Generic Variational Spacetime Optimization of Vortex Core Manifolds

XINGDI ZHANG, King Abdullah University of Science and Technology (KAUST), Saudi Arabia
PETER RAUTEK, King Abdullah University of Science and Technology (KAUST), Saudi Arabia
MARKUS HADWIGER, King Abdullah University of Science and Technology (KAUST), Saudi Arabia

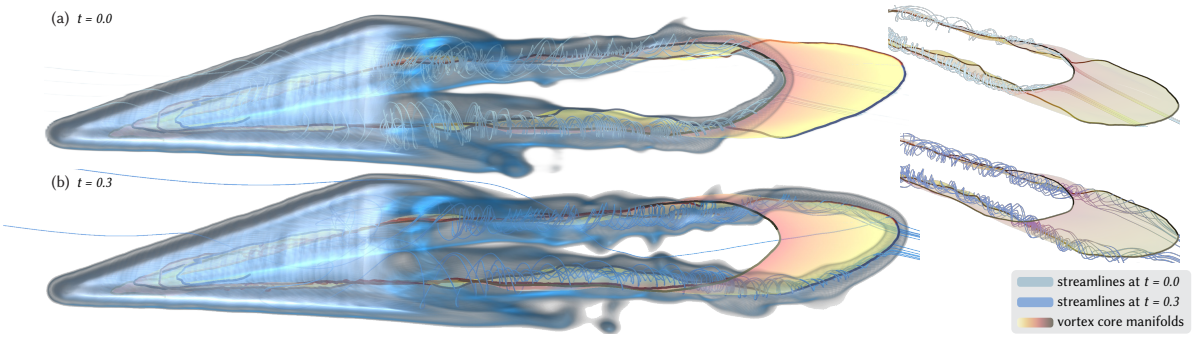

Fig. 1. Vortex cores in 3D fluid flow, such as the CFD simulation of airflow over a delta-shaped wing shown here, are curves in each time step, but trace out spacetime surfaces over time (orange-yellow color coding). Here, they are visualized in the context of an IVD (instantaneous vorticity deviation) volume rendering, together with streamlines in a corresponding optimal reference frame at times (a) $t = 0$; (b) $t = 0.3$. The Lagrangian evolution of vortex core lines in spacetime perfectly matches the instantaneous streamlines in the optimal reference frame as well as regions of high vorticity over the entire time interval.

The detection of vortex structures in fluid flow is a crucial task in continuum mechanics and flow visualization. However, vortex detection is an extremely challenging task that, despite its importance, is not yet fully solved for 3D unsteady flow fields, due to the complexities introduced by the time-dependence of unsteady flow. We introduce a generic variational framework for the computation of optimal vortex cores in 3D unsteady flow that combines a geometric vortex core model with explicit reference frame optimization. Instead of focusing on a specific vortex detection criterion, we use a generically defined Lagrangian that can incorporate different vortex criteria in a unified way. A key insight of our framework is that the two-manifolds comprising 3D vortex cores in spacetime can be obtained by solving the Euler-Lagrange equations in a single time step with only one independent variable. This is enabled by a Lagrangian that is pre-integrated in time according to the pushforward of the underlying flow. The combination of temporal pre-integration and solving for the optimal two-manifold using only one spatial parameter results in an extremely efficient algorithm.

CCS Concepts: • **Human-centered computing → Scientific visualization**.

Additional Key Words and Phrases: Flow Visualization, Vortex Core Extraction, Variational Methods, Lagrangian Coherent Structures, Objectivity.

ACM Reference Format:
Xingdi Zhang, Peter Rautek, and Markus Hadwiger. 2026. Generic Variational Spacetime Optimization of Vortex Core Manifolds. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3799902.3811230

Authors' Contact Information: Xingdi Zhang, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, xingdi.zhang@kaust.edu.sa; Peter Rautek, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, peter.rautek@kaust.edu.sa; Markus Hadwiger, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, markus.hadwiger@kaust.edu.sa.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/26/07
https://doi.org/10.1145/3799902.3811230

1 Introduction

The detection and analysis of vortex structures in fluid flow is a fundamental task across continuum mechanics, aerodynamics, climate science, and many other scientific disciplines. Vortices are the drivers of fluid motion, responsible for phenomena ranging from mixing and drag generation to transport. Despite their ubiquity, identifying these features in a mathematically rigorous and robust manner remains one of the most persistent challenges in flow visualization. While the intuition of a vortex as swirling motion around a core is simple, translating it into a consistent definition that holds up under time-dependent dynamics, and at the same time is invariant to reference frame changes, is notoriously difficult. As simulation sizes grow and unsteady 3D flow fields become the norm in many scientific domains, researchers require methods that are geometrically precise, physically consistent, and that scale computationally.

Ideally, a vortex core definition should satisfy two rigorous constraints [Günther and Theisel 2018]: It must be *objective* (i.e., invariant to changes of observer or reference frame), and *Lagrangian* (i.e., the core moves with the flow). The current state of the art often forces a trade-off between these properties and computational efficiency. Classical Eulerian methods, such as the Q [Hunt et al. 1988] or λ_2 [Jeong and Hussain 1995] criteria, respectively, are fast but lack objectivity as well as the property of being Lagrangian.

To address objectivity, reference frame optimization methods, such as *Generic Objective Vortices* [Günther et al. 2017], optimize a spatially varying reference frame to make the flow appear as steady as possible. While this makes local criteria objective, it does not guarantee that the extracted features are Lagrangian: Core lines may drift

away from the material path of the fluid particles. Conversely, Lagrangian Coherent Structures (LCS) [Haller 2015] provide physical consistency but typically identify transport barriers (separatrices) rather than geometrical core lines, and they require expensive dense particle integration over the entire spatio-temporal domain.

Finding a method that is at once generic, supporting various vortex criteria, Lagrangian, and computationally efficient, remains an open problem. The primary friction is the complexity of spacetime optimization: Efficiently extracting a feature that is optimal over a time interval. For example, Daßler and Günther [2024] is a powerful variational framework that regularizes features, but vortex cores lack temporal coherence. Günther and Theisel [2025] use reference frames following path lines, which is a strong limitation for 3D flow.

Key insight. Our key enabler of efficiency is that the temporal evolution of a vortex as a Lagrangian feature can be fully encoded in the Lagrangian function L of a variational problem defined in a single time step. Instead of optimizing spacetime manifolds in 4D, we utilize the underlying flow and its pushforward to *pre-integrate* an energy function along flow trajectories. This projects the behavior of the flow over a time interval onto a single time step, effectively collapsing the time dimension of the variational problem. This allows us to find vortex cores as optimal spacetime surfaces by solving the Euler-Lagrange equations in only one time step, using an efficient spatial solver to obtain results that are optimal in spacetime. However, pre-integration does not mean that we have to pre-compute L in the complete phase space for optimization: We "pre"-integrate on the fly wherever the Lagrangian L and its derivatives are required.

A generic framework. Our unified framework can incorporate arbitrary vortex detection criteria, as one component of a generically defined energy function with pre-defined regularization terms. The corresponding optimization is then solved via pre-defined Euler-Lagrange equations independent of the particular vortex criterion.

Contributions. In summary, our main contributions are:

- A generic variational framework for the extraction of vortex cores in 3D unsteady flow that unifies different vortex criteria, physical consistency, and geometric constraints.
- Our geometric vortex core model combines explicit optimization of an underlying rigid reference frame with minimization of the superposed temporal vortex core deformation.
- Our temporal Lagrangian energy pre-integration reduces the 4D optimization problem of vortex cores in spacetime to a spatial variational problem in a single time step.

2 Related Work

Despite extensive research, a rigorous vortex definition remains elusive, especially for 3D unsteady flows where time-dependence complicates steady flow intuitions [Günther and Theisel 2018; Lugt 1979]. For overviews, see recent state-of-the-art reports [Bujack et al. 2020; Günther and Baeza Rojo 2021; Günther and Theisel 2018].

Local Eulerian vortex criteria commonly analyze the velocity gradient tensor $\nabla \boldsymbol{v}$, e.g., to find regions where rotation dominates strain. The Q-criterion [Chong et al. 1990; Hunt et al. 1988] is efficient but susceptible to shear, while the λ_2-criterion [Jeong and Hussain 1995] offers improved robustness by detecting pressure minima. Other invariants like the Δ-criterion [Chong et al. 1990] and swirling strength [Zhou et al. 1999] focus on complex eigenvalues. While popular, these region-based methods are threshold-dependent and lack explicit core lines. Line-based methods, such as Parallel Vectors (PV) [Roth and Peikert 1998, 1999; Sujudi and Haimes 1995], extract cores where $\boldsymbol{v} \parallel \nabla \boldsymbol{v} \cdot \boldsymbol{v}$, but often suffer from numerical instability and noise [Daßler and Günther 2024]. A major drawback of Eulerian methods is that they are often not invariant to Euclidean changes of reference frame [Günther et al. 2017; Haller 2005].

Reference frame optimization methods compute Euclidean or spatially varying frames in which the flow appears steady in an objective (frame-indifferent) way [Günther et al. 2017; Hadwiger et al. 2019; Haller 2021; Theisel et al. 2021], including Killing fields on curved surfaces [Rautek et al. 2021], interactive frameworks for 2D and 3D exploration of the space of reference frames [Zhang et al. 2026, 2022, 2025b], and using neural networks [Kim and Günther 2019; Zhang et al. 2025a]. Special cases of invariance have also been explored [Günther and Theisel 2016; Günther and Theisel 2019, 2020; Kaszás et al. 2023]. However, all these methods focus on the reference frame itself, not explicitly on feature geometry, and they do not guarantee Lagrangian features [Günther and Theisel 2025].

Lagrangian coherent structures (LCS) are transport barriers, often computed via finite-time Lyapunov exponents (FTLE) [Haller 2002, 2015; Shadden et al. 2005], which is computationally expensive and identifies separatrices rather than cores. Lagrangian-averaged vorticity deviation (LAVD) [Haller et al. 2016] integrates instantaneous vorticity deviation (IVD) along particle trajectories, giving an objective scalar field for thresholding or isocontour extraction.

Vortex filaments have been studied extensively, from fluid mechanics [Saffman 1992] to computer graphics, e.g., visualization using Clebsch maps [Chern et al. 2017], preserving vorticity via covector fields and Lie derivatives [Nabizadeh et al. 2022] or wave functions [Chern et al. 2016], considering the influence of domain topology via cohomology [Yin et al. 2023], implicit vortex filament representations [Ishida et al. 2022], filaments in bubble rings [Padilla et al. 2019] or smoke rings [Weißmann et al. 2014], or modeling flow with vortex shedding and reconnection [Weißmann and Pinkall 2010]. Vortex dynamics are also studied on surfaces [Zhu et al. 2025].

Variational feature extraction methods define features as energy minimizers. Daßler and Günther [2024] recently unified feature extraction in visualization, such as isocontours, ridges, or vortex cores, within a generic variational optimization framework. This greatly improves feature regularization, but they only consider vortex cores at individual times. Our approach similarly builds on a variational foundation, but uses temporally pre-integrated Lagrangians. This combines coherent Lagrangian behavior, modeled over spacetime manifolds, with the efficiency of Eulerian solvers.

3 Flow Visualization Background

Reference Frames. As in previous work [Günther and Theisel 2025; Hadwiger et al. 2019; Rautek et al. 2021], we model reference frames relative to which flow velocities $\boldsymbol{v}(x,t)$ are measured by an *observer field*, which is a time-dependent velocity field that we denote by $\boldsymbol{w}(x,t)$, given on the same manifold as the input field $\boldsymbol{v}(x,t)$.

Euclidean reference frames. For physical computations in Euclidean space, the observer field $\boldsymbol{w}$ must be an infinitesimal rigid

motion, i.e., the derivative of a rigid transformation, which is a Killing vector field. A vector field $\boldsymbol{w}$ is Killing, if its velocity gradient tensor $\nabla \boldsymbol{w}$ (the covariant derivative of the vector field $\boldsymbol{w}$) is antisymmetric at all spacetime points (x, t). That is, for all vectors $\boldsymbol{X}$, $\langle \nabla \boldsymbol{w}(\boldsymbol{X}), \boldsymbol{X} \rangle = 0$. The inner product $\langle \cdot, \cdot \rangle$ is determined by the metric tensor field in the flow domain, i.e., for Euclidean space $\mathbb{R}^3$ it is determined by the Euclidean metric. The above relationship must be fulfilled at every point $x \in \mathbb{R}^3$, for any tangent vector $\boldsymbol{X} \in T_x\mathbb{R}^3$.

Flow Field Properties. The **flow map** (or *flow*) of a vector field $\boldsymbol{v}(x, t)$ is the family of diffeomorphisms mapping all points $x \in U \subset M$, for a given manifold M, from time t_0 to the corresponding points at time t, following the motion given by $\boldsymbol{v}$. We denote the flow map by

$$\begin{aligned} \phi_{t_0,t} : M \supset U &\to \phi_{t_0,t}(U) \subset M, \\ x &\mapsto \phi_{t_0,t}(x). \end{aligned} \tag{1}$$

The flow map is determined by the solution curves corresponding to the initial value problems $\phi_{t_0,t_0}(x) = x_0$ of the ODEs

$$\left.\frac{d}{dt}\right|_{t=\tau} \phi_{t_0,t}(x) = \boldsymbol{v}(\phi_{t_0,\tau}(x), \tau). \tag{2}$$

The *pushforward* $\phi_{t_0,t*}$ of the diffeomorphism $\phi_{t_0,t}$ transforms tangent vectors $\boldsymbol{X}$ to any curve $s \mapsto x(s) \in M$, $\boldsymbol{X} = (d/ds)\, x(s)$, via

$$\boldsymbol{X}_{t_0} \mapsto \phi_{t_0,t*}\left(\boldsymbol{X}_{t_0}\right) = \boldsymbol{X}_t. \tag{3}$$

In flow visualization, e.g., FTLE computations, the pushforward of the flow map is known as the spatial (deformation) gradient $\nabla \phi_{t_0,t}$.

Observed velocity and velocity gradient. For any velocity field $\boldsymbol{v}$, and a reference frame given by an observer field $\boldsymbol{w}$, we obtain the instantaneous observed velocity $(\boldsymbol{v} - \boldsymbol{w})$, and the observed velocity gradient tensor $\nabla(\boldsymbol{v} - \boldsymbol{w})$, respectively, relative to $\boldsymbol{w}$.

Observed time derivative. The observed time derivative [Hadwiger et al. 2019; Rautek et al. 2021] of the observed velocity field $(\boldsymbol{v} - \boldsymbol{w})$, relative to $\boldsymbol{w}$, is given by the time-dependent Lie derivative

$$\frac{D}{Dt}(\boldsymbol{v} - \boldsymbol{w}) = \left(\frac{\partial}{\partial t} + \mathcal{L}_{\boldsymbol{w}}\right)(\boldsymbol{v} - \boldsymbol{w}). \tag{4}$$

We can compute this derivative by expanding the Lie derivative $\mathcal{L}_{\boldsymbol{w}}$ above using the covariant derivative ∇, computed with the standard formula that is valid for any symmetric connection (as the Euclidean connection) and the corresponding covariant derivative. This gives

$$\frac{D}{Dt}(\boldsymbol{v} - \boldsymbol{w}) = \frac{\partial \boldsymbol{v}}{\partial t} - \frac{\partial \boldsymbol{w}}{\partial t} + \nabla \boldsymbol{v}(\boldsymbol{w}) - \nabla \boldsymbol{w}(\boldsymbol{v}). \tag{5}$$

Like the vector fields $\boldsymbol{v}$ and $\boldsymbol{w}$, the corresponding observed time derivative $(D/Dt)(\boldsymbol{v} - \boldsymbol{w})$ is also a time-dependent vector field on M.

4 Geometric Vortex Cores in 3D Flow

The first important ingredient of our generic framework is a consistent geometric model for a vortex core in 3D flow.

4.1 Steady vs. Unsteady Flow

Before discussing unsteady flow, we consider the case of steady flow in detail, because many vortex detectors, including our framework, build on earlier work primarily aimed at steady flow. Moreover, any vortex detector for unsteady flow should also work for steady flow.

Steady flow. Steady 3D flow fields $\boldsymbol{v}(x, t)$ are characterized by $\partial \boldsymbol{v}(x, t)/\partial t = 0$. In this case, we can also denote the field by $\boldsymbol{v}(x)$, but later we derive our method independently of whether the flow is steady or unsteady, writing $\boldsymbol{v}(x, t)$ for both. For steady flows, vortex cores are often defined as specific curves around which the flow exhibits infinitesimal swirling behavior, characterized by two conjugate complex eigenvalues of the velocity gradient tensor $\nabla \boldsymbol{v}$, or via the curl $\nabla \times \boldsymbol{v}$. Given local swirling behavior, a curve is often [Günther and Theisel 2018; Sujudi and Haimes 1995] classified as a vortex core line if, at every point $x \in \mathbb{R}^3$ on the curve, locally

$$\boldsymbol{v}(x) \parallel \nabla \boldsymbol{v} \cdot \boldsymbol{v}(x) \iff \boldsymbol{v}(x) \parallel \boldsymbol{a}(x) \iff \boldsymbol{v}(x) \parallel \boldsymbol{e}_3(\nabla \boldsymbol{v}(x)). \tag{6}$$

The term $\nabla \boldsymbol{v} \cdot \boldsymbol{v}$ is the directional derivative of the field $\boldsymbol{v}$ in direction $\boldsymbol{v}$, which for steady flow is the acceleration $\boldsymbol{a}$ of the field $\boldsymbol{v}$. If $\nabla \boldsymbol{v}(x)$ has complex eigenvalues λ_1, λ_2, the above condition means that the flow velocity $\boldsymbol{v}(x)$ must be parallel to the real eigenvector $\boldsymbol{e}_3$ of $\nabla \boldsymbol{v}(x)$ with real eigenvalue λ_3. However, if the criterion is true and $\boldsymbol{v}(x) \neq 0$, because the acceleration is given by $\boldsymbol{a}(x) = \lambda_3 \boldsymbol{v}(x)$, vortex core lines with non-vanishing velocity must be rather straight [Roth and Peikert 1996]. The underlying issue is that this criterion models the vector field as locally linear, i.e., $\boldsymbol{v}(y) = \nabla \boldsymbol{v}(x) \cdot \boldsymbol{y}$, with $\boldsymbol{y} = y - x$. A more sophisticated model for bent vortex cores uses higher-order derivatives of the velocity field [Roth and Peikert 1998]. These additional derivatives can, however, incur numerical difficulties [Günther and Theisel 2018]. Vortex core lines that fulfill criteria such as these are often extracted by tracing curves using the *parallel vectors (PV) operator* [Roth and Peikert 1999] or variants.

Using an explicit geometric description of vortex cores as smooth parameterized space curves $s \mapsto c(s) \in \mathbb{R}^3$, $s \in \mathbb{R}$, they should obey

$$\boldsymbol{v}(c(s)) = k(s) \cdot \frac{d}{ds} c(s), \quad k(s) \in \mathbb{R}. \tag{7}$$

Thus, fluid particles exactly on the core line must move tangentially to it. This includes stationary particles, with $\boldsymbol{v}(c(s)) = 0$ at critical points of $\boldsymbol{v}$. We require the explicit curve $s \mapsto c(s)$ to be *regular*, i.e., $(d/ds)\, c(s) \neq 0$. Thus, critical points on the core line must have $k(s) = 0$, not $(d/ds)\, c(s) = 0$. Comparing to Eq. 6, if $\nabla \boldsymbol{v}(c(s))$ has complex eigenvalues λ_1, λ_2, for $k(s) \neq 0$ Eq. 6 is fulfilled when $(d/ds)\, c(s) \parallel \boldsymbol{e}_3(\nabla \boldsymbol{v}(c(s)))$. For $k(s) = 0$, Eq. 6 is always fulfilled.

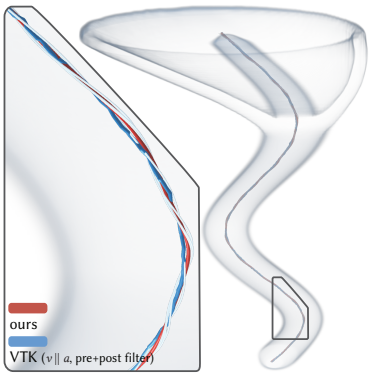

Fig. 2. **Steady tornado.** Our method naturally unifies steady and unsteady flow fields. Our vortex core line (red) in this steady flow field is confirmed with streamlines tightly swirling around it, and is slightly smoother than state of the art vortex core extraction in VTK [Schroeder et al. 2006], which gives a slightly jagged core line (blue), with pre- and post-filtering applied.

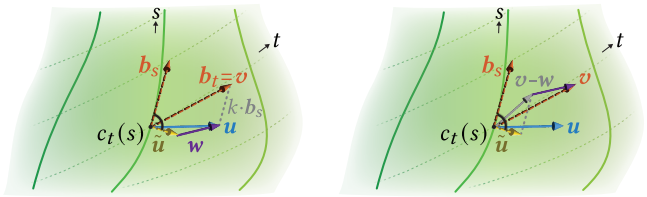

Fig. 3. **Vortex cores as 2-manifolds** in spacetime, parameterized by (s, t). At each time t, a vortex core is a curve $s \mapsto c_t(s)$. The velocities $\boldsymbol{v}$ of the particles comprising the core, as well as $\boldsymbol{u}$, are tangent to the 2-manifold. We decompose the velocity $\boldsymbol{v} = k_t(s) \cdot \boldsymbol{b}_s + \tilde{\boldsymbol{u}} + \boldsymbol{w}$, to be able to minimize $\tilde{\boldsymbol{u}}$.

Unsteady flow. We can generalize Eq. 7 to unsteady 3D flows, where $\partial \boldsymbol{v}(x,t)/\partial t \neq 0$, by defining that for each time t a vortex core is given by a smooth regular curve $s \mapsto c_t(s)$. Additionally, we require that the curves change smoothly with time t. Unlike for steady fields, even the particles on the vortex core now have to be allowed to move non-tangentially, to permit the core itself to move over time. All particles on a vortex core should therefore now obey

$$\boldsymbol{v}(c_t(s), t) = k_t(s) \cdot \frac{\partial}{\partial s} c_t(s) + \frac{\partial}{\partial t} c_t(s), \quad k_t(s) \in \mathbb{R}. \tag{8}$$

This allows particles to move tangentially to the core line, and at the same time allows motion of the core line itself via the vector $(\partial/\partial t)\, c_t(s)$. However, Eq. 8 does not guarantee that the vectors $(\partial/\partial s)\, c_t(s)$ and $(\partial/\partial t)\, c_t(s)$ are linearly independent. We will therefore define a consistent geometric model for 3D vortex cores in both steady and unsteady fields, as well as for unsteady fields where the vortex core itself might not move, as 2-manifolds in 4D spacetime.

4.2 Vortex Cores as Spacetime 2-Manifolds

To model vortex cores as 2-manifolds in 4D *spacetime*, with spacetime simply as $\mathbb{R}^3 \times \mathbb{R}$, we define smooth parameterized surfaces

$$(s, t) \mapsto \begin{pmatrix} c_t(s) \\ t \end{pmatrix} \in \mathbb{R}^3 \times \mathbb{R}, \quad s, t \in \mathbb{R}. \tag{9}$$

The partial derivatives of Eq. 9 give the basis vector fields (Fig. 3)

$$\boldsymbol{b}_s(s,t) = \begin{pmatrix} \frac{\partial}{\partial s} c_t(s) \\ 0 \end{pmatrix}, \quad \boldsymbol{b}_t(s,t) = \begin{pmatrix} \frac{\partial}{\partial t} c_t(s) \\ 1 \end{pmatrix}. \tag{10}$$

Because the curves $s \mapsto c_t(s)$ are required to be regular, the vectors $\boldsymbol{b}_s, \boldsymbol{b}_t$ are guaranteed to be linearly independent, even when the vortex core does not move over time. Using $\boldsymbol{b}_s, \boldsymbol{b}_t$, Eq. 8 becomes

$$\begin{pmatrix} \boldsymbol{v}(c_t(s), t) \\ 1 \end{pmatrix} = k_t(s) \cdot \boldsymbol{b}_s(s,t) + \boldsymbol{b}_t(s,t). \tag{11}$$

However, Eq. 11 is under-constrained for modeling vortex cores: Although not every surface $(s, t) \mapsto (c_t(s), t)$ fulfills Eq. 11, by setting $k_t(s) = 0$ we can see that *any* path surface does. The *path surface* corresponding to a seeding curve $s \mapsto c_{t_0}(s)$ at time t_0 is

$$(s, t) \mapsto \begin{pmatrix} c_t(s) \\ t \end{pmatrix} = \begin{pmatrix} \phi_{t_0,t}\left(c_{t_0}(s)\right) \\ t \end{pmatrix}. \tag{12}$$

This definition results in a *material parameterization* of the path surface, labeling each particle with a fixed parameter s over time.

Taking the partial derivatives of Eq. 12 gives the corresponding basis

$$\boldsymbol{b}_s(s,t) = \begin{pmatrix} \phi_{t_0,t*}\left(\frac{\partial}{\partial s} c_{t_0}(s)\right) \\ 0 \end{pmatrix}, \; \boldsymbol{b}_t(s,t) = \begin{pmatrix} \boldsymbol{v}(\phi_{t_0,t}(c_{t_0}(s)), t) \\ 1 \end{pmatrix}. \tag{13}$$

It is a crucial baseline of our unified approach that although our pre-integration always follows Eqs. 12, 13, while measuring the (zero) deformation of a core line $c_{t_0}(s) = c(s)$ in *steady* flow, the second basis vector will be changed naturally to $\boldsymbol{u} = (\boldsymbol{0}, 1)^T$ (cf. Sec. 5.1).

5 Method

Our framework consists of the following major components (Fig. 4).

Vortex core deformation. In order to define a generic framework that is able to capture the complex time-dependent dynamics of vortex cores in unsteady 3D flows, our first goal is to explicitly model, and thus to be subsequently able to minimize, the *deformation* of a vortex core over time (Sec. 5.1). To this end, we decompose a given flow field $\boldsymbol{v}$ into a sum of three different vector fields (Fig. 3):

- An underlying rigid motion, as a Killing field $\boldsymbol{w}$, which minimizes the observed time derivative on and around the core.
- A vector field $k_t(s) \cdot (\partial/\partial s)\, c_t(s)$ tangential to the (unknown) vortex core, capturing particle motion along the core line, including possible deformation tangential to the core line.
- A non-rigid deformation vector field $\tilde{\boldsymbol{u}}$ orthogonal to the (unknown) vortex core, which is the part of the velocity $\boldsymbol{v}$ that is not captured by the sum of the two previous fields.

Optimal local reference frame. We optimize the Killing field $\boldsymbol{w}$ for a time-dependent region $U(t)$ around a vortex core line candidate (Sec. 5.2). This region can be interactively selected, e.g., using a vortex lens [Rautek et al. 2024], or be sampled on a spatial grid with gradient descent for refinement from grid points to seed points. Figs. 4 (a,b) depict starting with a spatial search grid of candidate seed points, and refining seed point positions by simultaneously optimizing for a locally optimal reference frame as well as a refined seed point position for starting the actual vortex core optimization.

Energy pre-integration in time. A key insight of our framework is that *whatever vortex criterion is employed* at any point on a vortex core manifold, because vortex cores should be Lagrangian structures, i.e., be advected with the flow, all criteria and their corresponding energies (Sec. 5.3.3) can be pre-integrated in time according to the flow map of the underlying flow field $\boldsymbol{v}$, to define the Lagrangian L used in our spatial variational problem (Sec. 5.3.2).

We note that *pre*-integration in our context refers to the fact that the Lagrangian L does not depend on time. We do *not* actually need to pre-compute or store L for the entire phase space $(\boldsymbol{q}, \dot{\boldsymbol{q}})$, but "pre"-integrate on the fly wherever L or its derivatives are required.

Variational problem. Because we define a Lagrangian L resulting from energies that are pre-integrated in time, we can employ a phase space (Sec. 5.3.1) that corresponds to spatial curves $s \mapsto c_{t_0}(s)$, instead of to spacetime surfaces $(s, t) \mapsto (c_t(s), t)$, and efficiently solve one-parameter Euler-Lagrange equations to obtain the optimal curves that correspond to optimal spacetime surfaces (Figs. 4 (c,d)).

Vortex core spacetime manifolds. Once a vortex core is obtained as the spatial solution curve $s \mapsto c_{t_0}(s)$ of the Euler-Lagrange equations at time t_0 (Fig. 4 (c)), the corresponding spacetime surface

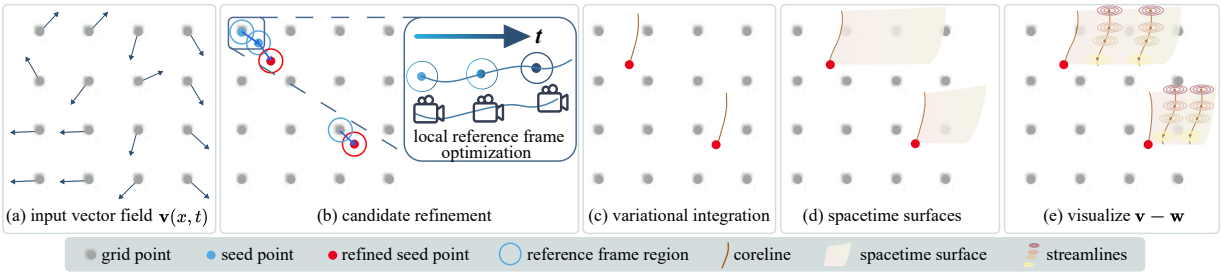

Fig. 4. **Pipeline.** (a) Given a time-dependent input flow field $\boldsymbol{v}$, we generate candidate seed points on a regular grid. (b) We then refine the candidate positions through a gradient descent approach: For each candidate seed point, we alternate between estimating the local time-dependent reference frame $\boldsymbol{w}$ that minimizes the observed time derivative, and refining the candidate position. (c) Once we found an optimal combination of reference frame and seed point, we integrate a vortex core line in space using our variational approach, solving the Euler-Lagrange equations. (d) Advecting core lines by the flow gives the corresponding spacetime surfaces. (e) Vortex core manifolds visualized together with streamlines in the optimal local reference frame $\boldsymbol{w}$ of each vortex core.

(Fig. 4 (d)) is simply determined by advecting the curve $s \mapsto c_{t_0}(s)$ through time according to the flow map of the flow field $\boldsymbol{v}$ (Eq. 12).

5.1 Vortex Core Deformation

We want to measure the deformation of the curve $s \mapsto c_t(s)$ over time, such that only the deformation component orthogonal to the curve is measured, while the deformation component tangential to it is not taken into account. The motivation is that our vortex core definition should allow unrestricted motion tangential to vortex core lines, for example due to physical effects such as vortex stretching.

Since a material parameterization (Eq. 12) results in $k_t(s) = 0$, it hides tangential motion along the vortex core. To be able to measure the deformation of vortex core lines, we thus require a more suitable basis at each point (s,t) of the path surface, allowing us to define that *a particle is moving* along *the core line if and only if* $k_t(s) \neq 0$. We introduce a new velocity field $\boldsymbol{u}$, and change Eq. 11 to the basis

$$\begin{pmatrix} \boldsymbol{v}(c_t(s),t) \\ 1 \end{pmatrix} = k_t(s) \cdot \begin{pmatrix} \frac{\partial}{\partial s} c_t(s) \\ 0 \end{pmatrix} + \begin{pmatrix} \boldsymbol{u}(c_t(s),t) \\ 1 \end{pmatrix}. \tag{14}$$

The velocity field $\boldsymbol{u}$ (see Fig. 3) models the *motion of the vortex core itself*, whereas the remaining term with $k_t(s)$ models the *motion of particles along the vortex core*. We further decompose the field $\boldsymbol{u}$ as

$$\begin{pmatrix} \boldsymbol{u}(c_t(s),t) \\ 1 \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{u}}(c_t(s),t) \\ 0 \end{pmatrix} + \begin{pmatrix} \boldsymbol{w}(c_t(s),t) \\ 1 \end{pmatrix}. \tag{15}$$

The *deformation vector field* $\tilde{\boldsymbol{u}}$ models the deformation of $s \mapsto c_t(s)$ superposed on an underlying Killing field $\boldsymbol{w}$, which, in general, is *not tangent* to the path surface (Eq. 12). Because $\boldsymbol{u}$ is tangent, this requires the field $\tilde{\boldsymbol{u}}$ to compensate. To only take into account the deformation not tangential to the curve, which is already accounted for in the factor $k_t(s)$ in Eq. 14, we are allowed to impose the condition

$$\tilde{\boldsymbol{u}}(c_t(s),t) \perp \frac{\partial}{\partial s} c_t(s). \tag{16}$$

Now, since the deformation vector field $\tilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{w}$ (Eq. 15) is orthogonal to $s \mapsto c_t(s)$, and because the observed velocity $\boldsymbol{v} - \boldsymbol{w}$ must be a linear combination of $(\partial/\partial s)\, c_t(s)$ and $\tilde{\boldsymbol{u}}$ (Eqs. 14, 15), we can derive that we can in fact compute $\tilde{\boldsymbol{u}}$ *without knowing the field* $\boldsymbol{u}$ via

$$\tilde{\boldsymbol{u}}(c_t(s),t) = (\boldsymbol{u}-\boldsymbol{w})(c_t(s),t) = \tag{17}$$
$$(\boldsymbol{v}-\boldsymbol{w})(c_t(s),t) - \left\langle (\boldsymbol{v}-\boldsymbol{w})(c_t(s),t), \frac{\partial}{\partial s} c_t(s) \right\rangle \frac{(\partial/\partial s)\, c_t(s)}{\|(\partial/\partial s)\, c_t(s)\|^2}.$$

This also shows that the optimal choice for $\boldsymbol{w}$ in Eq. 15 would give an observed time derivative that is zero or tangential to $s \mapsto c_t(s)$, because then a core line aligned with $\boldsymbol{v} - \boldsymbol{w}$ will not have to deform. However, before the actual optimization we do not know $s \mapsto c_t(s)$, and therefore we *minimize* the observed time derivative (Eq. 19).

We note that the last vector components in our spacetime equations naturally identify objective vectors (last comp. = 0) vs. (non-objective) velocity vectors (last comp. = 1). Thus, minimizing the deformation of the curve $s \mapsto c_t(s)$ via the magnitude of the deformation vector $\tilde{\boldsymbol{u}}$ is objective, and in any reference frame comes as close as possible to the steady case where a vortex core cannot deform non-tangentially. Moreover, we minimize the magnitude of the vector field $\tilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{w}$ using only the fields $\boldsymbol{v}, \boldsymbol{w}$ in Eq. 17 (Fig. 3).

5.2 Locally Optimal Reference Frames and Seed Points

Figs. 4 (a,b) depict our approach to start with a grid at time t_0 to initialize a set of candidate points. At each point, a local reference frame $\boldsymbol{w}$ is computed that minimizes the observed time derivative within a spatial region $U(t)$ that is advected with the point over time $t \in [t_0, T]$. A vector field $\boldsymbol{v}(x,t)$ is *steady with respect to some reference frame*, given by a field $\boldsymbol{w}(x,t)$, if at all points $x \in \mathbb{R}^3$, and for all times t, the observed time derivative (Eq. 4) vanishes, i.e.,

$$\frac{D}{Dt}(\boldsymbol{v}-\boldsymbol{w})(x,t) = 0. \tag{18}$$

If the field $\boldsymbol{w} = 0$, Eq. 18 becomes the $\partial \boldsymbol{v}/\partial t = 0$ of a steady vector field. For a general unsteady flow field $\boldsymbol{v}(x,t)$, and a Killing field $\boldsymbol{w}$ generating a Euclidean reference frame, Eq. 18 usually cannot be fulfilled exactly over a finite region. We thus aim to find the Killing field $\boldsymbol{w}$ that *minimizes* the observed time derivative integrated over a region $U(t)$ and a time interval $t \in [t_0, T]$. We determine $\boldsymbol{w}$ by

$$\boldsymbol{w}(x,t) = \operatorname{argmin}_{\tilde{\boldsymbol{w}}} \int_{t_0}^{T} \int_{U(t)} \left\| \frac{D}{Dt}(\boldsymbol{v}-\tilde{\boldsymbol{w}})(x,t) \right\|^2 dx\, dt. \tag{19}$$

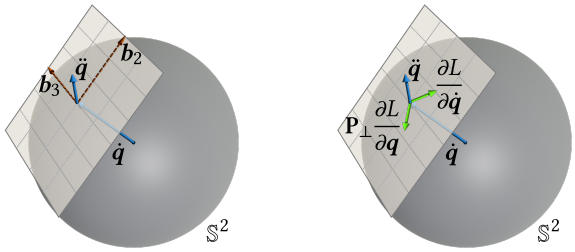

Fig. 5. **Solving the Euler-Lagrange equation with unit tangent vectors** of the curve $s \mapsto \boldsymbol{q}(s)$, $\|\dot{\boldsymbol{q}}\| = 1$, which are points on the manifold $\mathbb{S}^2$. Thus, $\ddot{\boldsymbol{q}} \in T_{\dot{\boldsymbol{q}}}\mathbb{S}^2$, i.e., embedded in $\mathbb{R}^3$, $\ddot{\boldsymbol{q}} \perp \dot{\boldsymbol{q}}$. We solve the Euler-Lagrange equation for $\ddot{\boldsymbol{q}}$ by further exploiting that $(\partial L/\partial \dot{\boldsymbol{q}}) \perp \dot{\boldsymbol{q}}$, and using $\mathbf{P}_\perp(\partial L/\partial \boldsymbol{q})$.

All candidate points are then filtered using their locally optimal reference frame $\boldsymbol{w}$ and a threshold-based filter for IVD and the Sujudi and Haimes [1995] criterion in the observed field $\boldsymbol{v} - \boldsymbol{w}$.

The remaining points are considered seed points $\boldsymbol{q}(s_0) \in \mathbb{R}^3$. We then refine their positions via gradient descent to decrease the Lagrangian $L(\boldsymbol{q}, \dot{\boldsymbol{q}}, s_0)$ (Eq. 22) using the current field $\boldsymbol{w}$. Since the locally optimal $\boldsymbol{w}$ depends on the position of the seed point, and vice versa, we alternate between optimizing the field $\boldsymbol{w}$ and the position of the seed point, similar to Rautek et al. [2024], until the position of the seed point converges to the refined point with its corresponding optimal reference frame $\boldsymbol{w}$. Both the position and reference frame are used to solve the variational problem. See Appendix C for details.

5.3 Optimal Vortex Cores

We want to compute a *vortex core* as the *optimal path surface* that is a spacetime 2-manifold given by Eq. 12. For the optimization, we will denote $\boldsymbol{q}(s) = c_{t_0}(s)$. In our framework, we compute an optimal curve $s \mapsto \boldsymbol{q}(s)$ at *one* chosen time t_0, but it will nevertheless be optimal over a chosen (pre-)integration time interval $[t_0, T]$.

5.3.1 Phase Space. We define a *Lagrangian* $L(\boldsymbol{q}, \dot{\boldsymbol{q}}, s)$ in the phase space $\mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R}$. The Lagrangian is a scalar cost function

$$\begin{aligned} L\colon \mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R} &\to \mathbb{R}, \\ (\boldsymbol{q}, \dot{\boldsymbol{q}}, s) &\mapsto L(\boldsymbol{q}, \dot{\boldsymbol{q}}, s). \end{aligned} \tag{20}$$

Here, $\boldsymbol{q} \in \mathbb{R}^3$ corresponds to a spatial point on a core line $s \mapsto \boldsymbol{q}(s)$ at time t_0, and $\dot{\boldsymbol{q}} \in \mathbb{S}^2$ corresponds to a unit tangent at that point. To facilitate the latter, we will work with an arc length parameter s.

An optimal vortex core is then defined as the *minimizer* of

$$S[\boldsymbol{q}(s)] = \int_{s_0}^{s_{max}} L(\boldsymbol{q}(s), \dot{\boldsymbol{q}}(s), s)\, ds. \tag{21}$$

To simplify computing the minimizer $s \mapsto \boldsymbol{q}(s)$, we will partially work in local spherical coordinates $(1, \theta, \phi)$ relative to each unit tangent $\dot{\boldsymbol{q}}$, in particular for computing derivatives on $\mathbb{S}^2$, and transform to global Cartesian coordinates (x, y, z) on the fly. See Fig. 5.

5.3.2 Lagrangian. Since the Lagrangian L is defined for all spatial points $\boldsymbol{q}$ and all possible unit tangent vectors $\dot{\boldsymbol{q}}$, and a vortex core manifold must be a curve advected by the flow, we observe that it is not necessary to solve 2-manifold Euler-Lagrange equations with independent parameters (s, t). Since we specifically want to minimize the integral of an energy term E over a *path surface* (Eq. 12), we can define a Lagrangian with only one independent parameter s:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}, s) = \int_{t_0}^{T} E\Big(\phi_{t_0,t}(\boldsymbol{q}(s)), \phi_{t_0,t*}(\dot{\boldsymbol{q}}(s)), t\Big)\, dt. \tag{22}$$

The energy E does not explicitly depend on the (arbitrary) curve parameter s, which is therefore also true for the Lagrangian L. However, for unsteady flow fields $\boldsymbol{v}$ the energy E does depend explicitly on time t, although the Lagrangian L of course does not. We note that Eq. 22 weights the energy E over each curve $s \mapsto c_t(s)$ equally as for the curve $s \mapsto c_{t_0}(s)$, with $S[\boldsymbol{q}(s)]$ ultimately including the arc length of the optimized curve $s \mapsto c_{t_0}(s)$, independent of whether a core line stretches or contracts over time. If this equal weighting were not desired, the $E(\cdot, \cdot, t)$ term would have to be multiplied by the "area" element of a spacetime surface, which here is $\left\|\phi_{t_0,t*}(\dot{\boldsymbol{q}}(s))\right\|$.

5.3.3 Energy. In order to trace vortex cores $s \mapsto c_{t_0}(s)$, t_0 fixed, we define the energy E determining the Lagrangian L via integration using Eq. 22 by (with relative optimization weights $\mu_c, \mu_d, \mu_r \in \mathbb{R}$)

$$\begin{aligned} E(\boldsymbol{c}, \dot{\boldsymbol{c}}, t) = \|\tilde{\boldsymbol{u}}(\boldsymbol{c}, \dot{\boldsymbol{c}}, t)\|^2 + \mu_c\, C(\boldsymbol{c}, \dot{\boldsymbol{c}}, t) + \\ \mu_d\, \|\boldsymbol{D}(\boldsymbol{c}, t)\|^2 \quad + \mu_r\, \|\boldsymbol{R}(\boldsymbol{c}, \dot{\boldsymbol{c}}, t)\|^2 . \end{aligned} \tag{23}$$

We define the individual energy terms (cf. Eqs. 6, 17, 18)

$$\begin{aligned} \tilde{\boldsymbol{u}}(\boldsymbol{c}, \dot{\boldsymbol{c}}, t) &= (\boldsymbol{v} - \boldsymbol{w})(\boldsymbol{c}, t) - \Big\langle (\boldsymbol{v} - \boldsymbol{w})(\boldsymbol{c}, t), \dot{\boldsymbol{c}} \Big\rangle \frac{\dot{\boldsymbol{c}}}{\|\dot{\boldsymbol{c}}\|^2}, \\ C(\boldsymbol{c}, \dot{\boldsymbol{c}}, t) &= \textit{generic objective scalar vortex criterion to minimize}, \\ \boldsymbol{D}(\boldsymbol{c}, t) &= \frac{D}{Dt}(\boldsymbol{v} - \boldsymbol{w})(\boldsymbol{c}, t), \\ \boldsymbol{R}(\boldsymbol{c}, \dot{\boldsymbol{c}}, t) &= \boldsymbol{e}_3(\nabla(\boldsymbol{v} - \boldsymbol{w})(\boldsymbol{c}, t)) - \Big\langle \boldsymbol{e}_3(\nabla(\boldsymbol{v} - \boldsymbol{w})(\boldsymbol{c}, t)), \dot{\boldsymbol{c}} \Big\rangle \frac{\dot{\boldsymbol{c}}}{\|\dot{\boldsymbol{c}}\|^2}. \end{aligned} \tag{24}$$

The term C is any objective scalar vortex criterion to be minimized, such as −IVD (the negation of instantaneous vorticity deviation) at the point $\boldsymbol{c} \in \mathbb{R}^3$. In fact, if $C(\boldsymbol{c}, \dot{\boldsymbol{c}}, t) = -\text{IVD}(\boldsymbol{c}, t)$, Eq. 22 will automatically compute −LAVD [Haller et al. 2016]. We note that using $C = -\|\nabla \times (\boldsymbol{v} - \boldsymbol{w})\|$ is similar and also objective. For terms C that depend on $\dot{\boldsymbol{c}}$, only $\dot{\boldsymbol{c}}/\|\dot{\boldsymbol{c}}\|$ must be used. The term $\boldsymbol{R}$ is used for regularization, to make sure there is a known preferred tangent direction (the real eigenvector $\boldsymbol{e}_3$) for the vortex line in case $\boldsymbol{v} - \boldsymbol{w} = 0$.

5.3.4 Solving the Euler-Lagrange Equation. To solve for the curve $s \mapsto \boldsymbol{q}(s)$, we obtain the minimizer of the action in Eq. 21 by solving

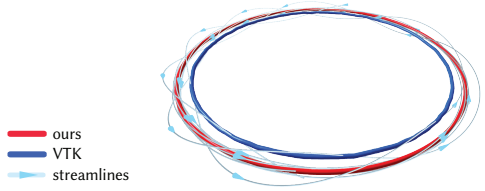

Fig. 6. **Vortex ring** core line extraction in an unsteady field, created from an analytical model of a steady, but spinning vortex ring (Roth and Peikert [1996]) observed in a moving reference frame. Blue core line: VTK [Schroeder et al. 2006]. Red core line: Our result shows close alignment with the swirling streamlines visualized in the moving reference frame.

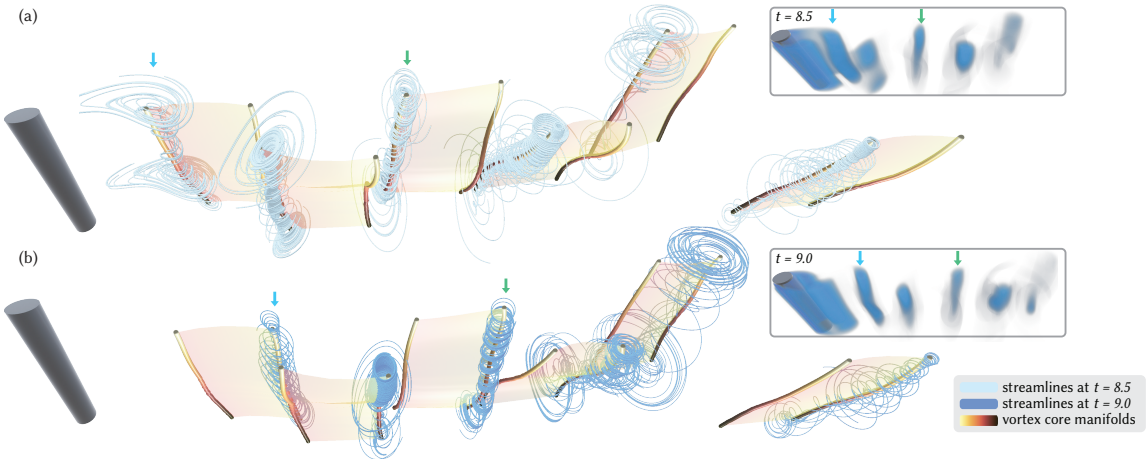

Fig. 7. **Spacetime surfaces of vortex cores in a von Kármán vortex street** behind an obstacle (CFD simulation) provide a comprehensive, continuous visualization of vortex shedding. Where conventional volume visualizations of the IVD field (insets) are often fuzzy and ambiguous, our method extracts precise geometric manifolds. Streamlines rendered in the co-moving frame confirm the tight alignment of our vortex cores and instantaneous fluid flow patterns. The blue and green arrows highlight two vortices and their corresponding streamlines at time (a) $t = 8.5$, and (b) $t = 9$, respectively.

the corresponding Euler-Lagrange equation with arc length parameterization constraint enforcing $\|\dot{\boldsymbol{q}}\| = 1$, which is (see Appendix A)

$$\frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{ds}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{d}{ds}\left(2\lambda \dot{\boldsymbol{q}}\right) = 0. \tag{25}$$

Equivalent to using a Lagrange multiplier with corresponding derivative $(d/ds)\,\lambda(s) = (1/2)\,(\partial L/\partial \boldsymbol{q}) \cdot \dot{\boldsymbol{q}}$, we can write Eq. 25 as

$$\mathbf{P}_{\perp}\frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{ds}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - 2\lambda \ddot{\boldsymbol{q}} = 0. \tag{26}$$

The operator $\mathbf{P}_{\perp} = (\mathbf{I} - \dot{\boldsymbol{q}}\dot{\boldsymbol{q}}^T)$ projects into the plane perpendicular to the unit tangent $\dot{\boldsymbol{q}}$. As derived in Appendix A, because our Lagrangian L is independent of $\|\dot{\boldsymbol{q}}\|$, we can solve Eq. 26 in the plane perpendicular to $\dot{\boldsymbol{q}}$ (Fig. 5). At parameter position s, with $(\boldsymbol{q}, \dot{\boldsymbol{q}})(s)$, we first construct a right-handed orthonormal frame $\left(\dot{\boldsymbol{q}}|\boldsymbol{b}_2|\boldsymbol{b}_3\right)$, and then solve for $\ddot{\boldsymbol{q}}(s)$ at that position by solving Eq. 26 explicitly as

$$\ddot{\boldsymbol{q}}(s) = \left[\left(\boldsymbol{b}_2|\boldsymbol{b}_3\right) \left(\bar{\mathbf{H}} + 2\lambda(s)\,\mathbf{I}^{2\times 2}\right)^{-1} \left(\boldsymbol{b}_2|\boldsymbol{b}_3\right)^T \frac{\partial L}{\partial \boldsymbol{q}} \right]_{(\boldsymbol{q},\dot{\boldsymbol{q}})(s)}. \tag{27}$$

We compute the curve $s \mapsto \boldsymbol{q}(s)$ by integrating this second-order ODE using a fourth-order Runge-Kutta (RK4) integrator. All partial derivatives in the 2×2 Hessian matrix $\bar{\mathbf{H}}$ are evaluated at $\dot{\boldsymbol{q}}$ in spherical coordinates (Appendix A). The scalar Lagrange multiplier function $\lambda(s)$ provides additional (optional) regularization if $\lambda > 0$.

6 Results and Evaluation

Our variational framework is implemented in C++ with an OpenGL-based renderer. Detailed implementation notes, pseudocode, and performance analysis are given in Appendix C.

Fig. 1 provides an empirical validation of our framework using the Delta Wing dataset, a scenario characterized by strong shear and vortex breakdown. Eulerian frame-by-frame extraction methods fail to ensure material coherence and suffer from topological instabilities. In contrast, our result demonstrates a drift-free spacetime surface that preserves topology. By solving the Euler-Lagrange equations for the pre-integrated energy Lagrangian, we ensure that the extracted spacetime manifold describes the same physical set of particles throughout the time interval $[0, 0.3]$. The strict alignment between the spacetime surface (red-yellow), the instantaneous streamlines (visualized in a co-moving reference frame), and the instantaneous vorticity deviation (IVD) volume, confirms that our method successfully identifies a feature with high geometric fidelity and persistence, a result that was previously not possible without computationally expensive 4D global optimization methods.

Fig. 2 evaluates the spatial quality of the extraction in a steady vector field using the SteadyTornado3D dataset. Our framework unifies steady and unsteady flows without special case treatment, which is not only important for steady fields but also for fields with restricted forms of unsteadiness. In contrast, the recent method of Günther and Theisel [2025] is not well-suited for steady flow fields, because reference frames are forced to follow path lines of the input field, which, as in the case of the steady tornado dataset, can prevent successful core line extraction. Moreover, standard local extraction algorithms, such as the Parallel Vectors (PV) operator implemented in the Visualization Toolkit, VTK [Schroeder et al. 2006], process grid cells independently. This often leads to discretization artifacts, resulting in jagged vortex core polylines (shown in blue) that require aggressive post-processing to be visually acceptable. Even with pre-smoothing of the data and post-smoothing of the geometry, the VTK result fails to capture the continuous nature of the vortex core line. In contrast, our variational framework treats the core line as a continuous curve minimizing a global energy functional. This formulation inherently acts as a geometric regularizer. By solving for the energy minimum, our method produces a smooth, differentiable curve (shown in red) that naturally resists grid-induced noise without the need for ad-hoc filtering parameters. The zoom-in confirms that this smoothness does not come at the cost of accuracy. The core line remains perfectly centered within the tightly swirling streamlines (shown in white), offering a precise and smooth result.

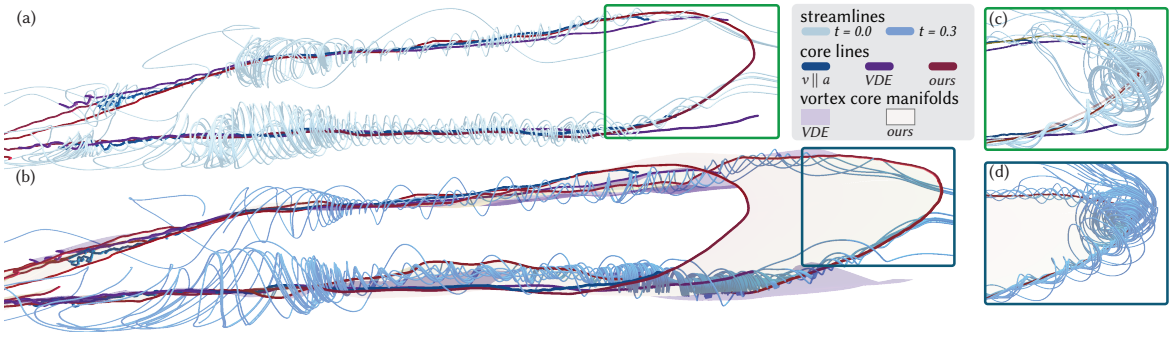

Fig. 8. **Baseline comparison** of VTK/parallel vectors ($v \parallel a$), VDE [Günther and Theisel 2025], and our method, showing results on two time steps of the Delta Wing CFD simulation. (a) $t = 0$ shows the first time step and the extracted core line results from the three different methods together with streamlines in the co-moving reference frame. (b) $t = 0.3$ shows how the spacetime surfaces of the core lines evolve computed with VDE [Günther and Theisel 2025] and our method. Only our method extracts the core line at the tip of the vortex. We verify that at the tip of the vortex a swirling motion is present by transforming the streamlines into a reference frame that moves at the same speed as the tip of the vortex. The insets (c) and (d) show the highlighted regions with streamlines closely swirling around the vortex core line. This confirms that our method is able to correctly extract the full core line over the entire time interval.

The analytical vortex ring shown in Fig. 6 serves as a ground truth benchmark for the comparison of curved core lines (Roth and Peikert [1996]) extracted with our approach (red) and standard techniques available in tools like VTK (blue). Standard local methods typically rely on the parallel vectors operator, which assumes locally straight vortex tubes. This assumption holds poorly for bending core lines, causing the extracted geometry (blue) to systematically deviate from the physical center of rotation due to the curvature of the ring. In contrast, our variational method (red) minimizes a global energy functional, making it robust to high curvature. It accurately recovers the exact center of the ring, aligning perfectly with the streamlines in the co-moving reference frame.

In Fig. 7 we demonstrate the capability of our method to capture and visualize the lifecycle of vortices in a vortex street with the extracted spacetime surfaces. We show the results of our method on a CFD simulation of a von Kármán vortex street behind a half-cylinder obstacle. Unlike traditional volume rendering (of the IVD field in the insets of Fig. 7), which suffers from ambiguities and fuzzy results, our method extracts a precise geometric representation of the vortices as continuous spacetime surfaces. Our spacetime manifolds allow for the simultaneous visualization of vortex formation and advection. The Lagrangian consistency of the extracted geometric structures is validated by the tight alignment of the core lines with the instantaneous streamlines visualized in the co-moving reference frame. The vortices marked by the blue and green arrows highlight corresponding vortex structures at different times. This confirms that our pre-integrated variational formulation correctly and coherently over time identifies the vortex structures and highlights the benefit of our method over direct volume visualization methods.

In Fig. 8, we evaluate the topological completeness of our core line extraction against two baseline methods: A standard local Parallel Vectors (PV) operator (via VTK) and the recent state-of-the-art method by Günther and Theisel [2025] based on the Vortex Deviation Error (VDE). The Delta Wing dataset presents a challenging scenario where the vortex core exhibits spatially varying convection velocities. For instance, the tip of the vortex travels downstream at a different speed than the stable leading edge. As shown in Fig. 8 (a), the local PV method yields a fragmented and truncated core line, failing to maintain connectivity due to its sensitivity to noise. While VDE produces a smoother result, our method extracts a significantly longer and more complete core line, and is able to successfully capture the vortex tip throughout the evolution time interval (shown in Fig. 8 (b) at $t = 0.3$). Validating the existence of the vortex at this tip requires careful treatment of the reference frame. Because the tip moves at a different velocity than the main body, a single global reference frame cannot reveal the swirling motion along the entire length of the core simultaneously. In the global co-moving frame (main view), the streamlines at the tip appear laminar due to the relative velocity difference. However, by shifting the observer to the local reference frame of the tip (shown in Figs. 8 (c) and (d)) we reveal the tight swirling motion of the streamlines around our extracted geometry. This confirms that our method correctly recovers the full extent of the vortex core, including the tip of the vortex core that is missed by previous methods.

To demonstrate the applicability of our framework to complex engineering scenarios, Fig. 9 presents results from a high-fidelity CFD simulation of an F22 Raptor airplane. This dataset is characterized by intricate boundary geometry and rapidly evolving vortex dynamics. Our method successfully extracts the coherent spacetime manifolds

Table 1. **Streamline winding numbers** (averaged over five time steps; higher is better) around vortex cores verify the local rotational behavior. Our method closely matches the ground truth. A "-" indicates either steady flow (when VDE is often not applicable) or where the ground truth is not known.

Dataset	Fig.	VTK	VDE	Ours	Ground Truth
DeltaWing3D	1, 8	0.59	2.13	**2.92**	-
SteadyTornado3D	2	**2.96**	-	**2.96**	2.96
VortexRing	6	1.87	**1.88**	**1.88**	1.88
Cylinder3D	7	1.99	**2.25**	2.23	-
TrefoilKnot	10	19.86	20.02	**21.01**	-
FourCenters3D	11	6.34	**7.22**	**7.22**	7.22

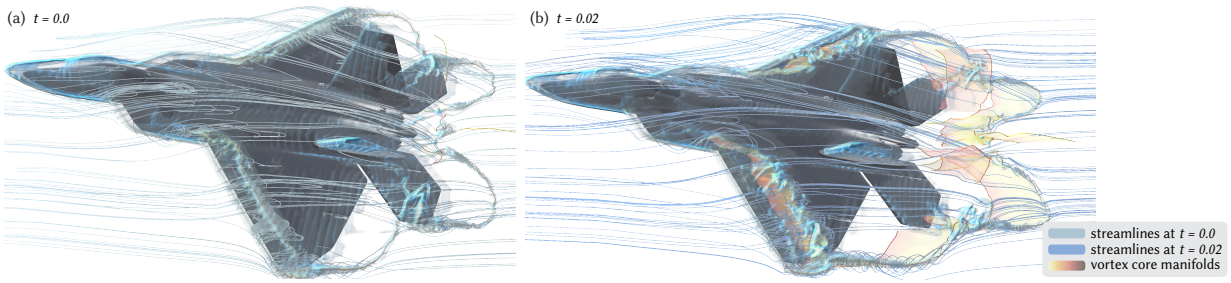

Fig. 9. **F22 Raptor vortex cores.** We show the spacetime surfaces resulting from our core line extraction at time $t = 0$ (left) and at time $t = 0.02$ (right) in the rapidly evolving CFD simulation of the airflow around an F22 plane. For animation sequences of the results please see the supplementary video.

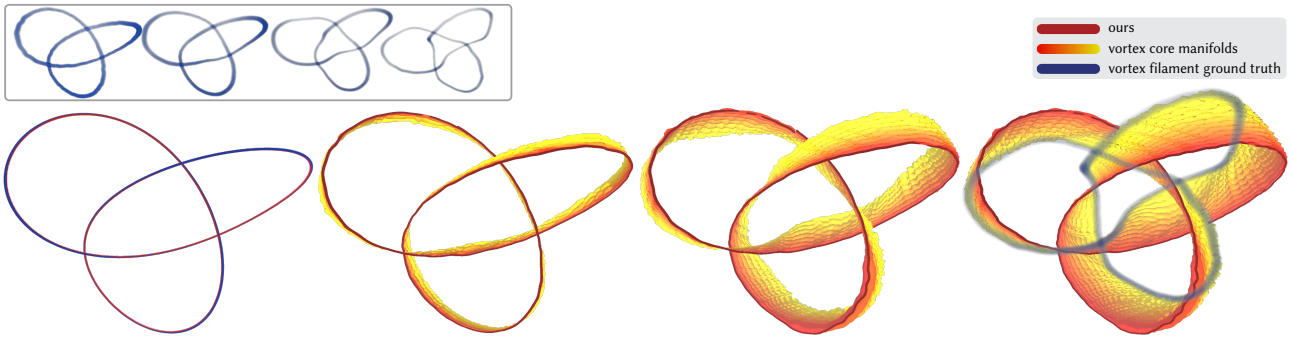

Fig. 10. **A vortex filament** (trefoil knot) simulated with the covector fluids code of Nabizadeh et al. [2022]. We show four time steps: Top-left inset: IVD volume renderings. Bottom row: The first image compares the filament ground truth (blue) with the result of our method (red) for this time step. Our pre-integration approach is able to follow the time evolution of the filament as long as its topology does not change. (See Nabizadeh et al. [2022] for topology changes.)

of the primary vortex cores, visualized here at $t = 0$ and $t = 0.02$. The smoothness and continuity of the extracted surfaces confirm that our variational formulation maintains topological stability even in the presence of highly unsteady flow features. Unlike frame-by-frame extraction, which often suffers from temporal flickering or fragmentation in such dynamic regimes, our spacetime optimization ensures a consistent representation of the vortex evolution.

Fig. 10 demonstrates that our method is also able to track vortex filaments with complicated geometry over time. The shown trefoil knot was simulated using the covector fluids code by Nabizadeh et al. [2022]. It evolves significantly over time, but as long as its topology does not change, our method is able to track its time evolution successfully. The first image shows that for the first time step we can faithfully trace the ground truth geometry of the filament.

Fig. 11 (Appendix B) shows that our framework can resolve multiple interacting coherent structures within a globally rotating system.

Streamline Winding Number Metric. To quantitatively evaluate the geometric coherence and rotational stability of extracted corelines, we introduce a *streamline winding number* metric. Given a discrete curve (with potentially varying sampling densities), we process each point independently. A local tangent vector is estimated via finite differences, from which we construct a local orthonormal frame and define a projection plane perpendicular to the tangent. On this plane, we sample eight neighboring seed points uniformly on a circle of radius 0.01 centered at the point. From these seeds, we integrate eight streamlines of the observed vector field using RK4 with step size 0.005 for 100 steps. The resulting trajectories are projected back onto the plane, and we compute the angular displacement of each streamline around the center. The absolute values of these angular changes are accumulated to measure local rotational strength.

Because the accumulated rotation scales with the number of discrete samples along the curve, we normalize by the number of points to obtain a discretization-invariant average winding number. Extending this to the spatiotemporal setting, we aggregate the winding numbers of the five core lines across five consecutive time steps, and normalize by the total number of points over all curves. The resulting averaged value is reported in Table 1.

For VTK, corelines are independently extracted at each time step and evaluated separately. For Ours and VDE, we compute an optimal initial curve and advect it to subsequent time steps using the flow map, then do the evaluation. For datasets with ground-truth core lines, we additionally report their winding numbers, demonstrating that our method closely matches the reference.

7 Discussion and Limitations

Dependence on Killing field w. A crucial component of our generic formulation is the geometric modeling of a vortex core with an explicit objective deformation vector field together with an underlying Killing field $\boldsymbol{w}$ modeling a Euclidean reference frame. We determine this reference frame through optimization before,

and independently of, the solution of the Euler-Lagrange equations, and before time-dependent energy functions are pre-integrated into the time-independent Lagrangian L. For this reason, the Killing field $\boldsymbol{w}$ must be well-chosen for any potential vortex core, which we approach as described in Sec. 5.2. However, this dependence on a specific Killing field $\boldsymbol{w}$ before an actual vortex core is computed variationally is currently a limitation of our method.

Objectivity. Our framework is objective by construction, provided the generic vortex criterion chosen for the term $\boldsymbol{C}$ in Eq. 24 is also objective. The main reason for this is that computing an observer field $\boldsymbol{w}$ using Eq. 19 makes all terms in Eq. 24 objective.

Topology changes. Our method currently cannot handle changing core line topology, such as bifurcations and merging. See the examples by Nabizadeh et al. [2022]. These can locally violate Lagrangian (material) coherence, terminating our Euler–Lagrange integration of the core line. In practice, this would usually also mean that the region $U(t)$ cannot be made steady enough (with small enough observed time derivative) by a single observer field $\boldsymbol{w}$.

Numerical accuracy. Regarding the numerical properties of our method, please see the detailed analysis in Appendix C.3.

8 Conclusions

We have presented a generic variational framework for vortex core optimization as spacetime manifolds. For high computational efficiency, our framework exploits the key insight that solving the Euler-Lagrange equations using temporally pre-integrated energy terms in fact behaves as though all possible core lines would be advected by the flow, although the actual core line is not known yet. This fact results from the combination of properties of the phase space for optimization with the Lagrangian nature of vortex cores.

Acknowledgments

We would like to thank Anna Hadwiger for illustrations and help with the figures, and Dominik Michels for help with the submission upload. This work was supported by King Abdullah University of Science and Technology (KAUST) baseline funding.

References

Irene Baeza Rojo and Tobias Günther. 2020. Vector Field Topology of Time-Dependent Flows in a Steady Reference Frame. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2019)* 26, 1 (Jan. 2020), 280–290. doi:10.1109/TVCG.2019.2934375

Roxana Bujack, Lin Yan, Ingrid Hotz, Christoph Garth, and Bei Wang. 2020. State of the Art in Time-Dependent Flow Topology: Interpreting Physical Meaningfulness Through Mathematical Properties. *Computer Graphics Forum* 39, 3 (June 2020), 811–835. doi:10.1111/cgf.14037

Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. 2017. Inside Fluids: Clebsch Maps for Visualization and Processing. *ACM Trans. Graph.* 36, 4, Article 142 (July 2017), 11 pages. doi:10.1145/3072959.3073591

Albert Chern, Felix Knöppel, Ulrich Pinkall, Peter Schröder, and Steffen Weißmann. 2016. Schrödinger's Smoke. *ACM Trans. Graph.* 35, 4, Article 77 (July 2016), 13 pages. doi:10.1145/2897824.2925868

Min S. Chong, Anthony E. Perry, and Brian J. Cantwell. 1990. A General Classification of Three-Dimensional Flow Fields. *Physics of Fluids* 2 (1990), 765–777. https://api.semanticscholar.org/CorpusID:120023145

Nico Daßler and Tobias Günther. 2024. Variational Feature Extraction in Scientific Visualization. *ACM Trans. Graph.* 43, 4, Article 109 (July 2024), 16 pages. doi:10.1145/3658219

Tobias Günther and Irene Baeza Rojo. 2021. Introduction to Vector Field Topology. In *Topological Methods in Data Analysis and Visualization VI*. Springer International Publishing, Cham, 289–326. doi:10.1007/978-3-030-83500-2_15

Tobias Günther, Markus Gross, and Holger Theisel. 2017. Generic Objective Vortices for Flow Visualization. *ACM Transactions on Graphics* 36, 4 (July 2017), 141:1–141:11. doi:10.1145/3072959.3073684

Tobias Günther and Holger Theisel. 2016. Rotation Invariant Vortices for Flow Visualization. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2015)* 22, 1 (Jan. 2016), 817–826. doi:10.1109/TVCG.2015.2467200

Tobias Günther and Holger Theisel. 2018. The State of the Art in Vortex Extraction. *Computer Graphics Forum* 37, 6 (2018), 149–173. doi:10.1111/cgf.13319

Tobias Günther and Holger Theisel. 2019. Objective Vortex Corelines of Finite-sized Objects in Fluid Flows. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2018)* 25, 1 (Jan. 2019), 956–966. doi:10.1109/TVCG.2018.2864828

Tobias Günther and Holger Theisel. 2020. Hyper-Objective Vortices. *IEEE Transactions on Visualization and Computer Graphics* 26, 3 (March 2020), 1532–1547. doi:10.1109/TVCG.2018.2868760

Tobias Günther and Holger Theisel. 2025. Objective Lagrangian Vortex Cores and their Visual Representations. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2024)* 31, 1 (Jan. 2025), 76–85. doi:10.1109/TVCG.2024.3456384

Markus Hadwiger, Matej Mlejnek, Thomas Theußl, and Peter Rautek. 2019. Time-Dependent Flow Seen Through Approximate Observer Killing Fields. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2018)* 25, 1 (Jan. 2019), 1257–1266. doi:10.1109/TVCG.2018.2864839

George Haller. 2002. Lagrangian Coherent Structures from Approximate Velocity Data. *Physics of Fluids* 14, 6 (06 2002), 1851–1861. doi:10.1063/1.1477449

George Haller. 2005. An Objective Definition of a Vortex. *Journal of Fluid Mechanics* 525 (2005), 1–26. doi:10.1017/S0022112004002526

George Haller. 2015. Lagrangian Coherent Structures. *Annual Review of Fluid Mechanics* 47 (2015), 137–162. doi:10.1146/annurev-fluid-010313-141322

George Haller. 2021. Can Vortex Criteria be Objectivized? *Journal of Fluid Mechanics* 508 (2021), A25. doi:10.1017/jfm.2020.937

George Haller, Alireza Hadjighasem, Mohammad Farazmand, and Florian Huhn. 2016. Defining Coherent Vortices Objectively from the Vorticity. *Journal of Fluid Mechanics* 795 (2016), 136–173. doi:10.1017/jfm.2016.151

Julian CR Hunt, Alan A Wray, and Parviz Moin. 1988. Eddies, Streams, and Convergence Zones in Turbulent Flows. In *Studying turbulence using numerical simulation databases, 2. Proceedings of the 1988 summer program*. Center for Turbulence Research, Stanford, CA, USA, 193–208.

Sadashige Ishida, Chris Wojtan, and Albert Chern. 2022. Hidden Degrees of Freedom in Implicit Vortex Filaments. *ACM Trans. Graph.* 41, 6, Article 241 (Nov. 2022), 14 pages. doi:10.1145/3550454.3555459

Jinhee Jeong and Fazle Hussain. 1995. On the Identification of a Vortex. *Journal of Fluid Mechanics* 285 (1995), 69–94. doi:10.1017/S0022112095000462

Bálint Kaszás, Tiemo Pedergnana, and George Haller. 2023. The Objective Deformation Component of a Velocity Field. *European Journal of Mechanics-B/Fluids* 98 (2023), 211–223. doi:10.1016/j.euromechflu.2022.12.007

Byungsoo Kim and Tobias Günther. 2019. Robust Reference Frame Extraction from Unsteady 2D Vector Fields with Convolutional Neural Networks. *Computer Graphics Forum (Proc. Eurovis 2019)* 38, 3 (2019), 285–295. doi:10.1111/cgf.13689

Hans J. Lugt. 1979. The Dilemma of Defining a Vortex. In *Recent Developments in Theoretical and Experimental Fluid Mechanics*. Springer, Berlin, Heidelberg, 309–321. doi:10.1007/978-3-642-67220-0_32

Mohammad Sina Nabizadeh, Stephanie Wang, Ravi Ramamoorthi, and Albert Chern. 2022. Covector Fluids. *ACM Trans. Graph.* 41, 4, Article 113 (July 2022), 16 pages. doi:10.1145/3528223.3530120

Marcel Padilla, Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. 2019. On Bubble Rings and Ink Chandeliers. *ACM Trans. Graph.* 38, 4, Article 129 (July 2019), 14 pages. doi:10.1145/3306346.3322962

Peter Rautek, Matej Mlejnek, Johanna Beyer, Jakob Troidl, Hanspeter Pfister, Thomas Theußl, and Markus Hadwiger. 2021. Objective Observer-Relative Flow Visualization in Curved Spaces for Unsteady 2D Geophysical Flows. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2020)* 27, 2 (Feb. 2021), 283–293. doi:10.1109/TVCG.2020.3030454

Peter Rautek, Xingdi Zhang, Bernhard Woschizka, Thomas Theußl, and Markus Hadwiger. 2024. Vortex Lens: Interactive Vortex Core Line Extraction Using Observed Line Integral Convolution. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2023)* 30, 1 (Jan. 2024), 55–65. doi:10.1109/TVCG.2023.3326915

Martin Roth and Ronald Peikert. 1996. Flow Visualization for Turbomachinery Design. In *Proceedings of IEEE Visualization 1996*. IEEE, San Francisco, CA, USA, 381–384. doi:10.1109/VISUAL.1996.568137

Martin Roth and Ronald Peikert. 1998. A Higher-Order Method for Finding Vortex Core Lines. In *Proceedings of IEEE Visualization 1998*. IEEE, Research Triangle Park, North Carolina, USA, 143–150. doi:10.1109/visual.1998.745296

Martin Roth and Ronald Peikert. 1999. The "Parallel Vectors" Operator–A Vector Field Visualization Primitive. In *Proceedings of IEEE Visualization 1999*. IEEE, San Francisco, CA, USA, 263–532. doi:10.1109/visual.1999.809896

Philip G. Saffman. 1992. *Vortex Dynamics*. Cambridge University Press.

Will Schroeder, Ken Martin, and Bill Lorensen. 2006. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics* (4th ed.). Kitware, Upper Saddle River, NJ, United States.

Shawn C. Shadden, Francois Lekien, and Jerrold E. Marsden. 2005. Definition and Properties of Lagrangian Coherent Structures from Finite-Time Lyapunov Exponents in Two-Dimensional Aperiodic Flows. *Physica D: Nonlinear Phenomena* 212, 3 (2005), 271–304. doi:10.1016/j.physd.2005.10.007

David Sujudi and Robert Haimes. 1995. Identification of Swirling Flow in 3-D Vector Fields. In *Proceedings of the 12th Computational Fluid Dynamics Conference*. AIP Publishing, Kobe, JAPAN, 792–799. doi:10.2514/6.1995-1715

Holger Theisel, Markus Hadwiger, Peter Rautek, Thomas Theußl, and Tobias Günther. 2021. Vortex Criteria can be Objectivized by Unsteadiness Minimization. *Physics of Fluids* 33 (2021), 107115. Issue 10. doi:10.1063/5.0063817

Steffen Weißmann and Ulrich Pinkall. 2010. Filament-Based Smoke with Vortex Shedding and Variational Reconnection. *ACM Trans. Graph.* 29, 4, Article 115 (July 2010), 12 pages. doi:10.1145/1778765.1778852

Steffen Weißmann, Ulrich Pinkall, and Peter Schröder. 2014. Smoke Rings from Smoke. *ACM Trans. Graph.* 33, 4, Article 140 (July 2014), 8 pages. doi:10.1145/2601097.2601171

Hang Yin, Mohammad Sina Nabizadeh, Baichuan Wu, Stephanie Wang, and Albert Chern. 2023. Fluid Cohomology. *ACM Trans. Graph.* 42, 4, Article 126 (July 2023), 25 pages. doi:10.1145/3592402

Xingdi Zhang, Amani Ageeli, Thomas Theußl, Markus Hadwiger, and Peter Rautek. 2026. Exploring 3D Unsteady Flow using 6D Observer Space Interactions. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2025)* 32, 1 (Jan. 2026), 517–526. doi:10.1109/TVCG.2025.3642506

Xingdi Zhang, Markus Hadwiger, Thomas Theußl, and Peter Rautek. 2022. Interactive Exploration of Physically-Observable Objective Vortices in Unsteady 2D Flow. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE VIS 2021)* 28, 1 (Jan. 2022), 281–290. doi:10.1109/TVCG.2021.3115565

Xingdi Zhang, Peter Rautek, and Markus Hadwiger. 2025a. VortexTransformer: End-to-End Objective Vortex Detection in 2D Unsteady Flow Using Transformers. *Computer Graphics Forum (Proc. Eurographics 2025)* 44, 2 (2025), e70042. doi:10.1111/cgf.70042

Xingdi Zhang, Peter Rautek, Thomas Theußl, and Markus Hadwiger. 2025b. Enhancing Material Boundary Visualizations in 2D Unsteady Flow through Local Reference Frame Transformations. *Computer Graphics Forum (Proc. Eurovis 2025)* 44, 3 (2025), e70128. doi:10.1111/cgf.70128

J. Zhou, R. J. Adrian, S. Balachandar, and T. M. Kendall. 1999. Mechanisms for Generating Coherent Packets of Hairpin Vortices in Channel Flow. *Journal of Fluid Mechanics* 387 (1999), 353–396. doi:10.1017/S002211209900467X

Cuncheng Zhu, Hang Yin, and Albert Chern. 2025. Viscous Vortex Dynamics on Surfaces. *ACM Trans. Graph.* 44, 6, Article 178 (Dec. 2025), 21 pages. doi:10.1145/3763320

A Solving the Euler-Lagrange Equation

We first summarize how we solve the Euler-Lagrange equation in our framework, and then provide the corresponding derivations.

A.1 Summary of Recipe

We solve the Euler-Lagrange equation with arc length parameterization constraint at the current point $(\boldsymbol{q}, \dot{\boldsymbol{q}})(s)$ by

$$\begin{aligned} \ddot{\boldsymbol{q}}(s) &= \begin{pmatrix} \ddot{q}^1 \\ \ddot{q}^2 \\ \ddot{q}^3 \end{pmatrix}(s) = \left(\boldsymbol{b}_2 | \boldsymbol{b}_3\right) \bar{\ddot{\boldsymbol{q}}}, \\ \bar{\ddot{\boldsymbol{q}}} &= \bar{\mathbf{H}}^{-1}\, \bar{\boldsymbol{r}}, \\ \bar{\mathbf{H}} &= \begin{pmatrix} \frac{\partial^2 L}{\partial\theta\partial\theta} & -\frac{\partial^2 L}{\partial\theta\partial\phi} \\ -\frac{\partial^2 L}{\partial\phi\partial\theta} & \frac{\partial^2 L}{\partial\phi\partial\phi} \end{pmatrix}\Bigg|_{(\boldsymbol{q},\dot{\boldsymbol{q}})}, \quad \bar{\boldsymbol{r}} = \left(\boldsymbol{b}_2 | \boldsymbol{b}_3\right)^T \frac{\partial L}{\partial \boldsymbol{q}}\Bigg|_{(\boldsymbol{q},\dot{\boldsymbol{q}})}. \end{aligned} \tag{28}$$

All partial derivatives in the 2×2 Hessian matrix $\bar{\mathbf{H}}$ are evaluated at $\dot{\boldsymbol{q}}$ transformed to spherical coordinates $\dot{\boldsymbol{q}} = (r, \theta, \phi) = \left(1, 0, \frac{\pi}{2}\right)$. The basis $\boldsymbol{b}_2, \boldsymbol{b}_3$ is any right-handed choice of orthonormal vectors such that $(\dot{\boldsymbol{q}}, \boldsymbol{b}_2, \boldsymbol{b}_3)$ form a right-handed orthonormal frame. That is,

$$\begin{gathered} \boldsymbol{b}_2 \perp \dot{\boldsymbol{q}},\ \boldsymbol{b}_3 \perp \dot{\boldsymbol{q}},\ \boldsymbol{b}_2 \perp \boldsymbol{b}_3,\ \dot{\boldsymbol{q}} = \boldsymbol{b}_2 \times \boldsymbol{b}_3, \\ \|\boldsymbol{b}_2\| = \|\boldsymbol{b}_3\| = 1. \end{gathered} \tag{29}$$

See Appendix A.3 for the corresponding definition of the spherical coordinate system (r, θ, ϕ) aligned with this choice of basis vectors.

As derived below, we can additionally use a Lagrange multiplier $\lambda > 0$ for regularization, meaning that in Eq. 28, instead of inverting the matrix $\bar{\mathbf{H}}$ given above, we invert the regularized matrix

$$\bar{\mathbf{H}}_{regularized} = \bar{\mathbf{H}} + 2\lambda(s)\, \mathbf{I}^{2\times 2}. \tag{30}$$

A.2 Derivation

Given a scalar Lagrangian function L, without constraints the corresponding Euler-Lagrange equation is given by

$$\frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{ds}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} = 0. \tag{31}$$

In order to enforce an arc length parameterization with $\|\dot{\boldsymbol{q}}\| = 1$, we define the Lagrangian for the problem augmented with a unit tangent constraint, by defining the augmented Lagrangian

$$\mathcal{L} = L + \lambda\left(\dot{\boldsymbol{q}} \cdot \dot{\boldsymbol{q}} - 1\right). \tag{32}$$

In terms of the original Lagrangian L, the augmented Euler-Lagrange equation with the arc length constraint is then given by

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{ds}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{d}{ds}\left(2\lambda\dot{\boldsymbol{q}}\right) &= 0, \\ \frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{ds}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - 2\dot{\lambda}\dot{\boldsymbol{q}} - 2\lambda\ddot{\boldsymbol{q}} &= 0. \end{aligned} \tag{33}$$

Taking the dot product of this equation with $\dot{\boldsymbol{q}}$ gives

$$\begin{aligned} \dot{\boldsymbol{q}} \cdot \left(\frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{ds}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - 2\dot{\lambda}\dot{\boldsymbol{q}} - 2\lambda\ddot{\boldsymbol{q}}\right) &= 0, \\ \dot{\boldsymbol{q}} \cdot \frac{\partial L}{\partial \boldsymbol{q}} - 2\dot{\lambda} - 2\lambda\dot{\boldsymbol{q}} \cdot \ddot{\boldsymbol{q}} &= 0, \\ \dot{\boldsymbol{q}} \cdot \frac{\partial L}{\partial \boldsymbol{q}} - 2\dot{\lambda} &= 0. \end{aligned} \tag{34}$$

The Lagrange multiplier derivative $\dot{\lambda} = (d/ds)\,\lambda(s)$ is therefore

$$\dot{\lambda} = \frac{1}{2}\dot{\boldsymbol{q}} \cdot \frac{\partial L}{\partial \boldsymbol{q}}. \tag{35}$$

Inserting $\dot{\lambda}$ into Eq. 33, we see that the term $\partial L/\partial \boldsymbol{q}$ is projected into the plane perpendicular to the unit tangent $\dot{\boldsymbol{q}}$. We can write this projection by directly using an operator $\mathbf{P}_\perp = (\mathbf{I} - \dot{\boldsymbol{q}}\dot{\boldsymbol{q}}^T)$, which gives the Euler-Lagrange equation with arc length constraint as

$$\mathbf{P}_\perp \frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{ds}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - 2\lambda\ddot{\boldsymbol{q}} = 0. \tag{36}$$

We can re-write this equation in a more explicit form by expanding the total derivative (d/ds) to obtain the equivalent Euler-Lagrange equation, with explicit second-order terms, given by

$$\ddot{q}^i(s) = \mathbf{H}^{-1}_{i,j}\left(\mathbf{P}_\perp \frac{\partial L}{\partial q^j} - \sum_{k=1}^{3} \frac{\partial^2 L}{\partial \dot{q}^j \partial q^k}\dot{q}^k(s) - \frac{\partial^2 L}{\partial s\, \partial \dot{q}^j}\right). \tag{37}$$

Because the Lagrangian in Eq. 22 does not explicitly depend on the curve parameter s, for Eq. 37 we immediately get the simplified

$$\ddot{q}^i(s) = \mathbf{H}^{-1}_{i,j}\left(\mathbf{P}_\perp \frac{\partial L}{\partial q^j} - \sum_{k=1}^{3} \frac{\partial^2 L}{\partial \dot{q}^j \partial q^k}\dot{q}^k(s)\right). \tag{38}$$

The augmented 3×3 Hessian matrix $\mathbf{H}_{i,j}$ in Eqs. 37, 38 is given by

$$\mathbf{H}_{i,j} = \left(\frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j}\right) + 2\lambda(s)\, \mathbf{I}^{3\times 3}. \tag{39}$$

In general, to obtain a unique solution to the Euler-Lagrange equation, this Hessian matrix must be non-singular, corresponding to

$$\det \mathbf{H}_{i,j} \neq 0. \tag{40}$$

From Eq.39, we can see that the Lagrange multiplier $\lambda(s)$ acts as a Tikhonov regularizer of the Hessian matrix, ensuring invertibility if $\lambda > 0$. However, we now consider the "pure" case when $\lambda = 0$. In this case, because we only allow regular curves and use an arc length parameterization, our Lagrangian (Eqs. 22, 23, 24) has the property that it is independent of the length of the vector $\dot{\boldsymbol{q}}$, giving that $(\partial L/\partial \dot{\boldsymbol{q}}) \perp \dot{\boldsymbol{q}}$. Because of this property, the 3×3 Hessian matrix in the phase space $\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}$ will be singular, because of one eigenvector $\boldsymbol{e}_1 = \dot{\boldsymbol{q}}$ with corresponding eigenvalue $\lambda_1 = 0$.

However, we can strictly work in the phase space $\mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R}$, i.e., in the unit tangent bundle (plus time) of the configuration space $\mathbb{R}^3$, instead of the tangent bundle, by enforcing that $\|\dot{\boldsymbol{q}}\| = 1$ via the projection operator $\mathbf{P}_\perp$ defined above. Correspondingly, $\ddot{\boldsymbol{q}} \perp \dot{\boldsymbol{q}}$, and solving the Euler-Lagrange equation only requires that a locally defined 2×2 Hessian matrix, which is orthogonal to $\dot{\boldsymbol{q}}$, is non-singular, in order to obtain a unique solution.

Solution in $\mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R}$. As for any arc length parameterization, because $\dot{\boldsymbol{q}} \in \mathbb{S}^2$, we know that $\ddot{\boldsymbol{q}} \in T_{\dot{\boldsymbol{q}}}\mathbb{S}^2$. That is, embedded in $\mathbb{R}^3$, we know that $\ddot{\boldsymbol{q}} \perp \dot{\boldsymbol{q}}$. Given the current point $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ in phase space, we can enforce and exploit this constraint as follows. At the current point, we change the 3D Cartesian coordinate system to one where $\boldsymbol{q}$ maps to the origin, and the x axis aligns with $\dot{\boldsymbol{q}}$. We choose the x axis, because in our context this will make it easier to use standard spherical coordinates. We define a right-handed orthonormal frame

of this coordinate system as $\mathbf{B} = (\dot{\boldsymbol{q}}|\boldsymbol{b}_2|\boldsymbol{b}_3)$, where the columns are the basis vectors. Because $\ddot{\boldsymbol{q}} \perp \dot{\boldsymbol{q}}$, we can re-write Eq. 38 ($\lambda = 0$) as

$$\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j}\right)(\boldsymbol{b}_2\ \boldsymbol{b}_3)\begin{pmatrix}\ddot{q}^{2'}\\ \ddot{q}^{3'}\end{pmatrix}(s) = \mathbf{P}_\perp \frac{\partial L}{\partial q^i} - \sum_{j=1}^{3}\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial q^j}\right)\dot{q}^j(s). \tag{41}$$

We need to compute the Hessian matrices above such that the constraint $\dot{\boldsymbol{q}} \in \mathbb{S}^2$ is strictly enforced, which can be done by computing the partial derivatives in spherical coordinates and transforming them to Cartesian coordinates using the corresponding coordinate Jacobians. Because $(\partial L/\partial \dot{\boldsymbol{q}}) \perp \dot{\boldsymbol{q}}$, the 3×3 Hessian matrix on the left-hand side of Eq. 41 is singular, with one eigenvector $\boldsymbol{e}_1 = \dot{\boldsymbol{q}}$ and eigenvalue $\lambda_1 = 0$. Furthermore, Eq. 41 contains two unknowns and three equations. Nevertheless, we can solve Eq. 41 exactly, solving for the unknowns $(\ddot{q}^{2'}, \ddot{q}^{3'})$ referred to the local Cartesian coordinate frame at the point $\boldsymbol{q}(s)$, given by the basis vectors $\mathbf{B} = (\dot{\boldsymbol{q}}|\boldsymbol{b}_2|\boldsymbol{b}_3)$, and then transforming to the global Cartesian coordinate system to obtain $\ddot{\boldsymbol{q}}(s)$ from $(\ddot{q}^{2'}, \ddot{q}^{3'})$. See Appendix A.3 for the full derivation.

A.3 Hessian Matrices in the Phase Space $\mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R}$

Given the current $\dot{\boldsymbol{q}}$, and the right-handed orthonormal frame $\mathbf{B} = (\dot{\boldsymbol{q}}|\boldsymbol{b}_2|\boldsymbol{b}_3)$, we define a spherical coordinate system (r, θ, ϕ), using the mathematical convention where $r > 0, \theta \in [0, 2\pi], \phi \in [0, \pi]$, such that, at the point $\dot{\boldsymbol{q}} = (r, \theta, \phi) = (1, 0, \frac{\pi}{2})$, we have the basis

$$\left.\frac{\partial}{\partial r}\right|_{(1,0,\frac{\pi}{2})} = \dot{\boldsymbol{q}}, \quad \left.\frac{\partial}{\partial \theta}\right|_{(1,0,\frac{\pi}{2})} = \boldsymbol{b}_2, \quad \left.\frac{\partial}{\partial \phi}\right|_{(1,0,\frac{\pi}{2})} = -\boldsymbol{b}_3. \tag{42}$$

LHS Hessian. We compute the 3×3 Hessian matrix on the left-hand side of Eq. 41 by first computing, at $\dot{\boldsymbol{q}} = (r, \theta, \phi) = (1, 0, \frac{\pi}{2})$,

$$\begin{aligned}\left(\frac{\partial^2 L}{\partial \dot{q}^{i'} \partial \dot{q}^{j'}}\right) &= \mathbf{J}\begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{\partial^2 L}{\partial\theta\partial\theta} + \cos\phi\sin\phi\frac{\partial L}{\partial\phi} & \frac{\partial^2 L}{\partial\theta\partial\phi} - \frac{\cos\phi}{\sin\phi}\frac{\partial L}{\partial\theta} \\ 0 & \frac{\partial^2 L}{\partial\phi\partial\theta} - \frac{\cos\phi}{\sin\phi}\frac{\partial L}{\partial\theta} & \frac{\partial^2 L}{\partial\phi\partial\phi}\end{pmatrix}\mathbf{J}^{-1}, \\ &= \mathbf{J}\begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{\partial^2 L}{\partial\theta\partial\theta} & \frac{\partial^2 L}{\partial\theta\partial\phi} \\ 0 & \frac{\partial^2 L}{\partial\phi\partial\theta} & \frac{\partial^2 L}{\partial\phi\partial\phi}\end{pmatrix}\mathbf{J}^{-1}, \quad \text{at } (r,\theta,\phi) = \left(1, 0, \frac{\pi}{2}\right).\end{aligned} \tag{43}$$

The five zero entries above result from $\partial L/\partial r = 0$, which results from the Lagrangian L being independent of $\|\dot{\boldsymbol{q}}\|$. From this 3×3 Hessian matrix in local Cartesian coordinates, transformed from spherical coordinates using the coordinate Jacobian matrix $\mathbf{J}$, we then compute the 3×3 Hessian in global Cartesian coordinates by

$$\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j}\right) = \mathbf{B}\left(\frac{\partial^2 L}{\partial \dot{q}^{i'} \partial \dot{q}^{j'}}\right)\mathbf{B}^T. \tag{44}$$

We compute the second-order partial derivatives in Eq. 43 by using second-order accurate finite differences. For neighborhood look-ups at $(1, \tilde{\theta}, \tilde{\phi})$, around $\dot{\boldsymbol{q}} = (1, 0, \frac{\pi}{2})$, with $\tilde{\theta} = 0, \pm\Delta\theta$, and $\tilde{\phi} = \frac{\pi}{2}, \frac{\pi}{2} \pm \Delta\phi$, respectively, the spherical coordinates $(1, \tilde{\theta}, \tilde{\phi})$ are converted to global Cartesian coordinates to compute L via Eq. 22 by

$$\tilde{\dot{\boldsymbol{q}}} = \begin{pmatrix}\tilde{\dot{x}}\\ \tilde{\dot{y}}\\ \tilde{\dot{z}}\end{pmatrix} = \mathbf{B}\begin{pmatrix}\cos\tilde{\theta}\sin\tilde{\phi}\\ \sin\tilde{\theta}\sin\tilde{\phi}\\ \cos\tilde{\phi}\end{pmatrix}. \tag{45}$$

We note that for convenience we compute derivatives around $\theta = 0$, instead of, e.g., $\theta = \pi$, and therefore will also have $\theta < 0$, but this is handled naturally by the periodicity of the trigonometric functions.

The coordinate Jacobian matrices $\mathbf{J}$ above are given by

$$\mathbf{J}(r,\theta,\phi) = \begin{pmatrix}\cos\theta\sin\phi & -r\sin\theta\sin\phi & r\cos\theta\cos\phi\\ \sin\theta\sin\phi & r\cos\theta\sin\phi & r\sin\theta\cos\phi\\ \cos\phi & 0 & -r\sin\phi\end{pmatrix}. \tag{46}$$

Therefore, at $\dot{\boldsymbol{q}} = (r, \theta, \phi) = (1, 0, \frac{\pi}{2})$, we have the Jacobian matrix

$$\mathbf{J}\left(1, 0, \frac{\pi}{2}\right) = \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{pmatrix}. \tag{47}$$

We will see below that in Eq. 41, because $\partial L/\partial r = 0$, we furthermore have Eq. 55. Together with the above, we therefore get Eq. 41 as

$$\begin{aligned}\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j}\right)(\boldsymbol{b}_2\ \boldsymbol{b}_3)\begin{pmatrix}\ddot{q}^{2'}\\ \ddot{q}^{3'}\end{pmatrix}(s) &= \mathbf{P}_\perp\frac{\partial L}{\partial \boldsymbol{q}}, \\ \mathbf{B}\left(\frac{\partial^2 L}{\partial \dot{q}^{i'} \partial \dot{q}^{j'}}\right)\mathbf{B}^T(\boldsymbol{b}_2\ \boldsymbol{b}_3)\begin{pmatrix}\ddot{q}^{2'}\\ \ddot{q}^{3'}\end{pmatrix}(s) &= \mathbf{P}_\perp\frac{\partial L}{\partial \boldsymbol{q}}, \\ \left(\frac{\partial^2 L}{\partial \dot{q}^{i'} \partial \dot{q}^{j'}}\right)\begin{pmatrix}0\\ \ddot{q}^{2'}\\ \ddot{q}^{3'}\end{pmatrix}(s) &= \mathbf{B}^T\mathbf{P}_\perp\frac{\partial L}{\partial \boldsymbol{q}}, \\ \begin{pmatrix}\frac{\partial^2 L}{\partial\theta\partial\theta} & -\frac{\partial^2 L}{\partial\theta\partial\phi}\\ -\frac{\partial^2 L}{\partial\phi\partial\theta} & \frac{\partial^2 L}{\partial\phi\partial\phi}\end{pmatrix}\begin{pmatrix}\ddot{q}^{2'}\\ \ddot{q}^{3'}\end{pmatrix}(s) &= (\boldsymbol{b}_2|\boldsymbol{b}_3)^T\frac{\partial L}{\partial \boldsymbol{q}}.\end{aligned} \tag{48}$$

Thus, Eq. 41 can be solved uniquely, if the Hessian matrix $\bar{\mathbf{H}}$ below is non-singular, by solving for the 2D vector $\ddot{\bar{\boldsymbol{q}}}$ in the 2×2 system

$$\bar{\mathbf{H}}\,\ddot{\bar{\boldsymbol{q}}} = \bar{\boldsymbol{r}}, \tag{49}$$

where

$$\bar{\mathbf{H}} = \left.\begin{pmatrix}\frac{\partial^2 L}{\partial\theta\partial\theta} & -\frac{\partial^2 L}{\partial\theta\partial\phi}\\ -\frac{\partial^2 L}{\partial\phi\partial\theta} & \frac{\partial^2 L}{\partial\phi\partial\phi}\end{pmatrix}\right|_{\dot{\boldsymbol{q}}=(1,0,\frac{\pi}{2})}, \quad \bar{\boldsymbol{r}} = (\boldsymbol{b}_2|\boldsymbol{b}_3)^T\frac{\partial L}{\partial \boldsymbol{q}}, \tag{50}$$

and then computing

$$\ddot{\boldsymbol{q}}(s) = \begin{pmatrix}\ddot{q}^1\\ \ddot{q}^2\\ \ddot{q}^3\end{pmatrix}(s) = (\boldsymbol{b}_2|\boldsymbol{b}_3)\,\ddot{\bar{\boldsymbol{q}}}. \tag{51}$$

RHS Hessian block (mixed partial derivatives). For the 3×3 Hessian matrix block on the right-hand side of Eq. 41, we first compute the following gradients on finite difference stencil positions on the spatial $\boldsymbol{q}$ grid, around the current point $\boldsymbol{q}$. That is, we compute the gradients below at positions $\tilde{\boldsymbol{q}}$ with

$$\tilde{\boldsymbol{q}} = \boldsymbol{q} + \begin{pmatrix}0, \pm\Delta x\\ 0, \pm\Delta y\\ 0, \pm\Delta z\end{pmatrix}. \tag{52}$$

Holding the tangent $\dot{\boldsymbol{q}}$, and the corresponding frame $\mathbf{B}\big|_{\dot{\boldsymbol{q}}}$ fixed, at each spatial look-up position $\tilde{\boldsymbol{q}}$ around $\boldsymbol{q}$, we compute the gradient

$$\begin{aligned}\left(\frac{\partial L}{\partial \dot{q}^i}\right)\Big|_{\tilde{\dot{\boldsymbol{q}}}} &= \mathbf{B}\Big|_{\dot{\boldsymbol{q}}} \cdot \mathbf{J}\,(r,\theta,\phi)\begin{pmatrix} 0 \\ \frac{1}{r\sin\phi}\frac{\partial L}{\partial \theta} \\ \frac{1}{r}\frac{\partial L}{\partial \phi}\end{pmatrix}, \\ &= \mathbf{B}\Big|_{\dot{\boldsymbol{q}}} \cdot \mathbf{J}\left(1,0,\frac{\pi}{2}\right)\begin{pmatrix} 0 \\ \frac{\partial L}{\partial \theta} \\ \frac{\partial L}{\partial \phi}\end{pmatrix}, \quad \text{at} \quad \left(1,0,\frac{\pi}{2}\right), \\ &= \mathbf{B}\Big|_{\dot{\boldsymbol{q}}} \cdot \begin{pmatrix} 0 \\ \frac{\partial L}{\partial \theta} \\ -\frac{\partial L}{\partial \phi}\end{pmatrix}, \quad \text{at} \quad \left(1,0,\frac{\pi}{2}\right).\end{aligned} \tag{53}$$

Again, the zero entry results from $\partial L/\partial r = 0$. By using the frame $\mathbf{B}$, these gradient vectors are given in global Cartesian coordinate components. From these gradients, we then compute the 3×3 Hessian matrix for Eq. 41, also in global Cartesian coordinate components, via finite differences of the gradients around the point $\boldsymbol{q}$, to estimate

$$\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial q^j}\right) = \frac{\partial}{\partial q^j}\Big|_{\boldsymbol{q}} \left(\frac{\partial L}{\partial \dot{q}^i}\right). \tag{54}$$

From this equation, and from Eq. 53, we can see that $\dot{\boldsymbol{q}}$ will also be an eigenvector with eigenvalue $\lambda_1 = 0$ of this Hessian matrix. Thus,

$$\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial q^j}\right)\dot{\boldsymbol{q}} = 0. \tag{55}$$

B Additional Results

Fig. 11 highlights the capability of our framework to resolve multiple interacting coherent structures within a globally rotating system. This analytic model involves four distinct vortices orbiting a common axis, presenting a scenario where local spin is superimposed onto a global rotational field. Standard Eulerian detectors often struggle to distinguish the local core rotation from the global orbital shear. Our results show four distinct spacetime surfaces that exhibit perfect topological stability throughout the animation sequence ($t = 0$ to $t = 0.8$). The strict alignment of the extracted cores with the instantaneous streamlines in the co-rotating frame confirms that our variational formulation is objectively robust, decoupling the internal vortex dynamics from the external bulk rotation, without requiring manual reference frame adjustments.

C Implementation Details

We implement our pipeline in C++ with OpenGL for rendering. All experiments were executed on an Intel(R) Xeon(R) Gold 6230R CPU @ 2.10 GHz with two processors. Our datasets include DeltaWing and F22Raptor, generated by our LBM simulations, and Cylinder3D, produced by Baeza Rojo and Günther [2020] using the Gerris flow solver. Our code is publicly available at https://github.com/Cindy-xdZhang/GenericVariationalVortexCore.

C.1 Performance

We solve the second-order ODE given by the Euler-Lagrange equation (Eq. 27) using a fourth-order Runge-Kutta (RK4) integrator. Our Lagrangian L contains an integral over a temporal window $t \in [t_0, T]$. Increasing this window increases the work per computation of L, and thus becomes the dominant factor in the overall RK4 runtime. In practice, we integrate the input vector field $\boldsymbol{v}(x,t)$ to compute the flow map $\phi_{t_0,t}$, and its pushforward $\phi_{t_0,t*}$, with a step size finer than the discrete temporal resolution of $\boldsymbol{v}(x,t)$, but we compute L by accumulating contributions of the energy E only at integer time steps aligned with the original data. As shown in Fig. 12, the per-step RK4 execution time grows noticeably with the temporal window length $|T - t_0|$. A second computational bottleneck is the local Killing field $\boldsymbol{w}$ estimation for each candidate point, which aggregates flow-derived quantities over a spatial neighborhood $U(t) \subset \mathbb{R}^3$. Enlarging the size of the local 3D volume $U(t)$ increases the amount of data processed, and thus the total observer computation time, as also reflected in Fig. 12.

We report runtime statistics for all baselines in Table 2, averaged over three runs. For our method, timings use the above hyperparameters. The *Candidate* column includes candidate filtering and refinement, while *Total* additionally includes RK4 integration to solve the Euler-Lagrange equation. For the VDE baseline [Günther and Theisel 2025], *Compute Field* measures VDE-field computation and Lagrangian integration, and *Extraction* measures extremal curve extraction from the integrated VDE field. Entries marked "–" denote cases where computation is prohibitively expensive or the input is steady (making VDE inapplicable). In both cases, we omit timings. For the VTK baseline, we report the time to extract results for a single time step.

C.2 Pseudocode

We provide pseudocode for the two core steps of our pipeline. Algorithm 1 covers candidate point generation, with grid filtering and seed refinement, and Algorithm 2 details the Euler-Lagrange integration and variational refinement used to trace core lines.

Local observer estimation. In Algorithm 1, when a local observer is needed, we fit a local Killing field $\boldsymbol{w}(x,t)$ by minimizing the observed time derivative in a least-squares sense, following Rautek et al. [2024]. The model has six unknowns (the six degrees of freedom of a Killing field in $\mathbb{R}^3$) per time step, and is fitted over a spatial neighborhood $U(t)$ around the current candidate point. This fit is coupled with seed refinement by alternately re-estimating the observer $\boldsymbol{w}$ and taking a descent step on the objective function L until convergence.

Table 2. **Performance comparison** (time in seconds), for our method and baseline approaches. For **Ours**, *Candidate* measures candidate filtering and refinement, and *Total* additionally includes the RK4 integration for solving the Euler–Lagrange equation. For VDE [Günther and Theisel 2025], *Compute Field* reports computing the VDE field and its Lagrangian integration, while *Extraction* reports extremal curve extraction from the integrated field. For VTK, timings correspond to extracting results for a **single time step**. "-" indicates prohibitively expensive cases or steady input flow fields where VDE is not directly applicable.

Dataset	Spatial Resolution	VTK	Ours		VDE [Günther and Theisel 2025]	
			Candidate	Total	Compute Field	Extraction
Tornado3D	[128, 128, 128]	59	3	3	-	-
VortexRing	[128, 128, 128]	72	3	3	15	45
RFC3D	[64, 64, 32]	6	0.7	4	5	80
F22Raptor	[300, 470, 150]	420	19	29	-	-
Cylinder3D	[320, 120, 40]	190	7	14	1990	1450
DeltaWing3D	[110, 629, 110]	181	4	12	7800	5680

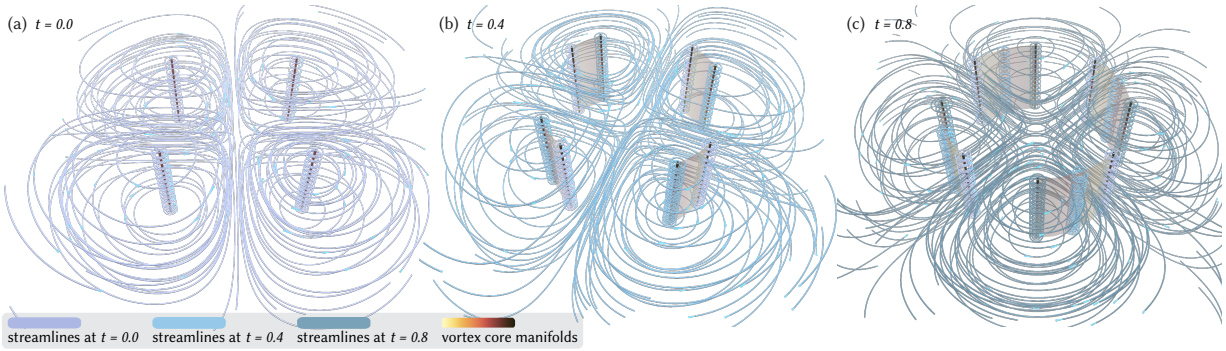

Fig. 11. **Four Centers 3D animation sequence** of an analytical model of four vortex structures rotating around a common axis in the center. The four spacetime surfaces of the core lines evolve over time. Snapshots at (a) $t = 0$, (b) $t = 0.4$ and (c) $t = 0.8$ show perfect alignment of the vortex core lines with the instantaneous streamlines visualized in the co-rotating reference frame. The streamlines perfectly swirl around the vortex core lines at all times.

Core line integration and refinement. Starting from a refined seed $\boldsymbol{q}(s_0)$ with initial tangent $\dot{\boldsymbol{q}}(s_0)$, we integrate the second-order Euler–Lagrange ordinary differential equation (Eq. 27) using a standard fourth-order Runge–Kutta (RK4) scheme. Specifically, at each integration step, the state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is advanced jointly, where the second-order dynamics $\ddot{\boldsymbol{q}}$ are evaluated to update $\dot{\boldsymbol{q}}$, and subsequently $\boldsymbol{q}$ is updated via $\dot{\boldsymbol{q}}$. We denote this procedure as RungeKutta4_Step in Algorithm C.2.

We then further reduce the functional $S[\boldsymbol{q}(s)]$ (Eq. 21) via a functional gradient-flow refinement using the Adam optimizer, following the approach of Daßler and Günther [2024].

C.3 Numerical Stability

We evaluate the numerical stability of our method, where the primary source of error arises from the internal flow map integration (i.e., the temporal pre-integration of Lagrangian trajectories). In Fig. 13, we compare three numerical integrators (Euler, RK4, and RK5) under varying step sizes on the VortexRing dataset. Each curve in the figure differs only in the choice of internal flow map integrator and its step size, while the outer Euler–Lagrange optimization parameters are fixed ($\Delta s = 2 \times 10^{-3}$, $N_s = 50$). In practice, we find that the outer optimization is not the dominant source of numerical failure.

The vertical axis reports the Chamfer distance between the optimized core line at time t_0 and the analytically defined ground truth vortex core line of the dataset, reflecting the error induced by different flow map integration schemes.

The results show that the Euler integrator suffers from significant numerical errors and frequently leads to failure. In contrast, RK4 and RK5 achieve comparable accuracy and demonstrate strong numerical stability. Notably, even under long integration horizons (e.g., $\Delta t = 0.1$ over 100 steps, corresponding to a total integration time of 10), both methods avoid numerical blow-up and maintain reliable performance.

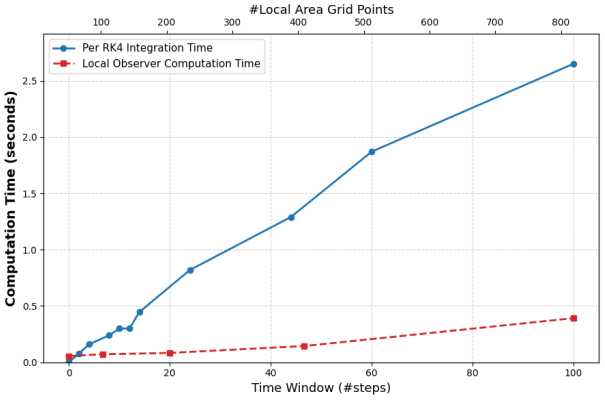

Fig. 12. **Computation time versus problem scale.** Execution time for one RK4 step while varying the temporal integration window in the target Lagrangian (blue circles), and the computation time for local Killing observer estimation while varying the size of the local spherical neighborhood around a candidate point (red squares).

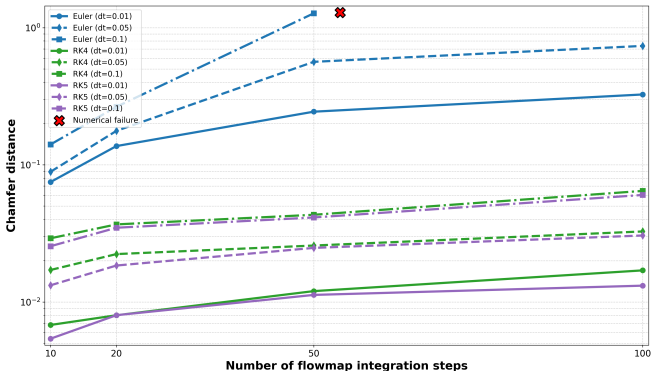

Fig. 13. **Numerical stability**. Comparison of Euler, RK4, and RK5 integrators with varying step sizes (dt) for the flow map integration, evaluated using Chamfer distance to ground truth core lines. Euler exhibits large errors and instability, while RK4 and RK5 remain stable and accurate even over a long integration time window.

Algorithm 1 Candidate Generation with Observer-Aware Seed Refinement

Input: Vector field $\boldsymbol{v}(x,t)$; time t_0; thresholds $\mathrm{IVD}_{\min}$, $V^r_{\max}$; merge radius r_m; refinement parameters $N_{\mathrm{IVP}}, \varepsilon, \eta$.
Output: Refined seed set $\mathcal{S}$
Define:
$\boldsymbol{v}^*(x,t) \leftarrow \boldsymbol{v}(x,t) - \boldsymbol{w}(x,t)$
$\boldsymbol{e}_3(x,t) \leftarrow$ real eigenvector of $\nabla \boldsymbol{v}^*(x,t)$
$\boldsymbol{v}^r \leftarrow \boldsymbol{v}^* - (\boldsymbol{v}^* \cdot \boldsymbol{e}_3)\boldsymbol{e}_3$
Stage 1: Grid filtering + merge
$C \leftarrow \emptyset$
for all grid points $\mathbf{p}$ **do**
  if $\mathrm{IVD}(\mathbf{p}, t_0) < \mathrm{IVD}_{\min}$ **then**
    continue
  end if
  if $\boldsymbol{e}_3(\mathbf{p}, t_0)$ does not exist **then**
    continue
  end if
  if $\|\boldsymbol{v}^r(\mathbf{p}, t_0)\| > V^r_{\max}$ **then**
    continue
  end if
  $C \leftarrow \mathrm{InsertMerged}(C, \mathbf{p}, \|\boldsymbol{v}^r(\mathbf{p}, t_0)\|, r_m)$
end for
Stage 2: Seed refinement
$\mathcal{S} \leftarrow \emptyset$
for all $\mathbf{p} \in C$ **do**
  $\boldsymbol{q} \leftarrow \mathbf{p}$
  for $k = 1$ to N_{IVP} **do**
    $\dot{\boldsymbol{q}} \leftarrow \boldsymbol{e}_3(\boldsymbol{q}, t_0)$
    $\boldsymbol{w}_{\boldsymbol{q}} \leftarrow \mathrm{ComputeLocalKillingObserver}(\boldsymbol{v}, \mathcal{U}(\boldsymbol{q}), t_0)$
    $\nabla_{\boldsymbol{q}} L \leftarrow \mathrm{FiniteDiffGrad}(L(\boldsymbol{q}, \dot{\boldsymbol{q}}))$
    if $\|\nabla_{\boldsymbol{q}} L\| < \varepsilon$ **then**
      break
    end if
    $\boldsymbol{q} \leftarrow \boldsymbol{q} - \eta \nabla_{\boldsymbol{q}} L$
  end for
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{q}\}$
end for
return $\mathcal{S}$
Function InsertMerged$(\cdot)$**:**
for all $\mathbf{c} \in C$ **do**
  if $\|\mathbf{c} - \mathbf{p}\| < r_m$ **then**
    if $\mathrm{score}(\mathbf{p}) < \mathrm{score}(\mathbf{c})$ **then**
      replace $\mathbf{c}$ with $\mathbf{p}$
    end if
    return C
  end if
end for
$C \leftarrow C \cup \{\mathbf{p}\}$
return C

Algorithm 2 Euler–Lagrange Integration with RK4

Input: Refined seed $\boldsymbol{q}(s_0)$ at time t_0; vector field $\boldsymbol{v}(x,t)$; observer $\boldsymbol{w}(x,t)$; time window $[t_0, T]$; step size Δs; max steps N_s; thresholds $\mathrm{IVD}_{\min}, \theta_{\max}$.
Output: Vortex core polyline C
Define:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}, s) \leftarrow \int_{t_0}^{T} E\big(\phi_{t_0,t}(\boldsymbol{q}),\ \phi_{t_0,t*}(\dot{\boldsymbol{q}}),\ t\big)\ dt$$

$s \leftarrow 0$
$\dot{\boldsymbol{q}} \leftarrow \boldsymbol{e}_3(\boldsymbol{q}(s_0), t_0)$
$\boldsymbol{q} \leftarrow \boldsymbol{q}(s_0)$
$C \leftarrow \{(\boldsymbol{q}, s)\}$
for $i = 1$ to N_s **do**
  $(\boldsymbol{q}', \dot{\boldsymbol{q}}', err) \leftarrow \mathrm{RungeKutta4\_Step}(\boldsymbol{q}, \dot{\boldsymbol{q}}, s)$
  if not $\mathrm{AcceptStep}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}', \dot{\boldsymbol{q}}', err)$ **then**
    break
  end if
  $\boldsymbol{q} \leftarrow \boldsymbol{q}'$
  $\dot{\boldsymbol{q}} \leftarrow \dot{\boldsymbol{q}}'$
  $s \leftarrow s + \Delta s$
  append $(\boldsymbol{q}, s)$ to C
end for
return C
Function AcceptStep$(\cdot)$**:**
if $\mathrm{IVD}(\boldsymbol{q}', t_0) < \mathrm{IVD}_{\min}$ **then**
  return false
end if
if $\boldsymbol{e}_3(\boldsymbol{q}', t_0)$ does not exist **then**
  return false
end if
if $\angle(\dot{\boldsymbol{q}}', \dot{\boldsymbol{q}}) > \theta_{\max}$ **then**
  return false
end if
if err indicates numerical failure **then**
  return false
end if
return true

C.4 Hyperparameters

We use the following settings throughout our experiments. For the integration to determine the flow map $\phi_{t_0,t}$, we set the step size to $dt = \frac{1}{10}\Delta t$, where Δt is the temporal spacing of the input vector field $\boldsymbol{v}(x,t)$. We refine each seed using $N_{\mathrm{IVP}} = 20$ gradient descent steps. The resulting coreline curve is further refined via functional gradient descent for $N_{\mathrm{EL}} = 20$ Adam steps, both with a learning rate of 10^{-3}.

For the Euler–Lagrange integration of the optimal curve, we use $\Delta s = 2 \times 10^{-3}$. The maximum number of RK4 steps N_s is chosen per dataset to achieve the desired coreline length and typically lies in $[50, 100]$; integration is performed in both directions.

We estimate gradients and Hessians via numerical differentiation with step size $\Delta h = 10^{-3}$. The temporal window $[t_0, T]$ is adjusted per dataset to match intrinsic time scales: We choose a window spanning about 20–50 input time steps, i.e., $T \in [t_0 + 20\Delta t,\ t_0 + 50\Delta t]$. The radius of the local spherical neighborhood $U(t)$ for extracting the Killing observer $\boldsymbol{w}(x,t)$ is typically 5–10 times the spatial grid spacing Δx.

We typically set the energy term $C(\boldsymbol{c}, \dot{\boldsymbol{c}}, t)$, that represents a generic vortex criterion to minimize, to the negative instantaneous vorticity deviation: $C(\boldsymbol{c}, \dot{\boldsymbol{c}}, t) = -\mathrm{IVD}(\boldsymbol{c}, t)$. We adopt the following default weighting for the individual terms of the energy E (Eq. 23):

$$\begin{aligned} \mathrm{weight}(\tilde{\boldsymbol{u}}) &= \mu_{\tilde{\boldsymbol{u}}} = 0.2, \\ \mathrm{weight}(\boldsymbol{C}) &= \mu_c = 0.1/\mu_{\tilde{\boldsymbol{u}}}, \\ \mathrm{weight}(\boldsymbol{D}) &= \mu_d = 0.0/\mu_{\tilde{\boldsymbol{u}}}, \\ \mathrm{weight}(\boldsymbol{R}) &= \mu_r = 0.2/\mu_{\tilde{\boldsymbol{u}}}. \end{aligned} \tag{56}$$

This balance emphasizes geometric coherence and vortex alignment, while de-emphasizing material acceleration when the flow exhibits strong rotational structures. Note that because the weighting of all energy terms has one less degree of freedom, in Eq. 23, $\mu_{\tilde{\boldsymbol{u}}}$ is not explicitly listed and is set to 1.0.