla x)_a$ at one point $x \in M$ [53, Proposition 8.1.4].

#### 4.2 Observer Fields

Generalizing further from the concept of one observer, as one time-dependent isometric transformation of space, we want to be able to describe an observer field—many observers—on a manifold $M$ as a vector field $u$ on $M$, where $u$ does not necessarily correspond to an exact (infinitesimal) isometry. One of the most important properties of $u$ is (Riemannian) metric $g$ is required to define the meaning of isometry.

2It is important to note, however, that checking $\nabla x$ for anti-symmetry needs to be done with care, because it is a tensor of mixed type. See Sec. 7.1.
To generalize objectivity, we define this concept as a general notion of vectors, is objective if it transforms as $\phi(x)^\ast = Q(t) \phi(x)$, a second-order tensor field $S$, as a linear transformation of vectors, is objective if it transforms as $S' = Q(t) S Q(t)^T$ [65, p.42].

This entire definition depends on the domain being Euclidean: points are position vectors; the difference between two points is a vector; all tangent spaces are copies of $\mathbb{R}^3$ with trivial parallel transport. This definition is therefore not valid for non-Euclidean (curved) manifolds.

5.2 Generalization of Objectivity

To generalize objectivity, we define this concept as a general notion of tensor fields being invariant with respect to a continuous symmetry group $G$, which is a Lie group. (Symmetry refers to a notion of being the same.) For example, if the group $G$ is chosen as the isometry group of a (Riemannian) manifold, two tensor fields are “the same” if they are isometric. Two fields being symmetries of each other then means that there exists a group element $g \in G$, such that the transformation rules given below hold. Then, given any observer transformation $r \mapsto g(r) \in G$, a given tensor field is objective if, for each fixed $r$, it simply follows the corresponding transformation $g := g(r)$.

5.2.1 Symmetry groups and group actions

Our notion of symmetry corresponds to the transformation behavior under a group action $\Phi$, with a given Lie group element $g \in G$, where $G$ is the chosen symmetry group. An action $\Phi$, specifically a smooth left action, of a Lie group $G$ on a manifold $M$, is a smooth map [33, p.209]

$$\Phi : G \times M \to M, \quad (g,x) \mapsto \Phi(g,x),$$

such that for every $g \in G$, the map

$$\phi_g : M \to M, \quad \phi_g(x) := \Phi(g,x),$$

is a diffeomorphism. (6)

Here, we focus on the general use of group actions $\Phi$ in our context, and defer details to later sections. For now, it is sufficient to understand that the diffeomorphisms $\phi_g$ will correspond to the flows of specific vector fields on $M$. These vector fields are generated by the action of the Lie algebra $g$ of the Lie group $G$ on $M$. See App. J for details.

For example, if $G$ is the group of all diffeomorphisms of $M$, these vector fields are all possible (smooth) vector fields on $M$. The important case for our framework is choosing the group $G$ as the isometry group of $M$. The corresponding vector fields are then the Killing vector fields on $M$, whose flows correspond to the isometries of $M$. See Sec. 7.

To obtain a generalized definition of objectivity, a crucial property of the diffeomorphism $\phi_g$ is that it enables us to use the corresponding differential, or pushforward. See Fig. 7. The pushforward is a map

$$d\phi_g : TM \to TM,$$

where each $(d\phi_g)_x$ at a point $x \in M$ is a linear map

$$(d\phi_g)_x : T_xM \to T_{\phi_g(x)}M,$$

$$v \mapsto (d\phi_g)_x(v).$$

The notation $(\cdot)_x$ means that the quantity in parentheses is located at $x \in M$, and $T_xM$ denotes the tangent space at $x$. We can simply imagine that the diffeomorphism $\phi_g$ transforms curves on $M$, and the differential $d\phi_g$ transforms their tangent vectors accordingly. See also App. U.

In components, the map $(d\phi_g)_x$ is differentiable for different points $x \in M$. In components, each $(d\phi_g)_x$ can still be a vector, but it will be a different matrix for each point $x \in M$.

5.2.2 Objective scalar fields

Being objective should mean invariant under transformation, which for scalar fields is trivial. We therefore define that a scalar field $f \in T^0_M$ on a manifold $M$ is objective when, under any diffeomorphism $\phi_g$, given by the group action $\Phi$ of a symmetry group $G$, it transforms as

$$f^\Phi = f \Phi(x) = f(x).$$

Abbreviated, we could write $f^\Phi = f$, but it is crucial to note that $f^\Phi$ is evaluated at the point $\phi_g(x)$, whereas $f$ is evaluated at the point $x$.

5.2.3 Objective vector fields

We now define that an arbitrary vector field $v \in T^1_M$ on a manifold $M$ is objective (with respect to a given symmetry group $G$), if, under the corresponding group action $\Phi$ with any $g \in G$, it transforms as

$$(v^\Phi)_x = (d\phi_g)_x(v).$$

We emphasize that $v^\Phi$ is an element of the tangent space $T_{\phi_g(x)}M$, whereas $v$ is an element of $T_xM$. Likewise, it is important to note that the differential $(d\phi_g)_x$ is a linear map defined on $T_xM$. We can say

Remark. A vector field is objective, if it is simply pushed forward by any diffeomorphism $\phi_g$, defined according to the group action $\Phi$. This definition of objectivity is valid for any smooth manifold where a notion of (smooth) symmetry is defined by a (smooth) symmetry group $G$.

Abbreviated transformation rule. For brevity, we can define the action $\Phi$, with $g \in G$, on any vector field $v$ on $M$, by the differential in Eq. 8, and abbreviate the objectivity criterion of Eq. 10 simply as

$$v^\Phi = g \cdot v.$$  

However, it is crucial that the meaning of the transformation represented by $g \in G$ in this shorthand notation is given by Eq. 10. In general, $g$ cannot be mapped to the same globally defined matrix, corresponding to the pushforward $d\phi_g$, even though this is possible in the Euclidean case. Nevertheless, this abbreviated form makes it easy to see the analogy with the definition of Truesdell and Noll. In Euclidean space, the two are equivalent. See App. D for more details. Our definition, however, gives a well-defined notion of objectivity for arbitrary manifolds $M$.
5.2.4 Objective tensor fields

The definition above for objective vectors generalizes to arbitrary tensor fields \( \mathbf{T} \), in the way that is derived in full in App. A. Here, we derive the common special case of a second-order tensor field \( \mathbf{S} \), interpreted as a linear map of vectors, i.e., \( \mathbf{S} : TM \to TM, v \mapsto \mathbf{S}(v) \). Such a \( (1,1) \) tensor field is objective, if under the group action \( \Phi \) with \( g \) it transforms as

\[
(S^*)_{\Phi(x)}(v^*) = (d\Phi_g)_x(S((d\Phi_g^{-1})_{\Phi(x)}(v^*))).
\] (12)

The pushforward \( d\phi_g^{-1} \) of the diffeomorphism \( \phi_g \) maps \( v \) from \( T_{\phi(x)}(M) \) to the corresponding \( v \) in \( T_x(M) \) according to Eq. 10 (inverted), and then applies Eq. 10 again to \( \mathbf{S}(v) \). \( \phi_g^{-1} \) is the inverse of \( \phi_g \), meaning \( \phi_{g^{-1}}(\phi_g(x)) = \phi_g^{-1}(x) = x \). As in Eq. 11, we can abbreviate

\[
S^* = gSg^{-1}.
\] (13)

However, it is again crucial to note that the meaning of this rule is given by Eq. 12. Fig. 7 illustrates how the pushforwards of \( \phi_g \) and \( \phi_g^{-1} \) map tangent vectors between the tangent spaces \( T_{\phi(x)}(M) \) and \( T_x(M) \).

We note that to prove objectivity of the Killing operator \( K \) (Eq. 25), which we use for computing observer fields, the general definition of objectivity from App. A is required, since \( K \mathbf{u} \) is a \((0,2)\) tensor (App. F.1).

5.3 Objectivity for Intrinsic Isometries

To realize the notion of observer transformations as isometric transformations of the underlying space in the general mathematical framework given above, we now consider the group \( G := \text{Isom}(M) \), i.e., the isometry group of the manifold \( M \). The group action \( \Phi \) is an isometry, with respect to a given metric \( g \), i.e., a smoothly varying inner product \( \langle \cdot, \cdot \rangle \), on \( M \), if for all \( g \in G \), and for any two vector fields \( v, w \) on \( M \), we have

\[
\langle (d\phi_g)_x(v), (d\phi_g)_x(w) \rangle_{\phi_g(x)} = \langle v, w \rangle_x.
\] (14)

As a direct consequence of Eq. 14 holding for the map \( \phi_g \), we get that for any vector field \( v \) that is objective according to Eq. 10, we have

\[
\|v^*\|_{\phi_g(x)} = \|v\|_x
\] (15)

Infinitesimal isometries and Killing vector fields. Computationally, instead of working with isometries directly, we work with their derivatives, which are infinitesimal isometries, corresponding to the Killing fields on \( M \). See Sec. 7. To describe Killing fields efficiently, we need the intrinsic velocity gradient tensor \( \nabla \mathbf{v} \) of a velocity field \( \mathbf{v} \) on \( M \).

6 General Velocity Gradient Tensors

For any smooth vector field \( \mathbf{v} \) given on a manifold \( M \), the velocity gradient tensor \( \nabla \mathbf{v} \) is a second-order tensor field on \( M \) that allows computing differential derivatives of the vector field \( \mathbf{v} \) in any direction.

In Cartesian coordinates, \( \nabla \mathbf{v} \) can simply be defined by partial derivatives \( \partial_i v^j \). However, this definition cannot be used for intrinsic computations on curved manifolds, because \( \partial_i v^j \) is in general not a tensor. For a curved (or flat) manifold \( M \), the tensor \( \nabla \mathbf{v} \) can, however, always be computed intrinsically as the covariant derivative of \( \mathbf{v} \) [22, p.241].

6.1 Intrinsic Covariant Derivatives

The covariant derivative \( \nabla \mathbf{v} \) of a vector field \( \mathbf{v} \) on a manifold \( M \) is a second-order tensor field on \( M \), which, at any \( x \in M \), maps an arbitrary vector \( \mathbf{x} \in T_x(M) \) to the derivative of the field \( \mathbf{v} \) in the direction \( \mathbf{x} \), i.e.,

\[
(\nabla \mathbf{v})_x : T_x(M) \to T_x(M),
\]

\[
\mathbf{x} \mapsto \nabla_x \mathbf{v} = (\nabla \mathbf{v})_x(\mathbf{x}).
\] (16)

This notation means that the covariant derivative of a vector field \( \mathbf{v} \), evaluated in the direction \( \mathbf{x} \), is the vector \( \nabla_x \mathbf{v} \), which we get by computing \( \nabla \mathbf{v} \)(\( x \)). We can refer \( \nabla \mathbf{v} \) to a basis \( \{\mathbf{e}_i\} \), with \( \nabla \mathbf{v} \) the product, and write the components as \( n \times m \) matrix \( \nabla v^i_j \), with row and column indices \( i, j \), respectively, and the dimension of \( M \); i.e., for a curved surface \( n = 2 \). However, unless the manifold \( M \) is flat and Cartesian or affine coordinates are used, \( \nabla \mathbf{v} \neq \partial \mathbf{v} \). The difference is a connection on \( M \) [22, p.242], corresponding to a notion of parallel transport [19, p.52] of vectors. The connection can be given in components via Christoffel symbols \( \Gamma^i_{jk} \), and we get for \( \nabla \mathbf{v} \) in components,

\[
\nabla v^i_j := \partial_j v^i + \Gamma^i_{jk} v^k.
\] (17)

In flat space with Cartesian or affine coordinates, all Christoffel symbols vanish [24, p.72] \( (\Gamma^i_{jk} \equiv 0) \), but only in this special case is \( \nabla \mathbf{v} = \partial \mathbf{v} \).

The \( \Gamma_{ijk} \) determine the covariant derivatives of the basis vector fields \( \{\mathbf{e}_i\} \) at each point \( x \in M \), referred back to the same basis \( \{\mathbf{e}_i\} \) at \( x \), i.e.,

\[
\nabla_x \mathbf{e}_i = \Gamma^i_{jk} \mathbf{e}_j, \quad \text{and therefore} \quad \Gamma^i_{jk} = \omega^i \langle \nabla_x \mathbf{e}_j \rangle.
\] (18)

We note that although \( \Gamma^i_{jk} \) comprises \( n^3 \) components, it is not a (third-order) tensor [24, p.82]. In general, a connection needs to be chosen [19, p.50]. We employ the Levi-Civita connection, which can be derived intrinsically from the metric of \( M \) [22, p.229]. See App. O, Eq. O.10. Even more easily, if we know an immersion of \( M \) into an ambient space \( \mathbb{R}^m \), as for a curved surface immersed in \( \mathbb{R}^3 \), we can obtain the \( \Gamma_{ijk} \) as

\[
\Gamma^i_{jk} = \tilde{\omega}^i (\partial_j \tilde{\mathbf{e}}_k).
\] (19)

The tilde symbols denote ambient space (co)vectors, e.g., for \( M \) with \( n = 2 \) and ambient \( m = 3 \), \( \{\partial_i \tilde{\mathbf{e}}_j\} \) comprises four 3D vectors, and \( \{\tilde{\omega}^i\} \) comprises two 3D covectors. It is crucial that the \( \{\tilde{\mathbf{e}}_j\} \) are chosen such that they correspond to orthogonal projection from \( \mathbb{R}^3 \) onto the tangent plane of the surface. See App. Q for a complete calculation for a simple atlas of charts for the sphere. The Levi-Civita connection given by this choice of \( \Gamma^i_{jk} \) corresponds to the metric on \( M \) that is induced by the immersion of \( M \) into the ambient space \( \mathbb{R}^m \) [19, p.39]. Crucially, however, the resulting Christoffel symbols are nevertheless intrinsic.

7 Isometries and Killing Fields in Curved Spaces

Our framework requires an efficient mathematical description of the intrinsic isometries of the manifold \( M \). A powerful way to do this is to work with the derivatives of diffeomorphisms that are isometries, i.e., to consider infinitesimal isometries. These correspond to specific vector fields on the manifold \( M \), which are called Killing vector fields. A vector field \( \mathbf{u} \) is a Killing vector field on \( M \), if the flow \( \phi_t \) (App. T) that it generates preserves the metric \( g \) on \( M \). The most general way of stating this mathematically is to require that the Lie derivative (App. G) of the metric \( g \), with respect to \( \mathbf{u} \), must vanish everywhere on \( M \). I.e.,

\[
\mathcal{L}_\mathbf{u} g = 0.
\] (20)

This means that for all \( x \in M \), we have \( \mathcal{L}_\mathbf{u} g \big|_x = 0 \). See [46, p.250].

The set of all possible Killing vector fields on a given (Riemannian) manifold \( M \) constitutes a Lie algebra, which is, in particular, a vector space. This enables us to consider the dimensionality of the set of

\[This is the unique metric-compatible connection, corresponding to a given (Riemannian) metric \( g \) on the manifold \( M \), that has zero torsion [22, p.242].\]
Applying the product rule for covariant derivatives [24, p.119], we therefore, Eq. 21 is given using the covariant derivative of a vector field.

Killing’s equation holds at all \( x \in M \) [22, p.529]. However, in its general form this equation is given for covectors, with components \( u_i \), and not for vectors, with components \( u^i \). The \( u_i \) can be computed as \( u_i = g_{ij} u^j \) [24, p.88], using a metric \( g \), referred to components \( g_{ij} \).

Killing’s equation is then given as

\[
\nabla_j u_i + \nabla_i u_j = 0. \tag{21}
\]

This is the same as \( \nabla u \) being anti-symmetric, which is the same as \( \langle u \rangle \) is curved, including the term in Eq. 30.

We get \( \mathcal{L}_u v = \frac{1}{2} \langle [v,u] \rangle \), which determines the Frobenius norm \( \text{tr} (g^{-1} T^T g^{-1} S) \). The crucial difference here is that because pairs of covariant and contravariant indices are contracted, Eq. 27 is an invariant tensor expression, giving the same result in all coordinate systems. \( \langle T, T \rangle_g = T^{ij} T_{ij} \), thus is the square of the norm. In matrix notation, we can write the inner product above as

\[
\langle T, S \rangle_g = \text{tr} (g^{-1} T^T g^{-1} S). \tag{28}
\]

We define the Killing energy \( E_u \), of the field \( u \), at a point \( x \in M \), as

\[
E_u := \langle K u \rangle_g^2 = \langle K u, K u \rangle_g. \tag{29}
\]

Killing’s equation is then given as

\[
\mathcal{L}_u v = \nabla v (u) - \nabla u (v). \tag{30}
\]

\[
\mathcal{L}_u = \frac{1}{2} \langle [v,u] \rangle \]

This is similar to the Frobenius inner product \( \text{tr} (T^T S) \) of matrices, which determines the Frobenius norm \( \text{tr} (T^T T) \). The crucial difference is that because pairs of covariant and contravariant indices are contracted, Eq. 27 is an invariant tensor expression, giving the same result in all coordinate systems. \( \langle T, T \rangle_g = T^{ij} T_{ij} \), thus is the square of the norm. In matrix notation, we can write the inner product above as

\[
\langle T, S \rangle_g = \text{tr} (g^{-1} T^T g^{-1} S). \tag{28}
\]

We define the Killing energy \( E_u \), of the field \( u \), at a point \( x \in M \), as

\[
E_u := \langle K u \rangle_g^2 = \langle K u, K u \rangle_g. \tag{29}
\]

The tensor \( K u \) is still a \( \frac{1}{2} \) tensor, just like \( \langle \nabla u_i + \nabla u_j \rangle \). However, Eq. 25 implies that now we can simply check any matrix representation of \( g \nabla u \) for anti-symmetry, to determine whether \( u \) is a Killing field.

7.2 Killing Energy

If the expression \( K u \) is not exactly zero, in order to quantify how close to being Killing an arbitrary vector field \( u \) is, we use the tensor norm of the tensor \( K u \) to define a scalar Killing energy (see also [9, Def. 5]). As for tensors, the tensor norm of a higher-order tensor \( T \) is defined as

\[
\|T\|_g^2 := \langle T, T \rangle_g. \tag{26}
\]

Killing fields on \( M \), and to write any Killing field as a linear combination of basis Killing fields. Fig. 5 shows an example on the sphere. Apps. L and M describe the isometries of the two-sphere in detail, with full calculations. App. K gives more details on Killing vector fields on curved surfaces in general. However, for computing observer fields via optimization (Sec. 9), as the most important part of our framework, we directly evaluate Killing’s equation on the unknown vector field \( u \).

8 Observed Time Derivatives in Curved Spaces

The observed time derivative defined in Eq. 1 measures the differential change of a tensor field with respect to the motion of a given observer field \( u \). We note that \( \partial / \partial t \) is, in fact, the time-dependent Lie derivative [45, p.95] with respect to the flow of the field \( u \). The crucial difference here is the semantic meaning of an observer field \( u \), as an intrinsic part of our operator. Our most important use of \( \partial / \partial t \) is applying it to the relative velocity field \( (v - u) \), which we minimize to compute the unknown observer field \( u \) (Sec. 9). We get \( \mathcal{L}_u u = 0 \),

\[
\frac{\partial}{\partial t} (v - u) = \frac{\partial v}{\partial t} - \frac{\partial u}{\partial t} + \mathcal{L}_u v. \tag{30}
\]

In matrix notation, we can therefore rewrite Killing’s equation as the vanishing of the differential “Killing operator” \( K \) applied to the field \( u \),

\[
K u := g \nabla u + (g \nabla u)^T = 0. \tag{25}
\]

\[\mathcal{L}_u = \frac{1}{2} \langle [v,u] \rangle \]

The expansion in components in the second row above comprises only partial derivatives. The full derivation is given in App. H. Thus, the observed time derivative can be derived solely from partial derivatives, even when the manifold \( M \) is curved, including the \( \mathcal{L}_u \) term in Eq. 30.

9 Computing Observer Fields

We compute a global observer field \( u \) on any two-manifold \( M \) with multiple charts using one global linear least-squares solve. The use of a single global sparse linear system is enabled by “baking” all transition maps (their Jacobian matrices) between charts into the system matrix.

9.1 Objective Function

Our optimization is conceptually very similar to the approach of Hadwiger et al. [30] for Euclidean space. However, all differential operators have to be substituted by their generalized version for arbitrary manifolds, i.e., covariant derivatives, Lie derivatives, and tensor norms.
We formulate a global optimization problem to find an observer field \( u \), from the space of all possible fields in some function space \( Y \) of vector fields (e.g., \( C^0 \) fields) on the manifold \( M \), as the minimizer of

\[
\min_{u \in \mathcal{Y}} \frac{1}{2} \int_M \left( E + \lambda D + \mu R \right) (u, x) \, dA(x)
\]

(32)

Here, \( dA \) is an infinitesimal area element of \( M \) at a point \( x \in M \), and the constants \( \lambda, \mu \in \mathbb{R} \) are relative weights. The individual terms are

\[
E(u, x, t) := \left\| Ku(x, t) \right\|^2_{g^u} \quad (33)
\]

\[
D(u, x, t) := \left\| \frac{\partial}{\partial t} (v - u)(x, t) \right\|^2_g \quad (34)
\]

\[
R(u, x, t) := \left\| (v - u)(x, t) \right\|^2_g \quad (35)
\]

The integral of \( E \) over the domain gives the Killing energy \( \int_M E u \, dA \) of the observer field \( u \) on \( M \). \( \left\| \cdot \right\|_g \) denotes the tensor and vector norms, respectively, depending on the argument, with respect to the metric \( g \).

**Discretization.** To solve Eq. 32, we discretize the manifold \( M \) with a triangle mesh. Vector components are stored at the vertices, and we label the corresponding points on the manifold \( M \) with \( x_1, x_2, \ldots, x_N \).

The Killing energy term \( E \) (Eq. 33) is computed by discretizing the Killing operator \( K \) (Eq. 25), to compute the covariant derivative terms of \( V \) in Eq. 25. The weights for all differential operators to evaluate \( Ku \) are stored in a matrix \( K \). We compute partial derivatives via triangle vertex 1-ring neighborhood stencils, computing weighted averages of function values at the mesh vertices. The weights are independent of the actual function values, and can therefore be precomputed for a fixed mesh geometry and used for any function on \( M \). See App. R.

The observed time derivative term \( D \) (Eq. 34) is computed from partial derivatives alone (Eqs. 30 and 31). We compute time derivatives via central differences (in \( t \)) on each vertex. The weights for all differential operators to compute \( D \) are stored in a matrix \( D \). The regularizer term \( R \) (Eq. 35) is represented by a matrix \( N \), which is simply an identity matrix times \(-1\), to compute \( -u \). All terms involving the (known) input field \( v \) are precomputed and stored in the corresponding locations of the matrices \( K, D, \) and the right-hand side \( b \) (see below).

### 9.2 Global Least-Squares Solution

To solve Eq. 32, we are looking for the minimizer of the least-squares problem \( \min_{u} \frac{1}{2} \left\| Au - b \right\|^2_{g^u} \), with matrix \( A \) and vector \( b \) as given below. The least-squares solution, weighted with the metric \( g \), is the solution of the normal equations of the weighted least-squares problem given by the square, positive-definite system \( A^T W A u = A^T W b \), with

\[
A := \begin{pmatrix} K & D \\ D & N \end{pmatrix}, \quad W := M \begin{pmatrix} G & 0 \\ 0 & \lambda G \end{pmatrix} \begin{pmatrix} 0 & \mu G \\ 0 & 0 \end{pmatrix}, \quad b := \begin{pmatrix} \begin{pmatrix} 0 \\ \frac{\partial}{\partial t} \end{pmatrix} \end{pmatrix}
\]

(36)

The matrix \( M \) is diagonal, with weights giving the area on \( M \) assigned to each vertex element \( x_i \), discretizing integration over area elements \( dA \). The other blocks of the matrix \( W \) encode \( g \) on \( M \), given by the blocks

\[
\bar{G} := \begin{pmatrix} g_{i1}^{-1} & 0 & \cdots & 0 \\ 0 & g_{i2}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{N}^{-1} \end{pmatrix}, \quad \bar{S} := \begin{pmatrix} g_{i1} & 0 & \cdots & 0 \\ 0 & g_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{N} \end{pmatrix}
\]

(37)

Here, \( g_{ij} \) refers to the \( n \times n \) component matrix \( g_{ij} \) of the metric \( g \) at the vertex \( x_i \in M \). For efficient computation of the tensor norm \( \left\| T \right\|_g \), we use the following approach to convert the tensor inner product in Eq. 27 to a simple contraction of two first-order tensors. We introduce \( n \)-dimensional first-order tensors \( T, S \), and a symmetric \( n \)-dimensional second-order tensor \( g^{ij} \), where \( n := n^2 \). That is, \( n = 4 \) for \( n = 2 \). We denote the \( n \times n \) components of \( g^{-1} \) by \( g^{ij} \), computed from \( g^{ij} \) as

\[
\bar{g} := g^{ik} g^{jk}, \quad x, t \in [1, \bar{n}], \quad (i, j) = \sigma^{-1}(s), (k, l) = \sigma^{-1}(t).
\]

(38)

The indices \( i, j, k, l \) are given by the inverse of the indexing function

\[
\sigma(i, j) := n \cdot (i - 1) + j, \quad \text{with } i, j \in [1, n].
\]

(39)

We can use this to rearrange the tensor inner product, to obtain

\[
T_{\bar{i}} := T_{\bar{k}l}, \quad t = \sigma(k, l), \quad S_{\bar{i}} := S_{\bar{s}j}, \quad s = \sigma(i, j),
\]

(40)

\[
T^\alpha = g^\alpha T_{\bar{i}}, \quad \text{and therefore}, \quad T^\alpha S_{\bar{s}} = g^\alpha g^{\beta\gamma} T_{\bar{k}l} S_{\bar{s}j} = T^\beta S_{\bar{\beta}j}.
\]

Thus, the tensor inner product \( (T, S) \) (Eq. 27) can simply be computed as the contraction \( T^\alpha S_{\bar{s}} \). Using this approach, the tensor norm of the Killing energy in the least-squares problem given above is computed by inserting the components \( \bar{g} \) (Eq. 38) of \( \bar{g} \) into the matrix \( \bar{G} \) above.

### 9.3 Multi-Chart Optimization

At each vertex \( x_i \in M \), there is only one vector for the field \( u \). We store vector components with respect to one selected chart for each \( x_i \). However, finite differences to estimate derivatives (App. R) have to be computed in the same chart, even when adjacent vertices are in different charts. We enable this by “baking” all required chart transition maps, i.e., the \( n \times n \) Jacobian matrices transforming from the coordinate basis of one chart to that of another chart, into the matrices \( K, D, \) and \( N \). The elements of these matrices are embedded such that all finite differences are computed using vector components referencing the same chart. More details for implementing this approach are explained in App. S.

### 9.4 Objectivity of Intrinsic Observer Field Computation

**Remark.** The computation of the observer field \( u \) as the minimizer of Eq. 32 is objective with respect to any observer transformation \( t \mapsto g(t) \in G \), with \( G = \text{Isom}(M) \) the isometry group of the manifold \( M \).

We give proofs in App. F. Except point (4) below, the reasons for this are quite intuitive and can be summarized as follows: (1) The norms used in Eq. 32 are invariant under isometries, because by definition any isometry does not change them (Eq. 14). (2) The Killing energy \( \int_M E u \, dA \) is time-independent, and measures how much the field \( u \) differs from an infinitesimal isometry. Since observer transformations are exact isometries of \( M \), they do not change the Killing energy. (3) A time-dependent observer transformation \( t \mapsto \tilde{g}(t) \), which with \( G = \text{Isom}(M) \) is a time-dependent isometry \( t \mapsto \tilde{g}(t) \) of \( M \), changes any velocity field on \( M \) by point-wise addition of the Killing field corresponding to the derivative \( X(t) \) of the isometry. However, the Killing field that is added due to the observer transformation cancels out in the relative velocity field \( (v - u) \). (4) The objectivity of the observed time derivative (Eq. 1) is far from obvious. However, Marsden and Hughes [45, p.101] prove that the time-dependent Lie derivative of an objective tensor field with respect to the flow of any vector field is objective. See App. F3.

That is, the norm of all terms in Eq. 32 does not change under the isometry \( t \mapsto \tilde{g}(t) \) of any observer transformation \( t \mapsto g(t) \), and because the minimizer of Eq. 32 is unique, as in [30], its computation is thus objective with respect to the group action \( \Phi \) of the isometry group of \( M \).

### 10 RESULTS

We have implemented a general framework for flow visualization on curved surfaces, given as triangle meshes, in C++, and OpenGL for rendering. For the sphere, we use six very simple charts, as described in detail in App. Q. We emphasize that the only 3D computations that we use are for initially obtaining the metric of the sphere as a 2D tensor field in each of the charts, and for visualization purposes in OpenGL. All our vector fields are given via 2D components, and all computations required for solving Eq. 32 are fully intrinsic in 2D. We solve the least-squares system given in Sec. 9.2 by using the conjugate gradient solvers provided by the C++ Eigen library, obtaining the minimizer \( u \) in multiple iterations. We have also tried direct methods using matrix factorization of the symmetric positive definite matrix \( A^T W A \). Due to the size of our grids, conjugate gradients obtain the minimum faster.
Table 1. Data sets with grid/mesh resolution.

<table>
<thead>
<tr>
<th>Data set</th>
<th># Vert.</th>
<th># Tri.</th>
<th>Time</th>
<th>Time steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNTHETIC VORTEX</td>
<td>10,242</td>
<td>20,480</td>
<td>0.2s</td>
<td>100</td>
</tr>
<tr>
<td>SYNTHETIC FOUR CENTERS</td>
<td>10,242</td>
<td>20,480</td>
<td>0.2s</td>
<td>100</td>
</tr>
<tr>
<td>JUPITER VORTEX STREET</td>
<td>40,962</td>
<td>81,920</td>
<td>0.3s</td>
<td>300</td>
</tr>
<tr>
<td>EARTH FLOW</td>
<td>163,842</td>
<td>327,680</td>
<td>8 days</td>
<td>32</td>
</tr>
<tr>
<td>EARTH FLOW (SUBDOMAIN)</td>
<td>32,400</td>
<td>64,796</td>
<td>8 days</td>
<td>32</td>
</tr>
<tr>
<td>EARTH FLOW (ADAPT. RES.)</td>
<td>62,412</td>
<td>124,820</td>
<td>8 days</td>
<td>32</td>
</tr>
</tbody>
</table>

Data sets. Table 1 gives an overview of the data sets that we have used in this paper and the video. We ran experiments with synthetic data sets and with simulated data on the sphere. The optimizations are global. For the earth flow data set we have, however, experimented with two approaches for region of interest optimization to reduce computation times. Table 2 gives computation times for the conjugate gradient solvers to optimize observer fields $u$. All computations were run on two Intel Xeon E5-2600 processors with 16 cores running at 2.3GHz.

Visualization. We have adapted the observer-relative visualization methods described by Hadwiger et al. [30] from Euclidean space to curved spaces. Since we need coordinate charts to compute the optimization, we can also use them as texture parametrization for LIC computations in texture space on the GPU. Corresponding to a given observer field $u$, we compute LIC images of observed stream lines [30]. Likewise, we also compute the geometry of integral curves for observed stream lines as well as for observed path lines. To visualize the time parameter of path lines, we increase the radial distance from the sphere according to the parameter $t$, scaled by a user-specified factor. This is helpful to see the time evolution of path lines. Additionally, we draw path line shadows on the sphere, to ensure that their position in space can be correctly interpreted. Moreover, we visualize time-dependent scalar fields relative to the observer field via color coding, and in the same way show texture patterns, e.g., Earth or Jupiter textures. We can dynamically switch the observer for any flow field, and directly visualize the corresponding observed fields. Figs. 1, 6, and 9 show observer-relative scalar fields (planet textures, vorticity magnitude), as well as observed LIC. Figs. 3, 6, and 8 show observed path lines.

10.1 Synthetic Data Sets

Synthetic Vortex. We transform a steady input field of a simple vortex into different time-dependent fields by rotating the input around the vertical axis of a sphere at different angular velocities (Fig. 3). The latter correspond to different scalings of the Killing field $x_2$ in Fig. 5.

From each transformed, time-dependent input field, we compute the corresponding observer field $u$ via optimization. In all cases, the minimizer $u$ of Eq. 32 is a steady Killing field. The observer fields then enable recovering the original steady vortex in Fig. 3 (d). Relative to its observer field, each unsteady field becomes the original steady field.

Synthetic Four Centers. Fig. 8(a) shows four vortices that rotate around each other clockwise, around a common point in the center of the sphere. We show that the observer field $u$ that is computed automatically via optimization recovers a steady field for the four vortices, as depicted in Fig. 8(b). In Fig. 8(c), we then observe the input field in (a) relative to a leftward rotation around the vertical axis of the sphere. Fig. 8(e) again depicts the field that is recovered automatically, this time relative to an observer field $u$, shown in Fig. 8(d), that rotates clockwise as well as rightward around the vertical axis of the sphere.

10.2 Global Planet-Scale Optimization

Jupiter Vortex Street. Fig. 9 depicts a 2D von Kármán vortex street centered around the equator of Jupiter. We have generated this data set for testing purposes by mapping a flat vortex street to a sphere by directly mapping a Cartesian x-y grid to a latitude-longitude grid.

Earth Flow. For real planet-scale experiments we use open source flow data from the EU Copernicus project. We use global wind data to show that large phenomena like hurricane Isabel (2003) are automatically tracked by the solution of our optimization problem, although no explicit feature tracking is performed. Figs. 1 and 6 show the input flow data and the observed flow fields. We are able to compute an observed flow field that is nearly steady. To get an undistorted visualization that is centered on the hurricane, we can pick one time-dependent Killing field as the observer that travels with the hurricane. The resulting visualization applies the inverse observer motion to the Earth texture.

10.3 Region of Interest Optimization

Our approach optimizes for the best observer field $u$ with a global optimization. However, often an optimization of the whole domain at the same resolution level is not necessary, since the user might only be interested in a smaller subdomain. We have tested two methods for this case. We either select a subdomain and run the global optimization only in this subdomain, or we adaptively remesh to obtain different, adaptive mesh resolutions for different regions on the sphere, and then run the optimization over the whole domain. The computation times can be greatly reduced from a full-resolution grid either way (Table 2).

11 Discussion and Conclusions

We have presented the first objective, optimization-based framework for computing time-dependent reference frames in curved spaces with the goal of turning unsteady input flow fields into (nearly) steady fields relative to a global observer field. A major motivation for this approach is to enable feature-relative visualization and analysis of the time evolution of features, such as scalar and vector field attributes of hurricane simulations, “in place.” A powerful example of this is shown in Fig. 1 (top left), where the path of hurricane Isabel is simply a path line of the automatically computed observer field $u$. We have seeded this path line manually, but this could also be done algorithmically. Moreover, path lines of $u$ anywhere inside the core region of the hurricane follow the general time-dependent diffeomorphism that constitutes the warping domain of the hurricane. We believe that our mathematical framework has great potential in applications such as studying features of interest in climatology and oceanography in a reference frame-independent way. Our method is objective, meaning that it is invariant with respect to any isometric observer transformation of the input flow field. This enables it to automatically follow the motion of features of interest, and makes it independent of super-imposed motions, including those of moving measurement devices capturing real-world geophysical flow phenomena. Finally, we believe that the fully intrinsically approach that we have taken, and the methodology from Riemannian geometry that we have used in our framework, could provide significant advantages and insight to other areas and applications of flow visualization as well.

Acknowledgments

We thank Anna Frühstück for the illustrations and for help with the figures and the video. Hurricane Isabel data courtesy of EU Copernicus project, path from National Hurricane Center/Wikipedia. This work was supported by King Abdullah University of Science and Technology (KAUST), and the KAUST Office of Sponsored Research (OSR) award OSR-2015-CCF-2533-01. This research used resources of the Core Labs of KAUST.


and-atmospheres; posted 27-June-2018].
[57] J. Solomon. Discrete killing fields for pattern synthesis and symmetry
detection. Undergraduate honors thesis, Stanford University, Palo Alto,
CA, 2010.


Berlin (ZIB), Berlin, Germany, 1997.

[60] D. Stalling and H.-C. Hege. Fast and resolution independent line integral

In *Proceedings of the 12th Computational Fluid Dynamics Conference*,


Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics.* Princeton

[64] Y. Tong, S. Lombeyda, A. N. Hirani, and M. Desbrun. Discrete multiscale
vector field decomposition. In *Proceedings of SIGGRAPH*, pp. 445–452,
2003.

Springer-Verlag, 1965.

[66] L. W. Tu. *Differential Geometry: Connections, Curvature, and Character-


[69] A. Vaxman, M. Campen, O. Diamanti, D. Panozzo, D. Bommes, K. Hilde-
brandt, and M. Ben-Chen. Directional Field Synthesis, Design, and Pro-

particle motion in unsteady flows. *IEEE Transactions on Visualization