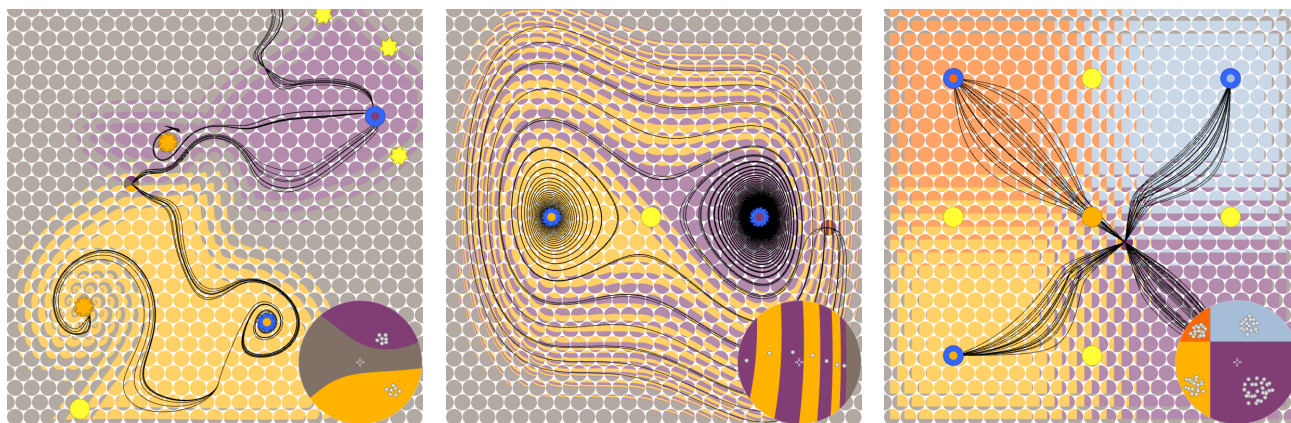


# Inertial Steady 2D Vector Field Topology

Tobias Günther and Holger Theisel

Visual Computing Group, University of Magdeburg



**Figure 1:** Visualization of inertial critical points and a glyph-based visualization of the asymptotic behavior of inertial particles. The sink (blue critical point) that is reached by an inertial particle depends on its initial position  $\mathbf{x}_0$  and the initial velocity  $\mathbf{v}_0$ . The glyph location encodes  $\mathbf{x}_0$  and the glyph disc color-codes the sink that is reached for a certain  $\mathbf{v}_0$ . The glyph center represents  $\mathbf{v}_0 = \mathbf{0}$  and the selected glyph is enlarged in the bottom right corner, showing the seeds of inertial particle trajectories. Left: BORROMEAN RINGS with particle diameter  $d_p = 70\mu\text{m}$  and maximal velocity  $v_{\text{max}} = 10\text{m/s}$  (occurs at glyph boundary), middle: DUFFING OSCILLATOR with  $d_p = 100\mu\text{m}$ ,  $v_{\text{max}} = 5\text{m/s}$ , and right: NINE CP with  $d_p = 70\mu\text{m}$ ,  $v_{\text{max}} = 10\text{m/s}$ . Inertial critical point types are listed in Table 1.

## Abstract

Vector field topology is a powerful and matured tool for the study of the asymptotic behavior of tracer particles in steady flows. Yet, it does not capture the behavior of finite-sized particles, because they develop inertia and do not move tangential to the flow. In this paper, we use the fact that the trajectories of inertial particles can be described as tangent curves of a higher dimensional vector field. Using this, we conduct a full classification of the first-order critical points of this higher dimensional flow, and devise a method to their efficient extraction. Further, we interactively visualize the asymptotic behavior of finite-sized particles by a glyph visualization that encodes the outcome of any initial condition of the governing ODE, i.e., for a varying initial position and/or initial velocity. With this, we present a first approach to extend traditional vector field topology to the inertial case.

This is the authors preprint. The definitive version is available at <http://diglib.eg.org/> and <http://onlinelibrary.wiley.com/>.

## 1. Introduction

Vector field topology is a powerful tool to study asymptotic flow behavior and compactly describe fluid dynamics. Traditional vector field topology in flow visualization considers the motion of massless tracer particles of zero size in a fluid, and not the motion of finite-sized objects immersed therein. These finite-sized objects are called

*inertial particles*, and in contrast to massless tracer particles, a number of forces is applied to them. The CFD literature gives strong evidence that inertial particle dynamics vary considerably from the infinitesimal dynamics [OOG08, HS08]. Inertial particles are of concern in a number of scientific and engineering disciplines, such as sand saltation modeling [SL99], soiling of cars [RSBE01], visual obscuration in helicopter landing maneuvers [KGRK14], formation

of rain [Bor11], jellyfish feeding [PD09], plant spores and pathogens carried by atmospheric flow [BH02], fuel injectors transporting droplets at supersonic speeds [MTP\*02] and observations of charged particles in magnetic fields [BZ89].

The motion of inertial particles is governed by an ODE, and their asymptotic behavior depends on the initial conditions, i.e., initial position (2D) and initial velocity (2D). Visualizing the possible outcome of any initial condition (4D) and understanding separatrices that divide the 4D spatio-velocity domain into compartments of homogeneous asymptotic inertial flow behavior are challenging tasks. Further, many concepts of vector field topology rely on the ability to integrate particles backwards. Extending these concepts to the inertial case is in part challenging, since inertial backward integration toward critical points (in the dynamical systems literature these are also known as fixed points, stationary points or singularities) exhibits a strong repelling behavior [HS08]. We explain this phenomenon by the use of differential vector field topology.

Inertial particle trajectories can be considered as tangent curves of a higher dimensional vector field [GT14]. In this paper, we conduct a full classification of the inertial critical points of this higher dimensional vector field, and show that the location of the critical points can be efficiently computed in 2D. Among others, we found that sources cannot exist in inertial flows, which explains the strong repelling behavior during backward integration. To visualize the asymptotic behavior of inertial particles, we propose a glyph visualization that encodes for every initial position and initial velocity the critical point that is reached in the limit. To obtain interactive rates, we compute the glyphs progressively and adaptive to the current view. With this, we present a first step to extend traditional vector field topology to the inertial case, and present a visualization that encodes the possible outcome of varying initial conditions to an ODE. We apply our extraction and visualization to a number of synthetic and real-world 2D steady flows, including the potential field of a benzene molecule and magnetic fields of decaying magnetic rings.

**Notation:** We denote a point/vector in 2D space by bold letters, e.g.,  $\mathbf{x}$ ,  $\mathbf{u}$ . Since for inertial particles the direction of movement does not only depend on the vector field  $\mathbf{u}$  but also depends on the current particle velocity  $\mathbf{v}$ , we denote 4D points/vectors as  $\tilde{\mathbf{x}}, \tilde{\mathbf{u}}$  where the first 2 components refer to the spatial locations and the last 2 refer to the current particle velocity. Further,  $\mathbf{0}_{2,2}$  denotes the  $2 \times 2$  zero matrix,  $\mathbf{I}_2$  is the  $2 \times 2$  identity matrix, and  $\mathbf{0}_2$  and  $\mathbf{0}_4$  are the 2D and 4D zero vector, respectively. For brevity, we omit the index when it can be inferred from context. We denote the imaginary unit  $i$ , and  $Re(z)$  and  $Im(z)$  the real and imaginary part of a complex number  $z$ .

## 2. Background and Related Work

### 2.1. Modeling of Inertial Particles

In contrast to massless tracer particles, inertial particles are not moving tangential to the underlying flow. Instead, they are subject to a number of forces, including a force exerted by the flow itself, buoyancy, Stokes drag, the force exerted due to the mass of the fluid moving with the particle and the Basset-Boussinesq memory term, cf. Haller and Sapsis [HS08]. Today's most accepted form of these forces was described by Maxey and Riley [MR83] for small rigid spherical particles. The properties of the solutions to

their equations of motion and the history of their corrections were recently documented by Farzamand and Haller [FH15].

Depending on the application, several assumptions can be made that simplify the equations of motion considerably. We follow the model described by Crowe et al. [CST98], which assumes that particles are spherical and very small in size. Additionally, the particle density is assumed to be far higher than the density of the surrounding fluid, which allows to neglect buoyancy. Further assuming dilute flow, the particle motion is dominated by drag forces, rather than particle-particle collision. Thus, we neglect collision handling and particle rotation, and assume one-way coupling, i.e., particles do not affect the surrounding fluid. These simplifications are common and were used in related CFD literature, e.g., [SGL10, PSGC11a, KGRK14, CGP\*10, BBC\*09, BBC\*11].

Günther et al. [GT14, GT15] described the trajectories of inertial particles as tangent curves of a high-dimensional vector field, which models both the rate of change of particle position  $\mathbf{x}$  and particle velocity  $\mathbf{v}$ . For a steady underlying 2D flow  $\mathbf{u}(\mathbf{x})$ , the governing 4D vector field becomes:

$$\tilde{\mathbf{u}}(\mathbf{x}, \mathbf{v}) = \frac{d}{dt} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \frac{\mathbf{u}(\mathbf{x}) - \mathbf{v}}{r} + \mathbf{g} \end{pmatrix} \quad \text{with} \quad \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} (0) = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{v}_0 \end{pmatrix} \quad (1)$$

where  $\mathbf{g}$  is a gravity vector,  $\mathbf{x}_0$  and  $\mathbf{v}_0$  are the initial particle position and velocity, and  $r$  is the particle response time. The  $4 \times 4$  Jacobian field of  $\tilde{\mathbf{u}}(\mathbf{x})$  was given in [GT14] as:

$$\tilde{\mathbf{J}}(\mathbf{x}, \mathbf{v}) = \begin{pmatrix} \mathbf{0}_{2,2} & \mathbf{I}_2 \\ \frac{1}{r} \mathbf{J} & -\frac{1}{r} \mathbf{I}_2 \end{pmatrix}. \quad (2)$$

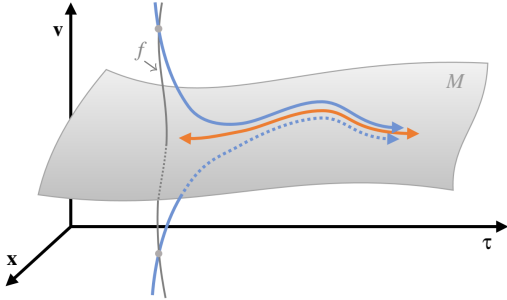
The response time  $r$  is characterized by particle diameter  $d_p$  and particle density  $\rho_p$ , as well as viscosity  $\mu$  of the surrounding fluid:

$$r = \frac{d_p^2 \rho_p}{18\mu} > 0 \quad (3)$$

Figuratively spoken, the response time is the time required for a particle released from rest in a gravity-free environment to acquire 63% of the velocity of the carrying fluid, cf. [CST98]. For all examples in the paper, we used as particle density  $\rho_p$  the density of dry sand, i.e.,  $\rho_p = 1600 \text{ kg/m}^3$ . The diameter  $d_p$  varies between  $d_p = 70 \mu\text{m}$  and  $d_p = 200 \mu\text{m}$ . Note that these equations of motion hold for  $d_p \ll \eta_k$ , with  $\eta_k$  being the Kolmogorov length scale. The surrounding medium was assumed to be air, thus the viscosity was set to  $\mu = 1.532 \cdot 10^{-5} \text{ kg/(m}\cdot\text{s)}$ . Given these parameters, response time  $r$  is in our experiments in the range  $r \in [0.028, 0.232]$ .

Alternative approaches to the modeling of inertial particles include the modeling of gravity and buoyancy as external forces by the material derivative, cf. [BBC\*09] and the explicit consideration of buoyancy to model both aerosols and bubbles [HS08].

A recent report on experimental and computational fluid dynamics of inertial particles in turbulent fluids was compiled by Balachandrar and Eaton [BE10]. Of interest in the inertial fluid dynamics literature are problems such as direct numerical simulations of jets [PSGC11a], stirring of sand during helicopter maneuvers [SGL10], comparisons between experimentally determined and simulated trajectories [OOG08], energy dissipation along trajectories [BBC\*09], or the effects of inertial particles on the underlying flow (two-way coupling) [PSGC11b].



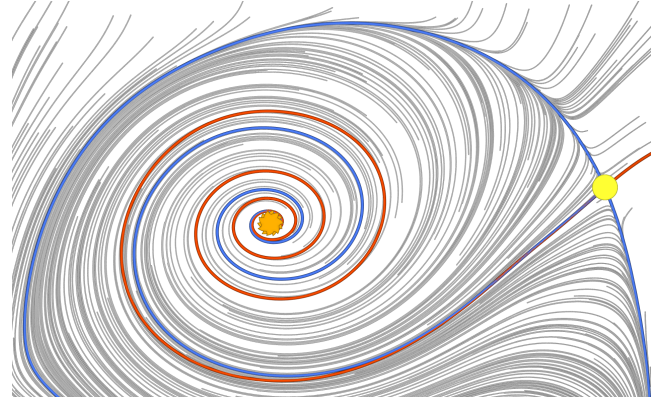
**Figure 2:** Inertial particle trajectories (blue) are attracted by a manifold  $M$ , and repel away from it during backward integration. The union of initial conditions  $(\mathbf{x}_0, \mathbf{v}_0)$  that converge to the same trajectory is called a fiber  $f$ . Haller and Sapsis [HS08] derived an ODE to move on the manifold  $M$  (orange) in order to integrate backward without repelling behavior. (Sketch is based on [HS08].)

## 2.2. Backward Integration

In traditional massless vector field topology, many visualization concepts rely on the ability to integrate particles backward in time. In the context of dispersed fluids, backward integration is also referred to as source inversion, which is of interest to detect localized sources of pollution in air or water. (These problems additionally require to recover diffusion processes.) While the equations of motion of inertial particles in Eq. (1) are well-defined and perfectly allow backward integration, there are certain pitfalls that make the backward integration challenging.

During forward integration, inertial particles are attracted to manifolds in the spatio-velocity domain and cluster on them, cf. [MBZ06, HS08]. In fact, multiple initial conditions of  $(\mathbf{x}_0, \mathbf{v}_0)$  might attract to the same spatio-velocity structure, as illustrated in Fig. 2. The union of initial conditions that converges to the same structure is called a fiber or a limit set in dynamical systems theory. Consequentially, a backward integration *repels away* from such an attracting manifold. Haller and Sapsis [HS08] derived an ODE that allows to move *on* an attracting manifold, which they called the *inertial equation*. Applied to our model, a first-order approximation essentially boils down to an integration in the vector field  $\mathbf{u} + r\mathbf{g}$ . This 2D field allows to integrate backward without repelling, but it (a) does not recover the initial velocity of an inertial particle [HS08], and (b) instabilities occur for larger particles [SH09] that drive inertial particle trajectories away from the attracting manifold on which the inertial equation is valid. For their particle model, Sapsis and Haller [SH09] derived a threshold that characterizes when this happens, which was applied in [SPH11]. Fig. 3 shows for example that separatrices recovered from backward integration in  $\mathbf{u} + r\mathbf{g}$  do not match the actual separatrices that are found via bisection and forward integration of the actual inertial flow of Eq. (1). In this paper, we give another intuitive explanation why this repelling behavior occurs by observation and classification of the inertial critical points in the spatio-velocity domain. The extraction of topology of inertial flow becomes much more complicated than for traditional massless flow, since backward integration toward critical points is practically impossible due to the strong repelling behavior.

Note that a second pitfall is of numerical nature. During integration with step size  $h$  the error in the position evolves at rate



**Figure 3:** The orange critical point acts as a source in Haller's inertial equation, and the red separatrix is found by backward integration from the saddle (yellow critical point on the right). It can be seen that inertial forward trajectories (gray) of particles released from rest for  $d_p = 70\mu\text{m}$  cross the red line. Blue lines are actual trajectories of inertial particles that connect the critical points.

$O(h)$ , while the error in the velocity evolves at  $O(h/r)$ , cf. [HS08]. Thus, for very small particles ( $r \ll 1$ ), the error in the velocity becomes very high. Adaptive higher-order Runge-Kutta integration or a scaling of the time step by factor  $r$  avoid this so-called singular perturbation problem, cf. [HS08, GT14].

## 2.3. Inertial Particles in Visualization

The observation of inertial particles is a relatively young field in the visualization community, which opened up new challenges. It centers around the extension of concepts that were originally devised for traditional massless flows in order to reach out to new application areas. Early work by Roettger et al. [RSBE01], conducted sand particle simulations to determine the soiling of cars in a wind tunnel. They visualized the sand concentration on the cars via heat maps.

To abstract from the underlying equations of motion, Günther et al. [GKKT13] extended the concept of *flow map*  $\phi$  (cf. [Hal01, SLM05]) to inertial particles in unsteady flow. We follow their notation and denote the mass-dependent flow map  $\phi^\tau(\mathbf{x}, r, \mathbf{v}_0)$  as a function that describes where an inertial particle with response time  $r$ , seeded at location  $\mathbf{x}$ , and with an initial velocity  $\mathbf{v}_0$  moves to during integration in a steady flow  $\mathbf{u}$  over a time interval  $\tau$ . This formulation hides the underlying equations of motions, which allows us to describe the asymptotic behavior of inertial particles in a generalized and abstract way. Based on the mass-dependent flow map, Günther et al. further defined four natural classes of inertial integral curves in unsteady flow, comprising mass-dependent versions of pathlines, streaklines and timelines, as well as a new class called *masslines*, which connect inertial particles that were released from the same location at the same time, but with *varying mass*.

Günther and Theisel [GT14] extracted the mass-dependent vortex cores of inertial particles in 3D flow by applying well-known concepts of massless flow to the high-dimensional 6D vector field in Eq. (1), as well as its unsteady version in 7D. Due to the dependence of swirling in the spatial and the velocity subspace, it turns out that

the vortex criteria in both steady and unsteady flow can be reduced to 3D parallel vectors problems.

Lagrangian coherent structures (LCS) have also been extended to inertial particles. Haller [Hal01] has demonstrated that LCS can be extracted as ridges of the finite-time Lyapunov exponent (FTLE) field. Inertial Lagrangian coherent structures (ILCS or pLCS) have been defined in [SH09] and [PD09] as ridges of an FTLE computation based on the trajectories of inertial particles. The inertial FTLE field measures the separation of nearby placed, equally-sized particles. More recently, Günther and Theisel [GT15] defined the finite-time mass separation (FTMS) that measures the separation of particles that were released from the same location but with different size. They developed an exploration tool for 2D flow that shows for every seed location in the spatial domain the inertial particle that exhibits the strongest separation. Based on that they provide additional coordinated views to explore the mass-induced separation in space-time, as well as the temporal evolution of the distribution of masses that were released from a given seed point. For simplicity, Günther et al. [GKKT13, GT14, GT15] always set the initial velocity to  $\mathbf{v}_0 = \mathbf{0}$ . In this paper, we aim to visualize the asymptotic behavior of inertial particles that start with variable initial velocity.

## 2.4. Traditional Vector Field Topology

Topological methods aim for a segmentation of the domain into regions of similar asymptotic flow behavior. They have been introduced to the visualization community by Helman and Hesselink [HH89] for 2D and were later extended to 3D [HH91], including the classification of first-order critical points by the Jacobian's eigenvalues, separatrices starting at saddles and attachment/detachment points at no-slip boundaries. Definitions of higher-order critical points [SKMR98], boundary switch points [dLvL99], closed separatrices [WS01], saddle connectors [TWHS03], boundary switch connectors [WTHS04a] and topology in two-parameter dependent vector fields [WTHS06] followed shortly later. Vector field topology in 3D was visualized in [GLL91, LDG98, MBS\*04]. See Weinkauff [Wei08] for a comprehensive introduction. Topological methods spurred work to smooth [WJE01], compress [LRR00, TRS03] and model [The02, WTHS04b] or edit [CML\*07] vector fields. For further reading we refer to the reports of Laramee et al. [LHZIP07] and Pobitzer et al. [PPF\*11].

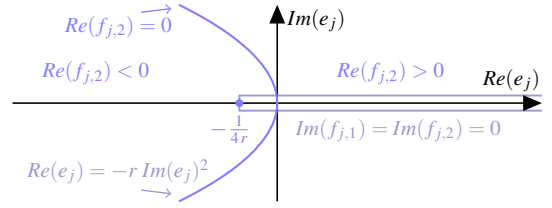
## 3. Inertial Topology

In this section, we utilize that the behavior of inertial particles is governed by a 4D vector field, in which inertial trajectories appear as tangent curves. To capture the asymptotic behavior of inertial particles, we conduct a topological analysis of this high-dimensional flow  $\tilde{\mathbf{u}}$ . For this, we consider inertial critical points and separatrices.

### 3.1. Inertial Critical Points

Given a steady 2D flow  $\mathbf{u}(\mathbf{x})$ , response time  $r$  and gravity  $\mathbf{g}$ , we search for isolated first-order critical points of the inertial 4D flow from Eq. (1) as locations where  $\tilde{\mathbf{u}}(\mathbf{x}, \mathbf{v}) = \mathbf{0}_4$ . This gives  $\mathbf{v} = \mathbf{0}_2$  as the solution for the velocity components and yields the following 2D condition for retrieving the location in the spatial subspace:

$$\mathbf{u}(\mathbf{x}) + r\mathbf{g} = \mathbf{0}_2. \quad (4)$$



**Figure 4:** A classification of the critical points in 4D. Each eigenvalue  $e_j$  of the massless Jacobian  $\mathbf{J}$  leads to two eigenvalues  $f_{j,1}, f_{j,2}$  of the 4D inertial Jacobian  $\tilde{\mathbf{J}}$ . Thereby,  $Re(f_{j,1}) \leq Re(f_{j,2})$  and  $Re(f_{j,1}) < 0$ .

Note that this is consistent with the massless case, since for  $r = 0$  it retrieves locations where  $\mathbf{u}(\mathbf{x}) = \mathbf{0}_2$ . For the special case of gravity-free environments, i.e.,  $\mathbf{g} = \mathbf{0}_2$ , we have a one-to-one relation between the critical points of  $\tilde{\mathbf{u}}$  and  $\mathbf{u}$ : a critical point  $\mathbf{x}_c$  of  $\mathbf{u}$  corresponds to the critical point  $(\mathbf{x}_c, \mathbf{0}_2)$  of  $\tilde{\mathbf{u}}$ . In order to classify the critical points of  $\tilde{\mathbf{u}}$ , we conduct an eigenanalysis of  $\tilde{\mathbf{J}}$  at the critical points. The four eigenvalues  $f_{1,1}, f_{1,2}, f_{2,1}$  and  $f_{2,2}$  of  $\tilde{\mathbf{J}}$  can be expressed in terms of the eigenvalues  $e_1, e_2$  of  $\mathbf{J}$  as:

$$f_{j,1} = \frac{-1 - \sqrt{1 + 4r e_j}}{2r}, \quad f_{j,2} = \frac{-1 + \sqrt{1 + 4r e_j}}{2r} \quad (5)$$

for  $j \in \{1, 2\}$ . Note that the eigenvalues  $f_{j,1}$  and  $f_{j,2}$  that stem from the eigenvalue  $e_j$  are not necessarily complex conjugates. Using the common convention that the square root of a number has a non-negative real part, we have  $Re(f_{j,1}) \leq Re(f_{j,2})$ . The proof of Eq. (5) is in a Maple sheet in the accompanying material. Eq. (5) gives a simple relation between the eigenvalues of  $\tilde{\mathbf{J}}$  and  $\mathbf{J}$ , which allows a complete classification of  $f_{j,k}$  based on  $e_j$ . From Eq. (5) follows

$$\frac{f_{j,1} + f_{j,2}}{2} = -\frac{1}{2r} < 0 \quad (6)$$

which entails interesting properties. It means that each pair of eigenvalues  $f_{j,1}, f_{j,2}$  is located diametrically opposite around the real-valued constant center  $-1/(2r)$  in the complex plane. Since, the mean of two eigenvalues is negative, it follows

$$Re(f_{j,1}) < 0. \quad (7)$$
















Hence, every inertial critical point in  $\tilde{\mathbf{u}}$  must exhibit in at least two directions attracting behavior (one for each pair). For  $Re(f_{j,2})$ , both a positive or a negative sign is possible. Here the condition is

$$Re(f_{j,2}) = 0 \iff Re(e_j) = -r Im(e_j)^2 \quad (8)$$

as shown in the appendix. Thus, the locations at which the sign flips describe a parabola in the complex plane. The remaining question is when  $f_{j,k}$  have imaginary parts, i.e., indicate a swirling behavior. Eq. (5) gives that for  $Im(f_{j,k}) = 0$  the radicand must be real and non-negative. This gives

$$Im(f_{j,k}) = 0 \iff Re(e_j) \geq -\frac{1}{4r} \text{ and } Im(e_j) = 0 \quad (9)$$

for  $k \in \{1, 2\}$ . Fig. 4 illustrates the classification of  $f_{j,1}, f_{j,2}$  depending on  $e_j$  in the complex plane. From this and the fact that  $e_1$  and  $e_2$  are either real-valued (on the abscissa) or complex conjugates, we get a complete geometric classification of all possible 4D inertial critical points, as shown in Table 1. There, the left-hand

massless CP (2D)		→		inertial CP (4D)
 rep. node	++	→	+ - + -	2/2 saddle 
 rep. focus	$\begin{smallmatrix} ++ \\ i i \end{smallmatrix}$	→	$\begin{smallmatrix} + - + - \\ i i i i \end{smallmatrix}$	2/2 saddle 
 center	$\begin{smallmatrix} \circ \circ \\ i i \end{smallmatrix}$	→	$\begin{smallmatrix} + - + - \\ i i i i \end{smallmatrix}$	2/2 saddle 
 attr. focus	--	→	$\begin{smallmatrix} + - + - \\ i i i i \end{smallmatrix}$	2/2 saddle 
	$i i$	→	$\begin{smallmatrix} - - - - \\ i i i i \end{smallmatrix}$	attr. focus 
 attr. node	--	→	$\begin{smallmatrix} - - - - \\ i i \end{smallmatrix}$	attr. focus 
		→	-- --	attr. node 
 saddle	+-	→	+ - - -	1/3 saddle 
		→	$\begin{smallmatrix} + - - - \\ i i \end{smallmatrix}$	1/3 saddle 

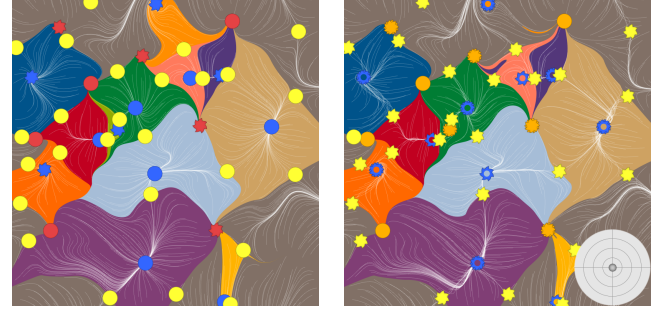
**Table 1:** Full classification of first-order inertial critical points in 2D flows. Left: shows for all possible 2D critical points whether eigenvalues have positive/negative/zero real part (+/-/0), and whether eigenvalues are complex (i). Right: all possible 4D critical points that can be deduced from the 2D points. Zags at glyph boundaries denote presence of 2 or 4 complex eigenvalues, respectively.

side shows all possible 2D critical points with the following coding for the eigenvalues: +/−/0 means positive/negative/zero real part, while  $i$  symbolizes the presence of an imaginary part. The right-hand side depicts all possible 4D critical points that can be deduced from the 2D points. Table 1 also shows the glyphs that we designed for each of the 2D and 4D critical points, respectively. (The glyph design of 2D critical points follows the common choices in the literature [Wei08].) Attracting nodes and foci in 4D have an empty interior that is used later for color-coding an additional property. The additional material contains an extended classification overview of inertial critical points. Table 1 reveals a number of observations of the behavior of the inertial critical points of  $\tilde{\mathbf{u}}$ :

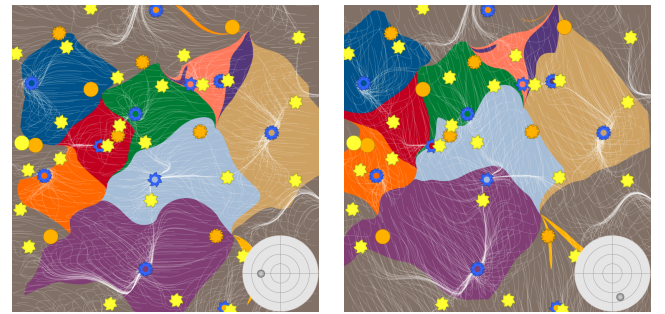
- $\tilde{\mathbf{u}}$  does not have sources (repelling critical points). All critical points have at least two eigenvalues with negative real parts. Thus, during backward integration an inertial particle cannot converge to a critical point, which explains topologically the strong repelling behavior of inertial backward integration.
- The 4D counterpart of a 2D saddle is always a 1/3 saddle, i.e., a saddle with 1 repelling direction and 3 eigenvalues denoting attracting behavior. This observation is consistent with findings in related CFD literature [HS08], namely that saddles stay saddles, though their eigenvalues have not been classified so far.
- For  $\mathbf{g} = \mathbf{0}$ , every 4D sink corresponds to a 2D sink, but not reversely: a 2D attracting focus corresponds in 4D to either a sink or a 2/2 saddle. An example of the latter is shown later in Fig. 5.

### 3.2. Inertial Separatrices

All locations in the spatio-velocity domain  $D \subseteq \mathbb{R}^2 \times \mathbb{R}^2$  that reach the same isolated first-order critical point  $\mathbf{p} \in D$  after infinite forward



**Figure 5:**  $\omega$ -basins of the underlying flow field  $\mathbf{u}(\mathbf{x})$  (left) and projections of the 4D  $\omega$ -basins of inertial flow onto the spatial subspace for  $d_p = 75 \mu\text{m}$  (right). Note that in the projection, the basins of inertial flow may consist of multiple pieces and that in this example an attracting focus turned into a 2/2 saddle.



**Figure 6:** Projections of the 4D  $\omega$ -basins of inertial flow onto the spatial subspace for  $d_p = 75 \mu\text{m}$ ,  $v_{\text{max}} = 10 \text{m/s}$  and different initial velocities. Left:  $\mathbf{v}_0 = (-5, 0)^T$  and right:  $\mathbf{v}_0 = (2, -6)^T$ .

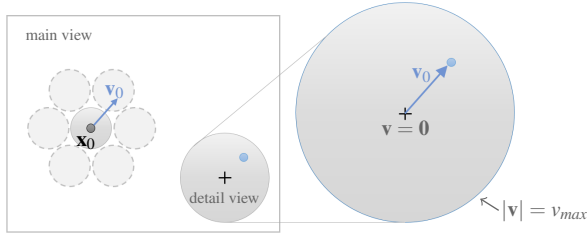
integration form a so-called  $\omega$ -basin, which in dynamical systems is frequently referred to as stable set of  $\mathbf{p}$ :

$$\mathcal{S}(\mathbf{p}) = \left\{ \begin{pmatrix} \mathbf{x}_q \\ \mathbf{v}_q \end{pmatrix} \in D : \phi^\tau(\mathbf{x}_q, r, \mathbf{v}_q) \rightarrow \mathbf{p} \text{ as } \tau \rightarrow \infty \right\} \quad (10)$$

Since there are no sources in the inertial flow, backward integration has zero probability to terminate in a critical point.

We are interested in the separatrices of the 4D inertial flow. They are 3-manifolds starting at 1/3 saddles which follows from a direct generalization from the 2D and 3D case: in 2D, separatrices are 1-manifolds (curves) starting from 1/1 saddles, while in 3D the separatrices are 2-manifolds starting from 1/2 saddles.

In 4D, the 2/2 saddles also indicate a separation but are topologically of less interest because they only separate the particles converging towards the saddle (building a 2-manifold, i.e., a random particle has a zero probability to be on this manifold) vs. particles passing by the saddle (an example for this is in Fig. 3). While the  $\omega$ -basins are connected components in 4D, their projection to the 2D spatial space can consist of several unconnected components. Fig. 5 compares the resulting  $\omega$ -basins of the underlying 2D flow  $\mathbf{u}(\mathbf{x})$  with the projections of 4D basins of the inertial flow with initial velocity  $\mathbf{v}_0 = \mathbf{0}$  for a gravity-free environment. Aside from response time  $r$  and gravity  $\mathbf{g}$ , the 4D basins depend on the initial velocity  $\mathbf{v}_0$ , as demonstrated in Fig. 6. We would like to capture the impact of  $\mathbf{v}_0$



**Figure 7:** Concept of the visualization. The location of a glyph in the domain corresponds to an initial position  $\mathbf{x}_0$ . A point inside a glyph corresponds to an initial velocity  $\mathbf{v}_0$ . Thereby, the glyph center corresponds to  $\mathbf{v} = \mathbf{0}$  and the boundary has  $|\mathbf{v}| = v_{max}$ .

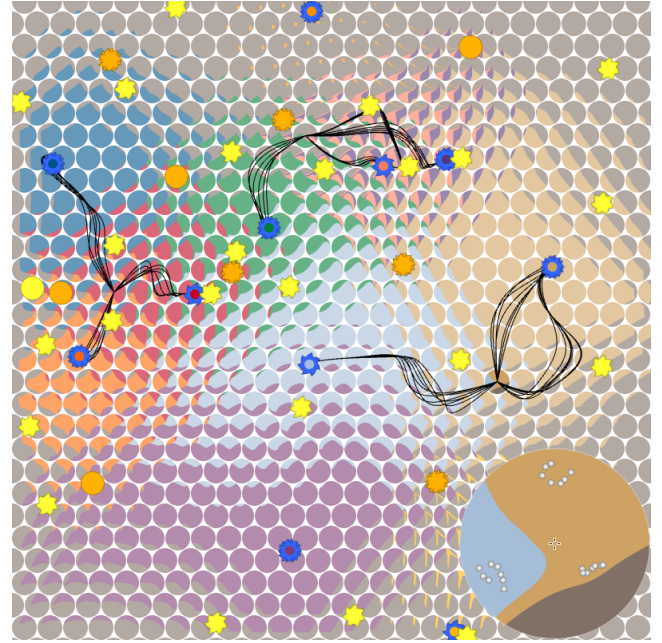
in a single visualization, hence we design a glyph that is explained in the next section.

#### 4. Glyph-based Visualization of Asymptotic Behavior

We set out to solve the following problem: For every possible location  $\mathbf{x}_0$  in the domain: show for every possible initial velocity  $\mathbf{v}_0$  (up to a certain maximal speed), which sink an inertial particle enters that is released at  $(\mathbf{x}_0, \mathbf{v}_0)$ . Essentially, we need to visualize a 4D scalar field (2D position + 2D velocity) that is computed by particle integration. Even for moderate discretizations, a full precomputation is not feasible, since that would exhaust too much computation time and memory.

**Glyph Design** To avoid this problem, we design an interactive glyph visualization that is rendered progressively and adaptive to the current view. Glyphs are frequently applied in flow visualization, e.g., in [HLNW11, HSJW14]. See the recent reports on glyph design by Ward [War08] and Borgo et al. [BKC\*13] for an overview. The idea of our glyph is as follows: Every glyph represents a single spatial position  $\mathbf{x}_0$ . The glyph is placed at its respective spatial position and we densely cover the domain with glyphs to obtain a discrete sampling of the spatial domain. With this, we limit the computation to this set of discrete spatial points, which reduces the 4D domain considerably.

Each glyph is disc-shaped and represents the  $\mathbf{v}_0$  plane. Thereby, each pixel of a glyph represents a discrete sample of the velocity domain. The angle represents the direction and the distance to the center the magnitude, up to some upper bound  $v_{max}$ , which can be adjusted interactively. The glyph concept is illustrated in Fig. 7. Each glyph covers only a certain number of pixels, which reduces the set of velocities to consider again considerably. We progressively add additional samples in every pixel for anti-aliasing purposes over the course of multiple frames, but a rather good preview is obtained with one sample already. The color of a glyph pixel encodes the sink that is reached after integration from  $\mathbf{x}_0$  with initial velocity  $\mathbf{v}_0$ . As color table, we used Kelly's colors of maximum contrast [GA10]. The size of the glyphs is specified in screen space and is interactively adjustable. In the examples throughout the paper, we set a typical glyph radius of 15–20 pixels. We layout the glyphs on hexagonal grids, which delivers the densest possible packing and reaches 90% screen space coverage. We set a glyph spacing of one pixel to make the circles easier distinguishable.



**Figure 8:** Glyphs show the asymptotic behavior of inertial particles that were released with  $d_p = 75 \mu\text{m}$ ,  $v_{max} = 10 \text{m/s}$ , and varying  $\mathbf{x}_0$  and  $\mathbf{v}_0$ . Note that sink glyphs contain the color of the corresponding  $\omega$ -basin. The user is provided detail views of selected glyphs in which seed points of inertial particle trajectories can be specified.

**Detail View** To support the exploration process, the user can select a glyph, for which a close-up is presented in the bottom-right corner of the screen. In this detail view, seed points of inertial trajectories can be added and removed by clicking on the close-up. The clicked coordinate determines the initial velocity  $\mathbf{v}_0$ , and the selected glyph itself represents the initial position  $\mathbf{x}_0$ . From there, inertial trajectories are integrated and displayed in the main view. Fig. 8 gives an example. In addition, hovering over a seed point in the detail view highlights the corresponding trajectory in the main view.

#### 5. Implementation

In order to enable interactive response in the exploration of the 4D space of possible initial conditions, we devise an efficient GPU-based computation of the glyph visualization. In the following, we elaborate on the memory layout and the progressive computation under constraint time budgets per frame.

##### 5.1. Glyph Memory Layout and Camera Navigation

We begin with the layout of the glyph data on the GPU, and its update during camera navigation.

**Data per glyph pixel.** Each glyph covers a certain number of pixels and for each of those pixels, we progressively generate subsamples that correspond to initial conditions  $(\mathbf{x}_0, \mathbf{v}_0)$ . We evaluate their color based on the classification of the sink that they reach in the limit, and progressively accumulate the evaluated colors. Thus, each pixel stores a floating point vector with four components, containing the

accumulated color (RGB) and the number of samples (A) generated so far. If a subsample is outside the glyph, the color evaluates to the background color in order to anti-alias the glyph boundary.

**Texture atlas and glyph data.** We maintain a memory pool of glyphs that is organized as a texture atlas. Thereby, each entry stores the aforementioned glyph pixel data of one glyph. In addition, we store in a buffer for each glyph its spatial position in the domain ( $\mathbf{x}_0$ ), the location of its contiguous pixel data in the texture atlas and a counter that accounts for the number of pixels that have been processed in the glyph so far. During rendering, we simply place the portion of the texture atlas that belongs to a glyph on the screen at the respective domain location. Note that the texture atlas needs to be recomputed whenever the size of the glyphs or a simulation parameter (particle diameter, particle density, viscosity or gravity) is varied by the user.

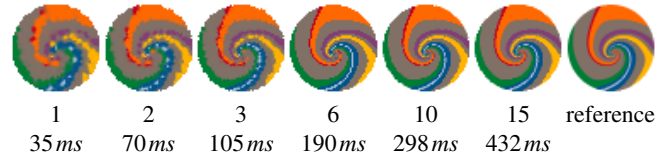
**Camera navigation.** Interaction requires a swift navigation of the camera, including panning and zooming. During panning, we move the camera pixel-wise, so that a pixel in the texture atlas always covers a full pixel on the viewport. This allows to make use of the full potential of the texture atlas, because the accumulated pixel data of the glyphs can be reused in the new view. Glyphs that leave the viewport are freed and their texture atlas space can be assigned to newly appearing glyphs. (We compute the placement of glyphs on the CPU and map the updated glyph data to the GPU.) During zooming, we maintain the size of the glyphs. In order to preserve the dense covering we scale the underlying coordinates. Zooming requires to reset the texture atlas, since the initial position  $\mathbf{x}_0$  of each glyph changes.

## 5.2. Progressive Computation

In the following, we explain how the progressive computation can be efficiently deferred over multiple frames, and how previews can be generated.

**Processing order.** As the integration-based evaluation of an initial condition is computationally expensive, only a certain number of glyph pixels can be processed each frame to maintain interactive response. The order in which the pixels of a glyph are processed is pre-determined and stored in a randomized pixel permutation vector, in order to obtain a roughly uniform sampling of the glyph. (The permutation repeats when all pixels were processed to progressively add further samples for anti-aliasing.) The procedure for each GPU thread goes as follows:

- Select the glyph (determines  $\mathbf{x}_0$ ) and its next pixel to process (lookup in pixel permutation vector).
- Take a low-discrepancy (quasi-)random sample inside the pixel to progressively anti-alias the pixel. The location is determined via Halton sequences on the CPU side and is for all pixels that are processed in an iteration the same.
- Convert the subpixel location  $\mathbf{p}_s$  in the disk to the initial velocity as  $\mathbf{v}_0 = v_{max} \cdot (\mathbf{p}_s - \mathbf{p}_d) / r_d$ , with  $\mathbf{p}_d$  being the center of the velocity disc and  $r_d$  its radius.
- Integrate an inertial trajectory until it enters a sink (4D fourth-order Runge Kutta) and lookup the color of the sink. If a trajectory



**Figure 9:** Close-up on the selected glyph in Fig. 11(a). Results of the hole filling after a certain number of frames and total passed time (below the images). Here, for a glyph with a radius of 25 pixels (2133 pixels in total), we process 400 pixels per frame.

leaves the domain or does not terminate in a critical point, the kernel evaluates to gray.

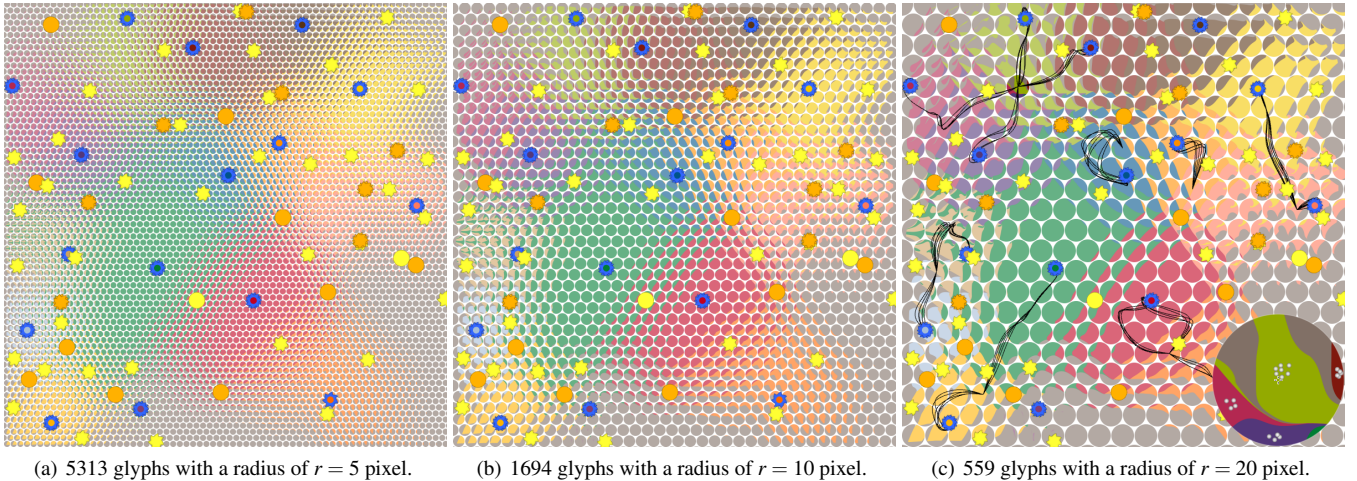
**Hole filling.** Since only a certain number of pixels can be processed each frame, we need to fill the holes to generate adequate previews. In this paper, we implemented a Shepard interpolation among the already processed pixels on the GPU. For this, we dispatch one thread group per glyph and load the already processed pixels in blocks to shared memory, iterate block by block the pending pixels and sum up the weighted distances to the processed pixels in shared memory. This method is simple and fast enough. Alternative approaches would be multi-scale push-pull, (truncated) Gaussian kernels or anisotropic diffusion (possibly heuristically in direction of the flow). A more sophisticated and general sampling and reconstruction framework was described by Frey et al. [FSME14]. Fig. 9 demonstrates results of our hole filling. In this example, an iteration took 29ms with additional 6ms on hole filling (on average). Here, after 6 frames each pixel was processed once. The following frames add further subsamples for anti-aliasing.

**Detail view.** In the detail view, the number of pixels is small enough to process each pixel in the first iteration, which leaves no holes. Afterwards, we process in each frame only a fraction of the pixels, which adds further subsamples for anti-aliasing. We found that the extra time for generating more samples in the detail view is better spent on filling holes in the main view. The time it takes to process a pixel depends on the data set (average time to reach a sink) and the integration step size. In our experience it typically takes less than 1.5 seconds to generate a sample for every glyph pixel on the viewport. Further timings are listed later in Section 6.1.

**Additional query type.** Figs. 5 and 6 illustrate an additional query type, in which we select one certain  $\mathbf{v}_0$ . In this case, no glyphs are needed and we compute for every pixel on the screen, which sink is reached in the limit, when releasing an inertial particle at the pixel. In contrast to the glyph visualization, here, a pixel depicts the spatial domain. Again, we progressively generate more samples in each pixel to fight aliasing. Technically, we split the domain into tiles of  $8 \times 8$  pixels, determine a pixel processing permutation, and update every iteration a certain number of pixels. In the hole filling step, we perform the Shepard interpolation per tile.

## 6. Results

We applied our glyph-based visualization to several synthetic and real-world data sets. The real-world data sets also serve as synthetic testing ground, in which we assume that the equations of motion



**Figure 10:** Another random data set, containing the various types of inertial critical points. Here, shown for varying glyph sizes,  $d_p = 70\mu\text{m}$  and  $v_{\text{max}} = 10\text{m/s}$ . Several inertial trajectories were interactively placed in the right image.

hold. Generally, we assume all domain sizes to be in meters and time in seconds. While the extraction and classification of inertial critical points is tailored to the underlying equations of motion, our glyph-based visualization of asymptotic behavior is independent of the equations of motion, as it is based on the abstract inertial flow map. This means, the glyph concept can be applied to any particle model.

**Borromean Rings.** Fig. 1 (left) shows a slice of the 3D BORROMEAN RINGS sequence [CB11]. It contains a simulation of a magnetic field with field lines initially interlocked in the shape of Borromean rings. Over time the rings decay and topologically reconnect to two smaller magnetic knots that drift apart. For the purpose of generating test data, we selected one slice that contains both swirling and non-swirling inertial critical points, and show the trajectories of several manually seeded inertial particles (black).

**Forced-Damped Duffing Oscillator.** The FORCED-DAMPED DUFFING oscillator is a dynamical system that experiences chaotic behavior. It served as test data for studying the separation of inertial particles in terms of inertial FTLE (separation due to spatial variation) [HS11] and FTMS (separation due to mass variation) [GT15]. Observed in phase space, its phase lines can be described as tangent curves of the 2D unsteady vector field:

$$\mathbf{u}(x, y, t) = \begin{pmatrix} y \\ x - x^3 - 0.25y + 0.4 \cos t \end{pmatrix}. \quad (11)$$

in the spatial domain  $[-2, 2]^2$ . As we observe steady vector fields, we select a particular time slice  $t = 0$ . As shown in Fig. 1 (middle), the flow contains two attracting foci, and a non-swirling saddle. Dependent on the initial position and initial velocity, an inertial particle reaches one focus or the other. In this example, we released seven inertial particles from the same initial position, but with varying initial velocity. Their initial velocity is depicted in the glyph detail view, and their trajectories are drawn in the main view. The glyph tells for the respective initial conditions the asymptotically coherent regions distinctly apart.

**Nine Critical Points.** The NINE CP data set is an analytic 2D flow that contains nine non-swirling critical points: four sinks at  $(\pm 1, \pm 1)$ , four saddles between them at  $(0, \pm 1)$  and  $(\pm 1, 0)$ , and a source at  $(0, 0)$ . The flow is given as:

$$\mathbf{u}(x, y) = \begin{pmatrix} x(1-x)(1+x) \\ y(1-y)(1+y) \end{pmatrix} \quad (12)$$

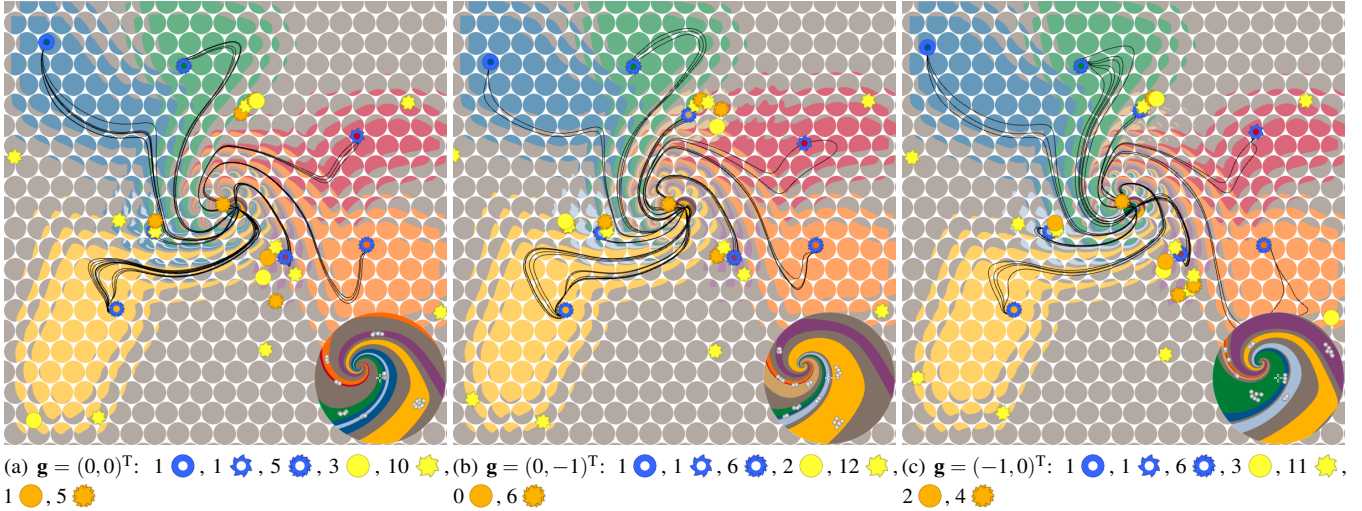
and we observe it in the domain  $[-1.5, 1.5]^2$ . For  $\mathbf{g} = \mathbf{0}$ , the inertial critical points stay at the same locations, but their type changes, as shown in Fig. 1 (right). The domain contains four  $\omega$ -basins, in which we released several inertial particles at the same location but with varying initial velocity.

**Random Flows.** We constructed several conservative flows as gradient vector fields of a random scalar field. For this, we placed random scalars in  $[-1, 1]$  on a  $12 \times 12$  (Figs. 5, 6, 8) or  $14 \times 14$  (Fig. 10) regular grid in the spatial domain  $[-1, 1]^2$ , and bilinearly interpolated values in between. The flows exhibit a variety of critical points and serve as explanatory examples, including a showcase for varying glyph sizes in Fig. 10.

**Trefoil Knot.** The TREFOIL KNOT data set is another 3D magnetic field and courtesy of Candelaresi and Brandenburg [CB11]. It contains three interlocked magnetic rings that decay over time. For demonstration purposes, we selected a 2D slice that shows swirling patterns. It contains seven swirling and non-swirling attracting critical points and their respective  $\omega$ -basins are winding in the middle of the domain around a swirling saddle. In Fig. 11, we seeded inertial trajectories at a glyph near the aforementioned swirling saddle, and examine the influence of the gravity vector. We selected initial velocities in the detail view of the selected glyph, which eventually reach their respective attracting node or focus.

**Benzene Molecule.** The BENZENE data set contains the central z-slice of the analytic approximation of a 3D potential field of a benzene molecule [ZSH96]. The vector field is shown in Fig. 12 for varying  $d_p$  and contains a large number of symmetrical inertial





**Figure 11:** In the TREFOIL data set, spiral patterns occur in the glyphs, showing that the magnitude of an initial velocity has great impact. Here, shown for varying gravity, with  $d_p = 70\mu\text{m}$  and  $v_{\max} = 5\text{m/s}$ . Note that a change in gravity causes critical points to move, change their type and appear/disappear.

Data set	inertial critical points						glyph parameters			computation time			
	# attr.		# 1/3 sad.		# 2/2 sad.		# glyphs on viewport	# pixels per glyph	# glyph pixels per iteration	CP extr. [in ms]	iteration [in ms]	hole filling [in ms]	
Borromean, Fig. 1	1	0	1	1	3	0	2	907	801	50	0.49	50.6	5.36
Duffing, Fig. 1	0	0	2	1	0	0	0	907	801	32	0.47	51.7	6.37
Nine CP, Fig. 1	4	0	0	4	0	1	0	907	801	200	0.29	25.1	4.11
Random, Fig. 10(c)	0	1	15	2	38	10	8	559	1385	500	0.72	27.7	6.13
Trefoil, Fig. 11(a)	1	1	5	3	10	1	5	559	1385	400	1.28	27.6	5.90
Benzene, Fig. 12(b)	24	6	0	34	8	13	0	1655	373	373	0.74	20.3	—

**Table 2:** This table summarizes the extraction results of inertial critical points, reports the glyph parameters that we set (number of glyphs on the viewport, number of pixels per glyph, and number of pixels per glyph that are processed each frame), and lists the computation timings for the extraction of inertial critical points, the time it takes to process one frame and the maximal time needed for the hole filling.

critical points. For reference, we placed inertial particle trajectories in the domain, and it can be seen that the inertia has great influence on their paths. With increasing diameter, more inertial critical points exhibit swirling behavior, and the stronger is the impact of the initial velocity on the critical point that is reached in the limit.

### 6.1. Performance

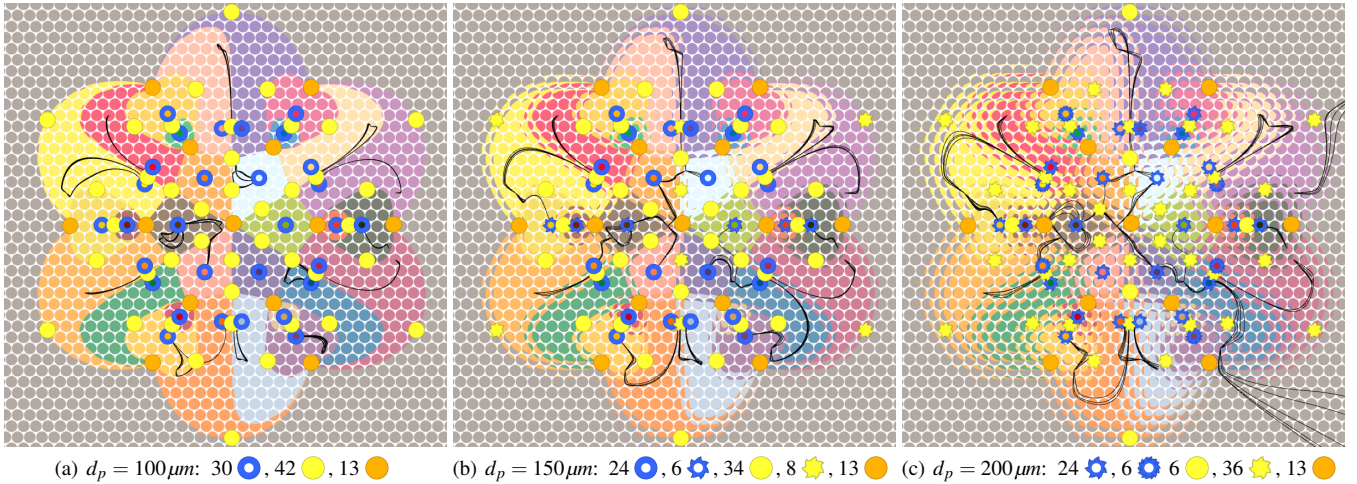
We used an Intel Core i7-2600K CPU with 3.4 GHz and 24 GB RAM, and an Nvidia GeForce GTX 970 GPU with 4 GB VRAM. All results were rendered at a resolution of  $800 \times 800$  pixels with  $4 \times$  MSAA. Table 2 reports the numbers of extracted inertial critical points for our test data sets, shows the glyph parameters that we set and lists the computation timings. The glyph parameters include the number of glyphs on the viewport and the number of pixels per glyph. Naturally, the larger the glyphs the less glyphs fit on the viewport. The third entry in the glyph parameters column reports the number of glyph pixels that we process each frame per glyph. This number balances the quality of the preview for performance and was chosen so that roughly 20 – 50 fps were reached. Its setting depends on the data set, mainly on the average time it takes an inertial particle to enter an attracting node or leave the domain,

and the chosen numerical step size. In the FORCED DUFFING for example, inertial particles follow a rather long path, compared to the BENZENE data set, in which each glyph pixel could be processed every frame. Here, hole filling was not even necessary. Hole filling takes place until every glyph pixel was processed at least once. In the other data sets, this was achieved after 3 – 25 frames.

The extraction of inertial critical points takes almost the same time as for the massless case, since only the field  $r\mathbf{g}$  is added to  $\mathbf{u}(\mathbf{x})$  prior to extraction. The extraction is a root finding problem, for which we used a recursive subdivision of the spatial domain up to an accuracy of  $1 \cdot 10^{-10}$ . The hole filling cost grows with the number of already processed pixels from close to zero up to the value given in the table, as the number of pixels grows that are taken into account during Shepard interpolation. Reconstruction kernels with local support can reduce this cost.

### 6.2. Limitations

We encoded the mapping of sink glyphs to their respective  $\omega$ -basins by color. The number of easily distinguishable colors is limited and our current table only supports up to 20 colors (excluding black and



**Figure 12:** The BENZENE data set, contains a large number of inertial critical points. Here, shown for varying  $d_p$  and  $v_{max} = 20\text{m/s}$ . The inertial trajectories (black) were released in all three experiments with the same initial position and initial velocity.

white) [GA10]. For the BENZENE data set in Fig. 12, we repeated the colors at 30% more brightness. Alternatively, textures could be used to generate further distinguishable patterns. Currently, we use a random permutation to determine the order in which the pixels of a glyph are processed. Frey et al. [FSME14] used a multi-resolution capacity-constrained sample distribution, which could be the basis for a more evenly distributed pixel processing permutation. Flows with thousands or critical points are generally perceptually difficult for all topological methods. Topological simplification addresses this problem, see Chen et al. [CML\*07] and the references therein.

## 7. Conclusions

The paper presents – to the best of our knowledge – the first approach to a topology-based visualization of inertial flows. We conducted a full classification of critical points in 2D steady inertial flow. We studied the relation between eigenvalues of the massless 2D Jacobian and the inertial 4D Jacobian. Among others, we made the observation that in inertial flow, sources cannot exist. This explains the strong repelling behavior of backward integration from a topological point of view, and is consistent with the findings in related CFD literature [HS08]. In order to study the asymptotic behavior of inertial particles, we developed a glyph that encodes for both varying initial position and initial velocity, which sink is reached in the limit. We see potential use of this in general dynamical systems theory. The motion of inertial particles is a subclass of this large area, and studying topologically the possible outcome of initial conditions in general dynamical systems (ODEs) seems very useful.

In future work, we look forward to extensions to 3D, i.e., a full classification of the 6D critical points and the visualization of their separation manifolds. The 3D case poses significant additional challenges concerning the rendering and visual representation of the segmentation. As for an extension to unsteady flows, the same issues arise as in the traditional massless case: due to the limited time range, asymptotic behavior can usually not be observed. As an alternative, finite-time approaches have been proposed, both in the massless

and the inertial case. Aside from extensions to more general particle models, we also look forward to adaptations that allow for spatially and temporally varying response times. Both will allow to reach out to further applications. The glyph design can be evaluated by the principles of Borgo et al. [BKC\*13]. Finally, there are many more concepts from traditional vector field topology that can be extended to the inertial case, including saddle connectors, boundary switches, closed orbits etc. The extraction of some of these features could be quite challenging, since backward integration toward critical points is practically not possible.

## Acknowledgements

We wish to thank our anonymous reviewers for their helpful comments. This work was supported by DFG grant no. TH 692/13-1.

## Appendix

To show the condition for  $Re(f_{j,2}) = 0$  in Eq. (8), we compute

$$Re(f_{j,2}) = Re\left(\frac{-1 + \sqrt{1 + 4r(Re(e_j) + iIm(e_j))}}{2r}\right) \quad (13)$$

$$= Re\left(\frac{-1 + \sqrt{1 + 4r(-rIm(e_j)^2 + iIm(e_j))}}{2r}\right) \quad (14)$$

$$= Re\left(\frac{-1 + \sqrt{-(i - 2rIm(e_j))^2}}{2r}\right) \quad (15)$$

$$= Re(iIm(e_j)) = 0 \quad (16)$$

where the step from (13) to (14) inserts the right-hand side of Eq. (8), the step from (14) to (15) factors the radicand and the step from (15) to (16) follows by computation of the square root of a complex number  $z$  as  $\sqrt{z} = \sqrt{|z|} \frac{z+|z|}{|z+|z||}$  and from subsequent simplification.

## References

- [BBC\*09] BENZI R., BIFERALE L., CALZAVARINI E., LOHSE D., TOSCHI F.: Velocity-gradient statistics along particle trajectories in turbulent flows: The refined similarity hypothesis in the Lagrangian frame. *Phys. Rev. E* 80 (Dec 2009), 066318. 2
- [BBC\*11] BEC J., BIFERALE L., CENCINI M., LANOTTE A. S., TOSCHI F.: Spatial and velocity statistics of inertial particles in turbulent flows. *Journal of Physics: Conference Series* 333, 1 (2011), 012003. 2
- [BE10] BALACHANDAR S., EATON J. K.: Turbulent dispersed multiphase flow. *Annual Review of Fluid Mechanics* 42, 1 (2010), 111–133. 2
- [BH02] BROWN J. K. M., HOVMØLLER M. S.: Aerial dispersal of pathogens on the global and continental scales and its impact on plant disease. *Science* 297, 5581 (2002), 537–541. 2
- [BKC\*13] BORGIO R., KEHRER J., CHUNG D. H. S., MAGUIRE E., LARAMEE R. S., HAUSER H., WARD M., CHEN M.: Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics 2013 – State of the Art Reports (STARs)* (2013), pp. 39–63. 6, 10
- [Bor11] BORDÁS R.: *Optical measurements in disperse two-phase flows: Application to rain formation in cumulus clouds*. PhD thesis, University of Magdeburg, 2011. 2
- [BZ89] BÜCHNER J., ZELENYI L. M.: Regular and chaotic charged particle motion in magnetotail-like field reversals: 1. Basic theory of trapped motion. *Journal of Geophysical Research: Space Physics* (1978–2012) 94, A9 (1989), 11821–11842. 2
- [CB11] CANDELARESI S., BRANDENBURG A.: Decay of helical and nonhelical magnetic knots. *Phys. Rev. E* 84 (2011), 016406. 8
- [CGP\*10] CASCIOLA C. M., GUALTIERI P., PICANO F., SARDINA G., TROLANI G.: Dynamics of inertial particles in free jets. *Physica Scripta* 2010, T142 (2010), 014001. 2
- [CML\*07] CHEN G., MISCHAIKOW K., LARAMEE R. S., PILARCZYK P., ZHANG E.: Vector field editing and periodic orbit extraction using morse decomposition. *IEEE Transactions on Visualization and Computer Graphics* 13, 4 (2007), 769–785. 4, 10
- [CST98] CROWE C., SOMMERFELD M., TSUJI Y.: *Multiphase Flows with Droplets and Particles*. CRC Press, 1998. 2
- [dLvL99] DE LEEUW W., VAN LIERE R.: Collapsing flow topology using area metrics. In *Proceedings of the Conference on Visualization* (1999), VIS '99, pp. 349–354. 4
- [FH15] FARAZMAND M., HALLER G.: The Maxey–Riley equation: Existence, uniqueness and regularity of solutions. *Nonlinear Analysis: Real World Applications* 22 (2015), 98–106. 2
- [FSME14] FREY S., SADLO F., MA K.-L., ERTL T.: Interactive progressive visualization with space-time error control. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization)* 20, 12 (2014), 2397–2406. 7, 10
- [GA10] GREEN-ARMYTAGÉ P.: A colour alphabet and the limits of colour coding. *JAIC-Journal of the International Colour Association* 5 (2010), 6, 10
- [GKKT13] GÜNTHER T., KUHN A., KUTZ B., THEISEL H.: Mass-dependent integral curves in unsteady vector fields. *Computer Graphics Forum (Proc. EuroVis)* 32, 3 (2013), 211–220. 3, 4
- [GLL91] GLOBUS A., LEVIT C., LASINSKI T.: A tool for visualizing the topology of three-dimensional vector fields. In *Proc. IEEE Visualization* (1991), pp. 33–40. 4
- [GT14] GÜNTHER T., THEISEL H.: Vortex cores of inertial particles. *IEEE Trans. on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization)* 20, 12 (2014), 2535–2544. 2, 3, 4
- [GT15] GÜNTHER T., THEISEL H.: Finite-time mass separation for comparative visualizations of inertial particles. *Computer Graphics Forum (Proc. EuroVis)* 34, 3 (2015), 471–480. 2, 4, 8
- [Hal01] HALLER G.: Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena* 149, 4 (Mar. 2001), 248–277. 3, 4
- [HH89] HELMAN J. L., HESSELINK L.: Representation and display of vector field topology in fluid flow data sets. *Computer* 22, 8 (1989), 27–36. 4
- [HH91] HELMAN J., HESSELINK L.: Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications* 11 (May 1991), 36–46. 4
- [HLNW11] HLAWATSCH M., LEUBE P. C., NOWAK W., WEISKOPF D.: Flow radar glyphs – static visualization of unsteady flow with uncertainty. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization)* 17, 12 (2011), 1949–1958. 6
- [HS08] HALLER G., SAPSIS T.: Where do inertial particles go in fluid flows? *Physica D Nonlinear Phenomena* 237 (May 2008), 573–583. 1, 2, 3, 5, 10
- [HS11] HALLER G., SAPSIS T.: Lagrangian coherent structures and the smallest finite-time Lyapunov exponent. *Chaos* 21, 2 (2011), 023115. 8
- [HSJW14] HLAWATSCH M., SADLO F., JANG H., WEISKOPF D.: Pathline glyphs. *Computer Graphics Forum (Proc. Eurographics)* 33, 2 (2014), 497–506. 6
- [KGRK14] KUTZ B. M., GÜNTHER T., RUMPF A., KUHN A.: Numerical examination of a model rotor in brownout conditions. In *Proceedings of the American Helicopter Society* (May 2014), no. AHS2014-000343 in AHS 70th Annual Forum. 1, 2
- [LDG98] LÖFFELMANN H., DOLEISCH H., GRÖLLER E.: Visualizing dynamical systems near critical points. In *In Spring Conference on Computer Graphics and its Applications* (1998), pp. 175–184. 4
- [LHZP07] LARAMEE R., HAUSER H., ZHAO L., POST F.: Topology-based flow visualization, the state of the art. In *Topology-based Methods in Visualization*, Hauser H., Hagen H., Theisel H., (Eds.), Mathematics and Visualization. Springer Berlin Heidelberg, 2007, pp. 1–19. 4
- [LRR00] LODHA S., RENTERIA J., ROSKIN K.: Topology preserving compression of 2D vector fields. In *In Proc. IEEE Visualization* (2000), pp. 343–350. 4
- [MBS\*04] MAHROUS K., BENNETT J., SCHEUERMANN G., HAMANN B., JOY K., ET AL.: Topological segmentation in three-dimensional vector fields. *IEEE Transactions on Visualization and Computer Graphics* 10, 2 (2004), 198–205. 4
- [MBZ06] MOGRABI E., BAR-ZIV E.: On the asymptotic solution of the Maxey–Riley equation. *Physics of Fluids* 18, 5 (2006). 3
- [MR83] MAXEY M. R., RILEY J. J.: Equation of motion for a small rigid sphere in a nonuniform flow. *Physics of Fluids* 26, 4 (1983), 883–889. 2
- [MTP\*02] MACPHEE A. G., TATE M. W., POWELL C. F., YUE Y., RENZI M. J., ERCAN A., NARAYANAN S., FONTES E., WALTHER J., SCHALLER J., ET AL.: X-ray imaging of shock waves generated by high-pressure fuel sprays. *Science* 295, 5558 (2002), 1261–1263. 2
- [OOG08] OUELLETTE N. T., O'MALLEY P. J. J., GOLLUB J. P.: Transport of finite-sized particles in chaotic flow. *Phys. Rev. Lett.* 101 (2008), 174504. 1, 2
- [PD09] PENG J., DABIRI J. O.: Transport of inertial particles by Lagrangian coherent structures: Application to predator–prey interaction in jellyfish feeding. *Journal of Fluid Mechanics* 623 (3 2009), 75–84. 2, 4
- [PPF\*11] POBITZER A., PEIKERT R., FUCHS R., SCHINDLER B., KUHN A., THEISEL H., MATKOVIC K., HAUSER H.: The state of the art in topology-based visualization of unsteady flow. *Computer Graphics Forum* 30, 6 (2011), 1789–1811. 4
- [PSGC11a] PICANO F., SARDINA G., GUALTIERI P., CASCIOLA C.: DNS of a free turbulent jet laden with small inertial particles. In *Direct and Large-Eddy Simulation VIII*, Kuerten H., Geurts B., Armenio V., Fröhlich J., (Eds.), vol. 15 of *ERCOFTAC Series*. Springer Netherlands, 2011, pp. 189–194. 2

- [PSGC11b] PICANO F., SARDINA G., GUALTIERI P., CASCIOLA C. M.: Particle-laden jets: Particle distribution and back-reaction on the flow. *Journal of Physics: Conference Series* 318, 5 (2011), 052018. 2
- [RSBE01] ROETTGER S., SCHULZ M., BARTELHEIMER W., ERTL T.: Automotive soiling simulation based on massive particle tracing. In *Data Visualization 2001*, Eurographics. Springer Vienna, 2001, pp. 309–317. 1, 3
- [SGL10] SYAL M., GOVINDARAJAN B., LEISHMAN J. G.: Mesoscale sediment tracking methodology to analyze brownout cloud developments. In *Proceedings of the American Helicopter Society, 66th Annual Forum* (2010). 2
- [SH09] SAPSIS T. P., HALLER G.: Inertial particle dynamics in a hurricane. *Journal of the Atmospheric Sciences* (2009). 3, 4
- [SKMR98] SCHEUERMANN G., KRUGER H., MENZEL M., ROCKWOOD A.: Visualizing nonlinear vector field topology. *IEEE Transactions on Visualization and Computer Graphics* 4, 2 (1998), 109–116. 4
- [SL99] SHAO Y., LI A.: Numerical modelling of saltation in the atmospheric surface layer. *Boundary-Layer Meteorology* 91 (1999), 199–225. 1
- [SLM05] SHADDEN S., LEKIEN F., MARSDEN J.: Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena* 212, 3-4 (2005), 271–304. 3
- [SPH11] SAPSIS T., PENG J., HALLER G.: Instabilities on prey dynamics in jellyfish feeding. *Bull Math Biol.* 73, 8 (2011), 1841–1856. 3
- [The02] THEISEL H.: Designing 2D vector fields of arbitrary topology. *Computer Graphics Forum (Proc. Eurographics)* 21, 3 (2002), 595–604. 4
- [TRS03] THEISEL H., RÖSSL C., SEIDEL H.-P.: Compression of 2D vector fields under guaranteed topology preservation. *Computer Graphics Forum (Proc. Eurographics)* 22, 3 (2003), 333–342. 4
- [TWHS03] THEISEL H., WEINKAUF T., HEGE H.-C., SEIDEL H.-P.: Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields. In *Proc. IEEE Visualization* (2003), pp. 225–232. 4
- [War08] WARD M. O.: Multivariate data glyphs: Principles and practice. In *Handbook of Data Visualization*, Springer Handbooks Comp. Statistics. Springer Berlin Heidelberg, 2008, pp. 179–198. 6
- [Wei08] WEINKAUF T.: *Extraction of Topological Structures in 2D and 3D Vector Fields*. PhD thesis, University Magdeburg, 2008. 4, 5
- [WJE01] WESTERMANN R., JOHNSON C., ERTL T.: Topology-preserving smoothing of vector fields. *IEEE Transactions on Visualization and Computer Graphics* 7, 3 (2001), 222–229. 4
- [WS01] WISCHGOLL T., SCHEUERMANN G.: Detection and visualization of closed streamlines in planar flows. *IEEE Transactions on Visualization and Computer Graphics* 7, 2 (2001), 165–172. 4
- [WTHS04a] WEINKAUF T., THEISEL H., HEGE H.-C., SEIDEL H.-P.: Boundary switch connectors for topological visualization of complex 3D vector fields. In *VisSym* (2004), pp. 183–192. 4
- [WTHS04b] WEINKAUF T., THEISEL H., HEGE H.-C., SEIDEL H.-P.: Topological construction and visualization of higher order 3D vector fields. *Computer Graphics Forum (Proc. Eurographics)* 23, 3 (2004), 469–478. 4
- [WTHS06] WEINKAUF T., THEISEL H., HEGE H.-C., SEIDEL H.-P.: Topological structures in two-parameter-dependent 2D vector fields. *Computer Graphics Forum (Proc. Eurographics)* 25, 3 (2006), 607–616. 4
- [ZSH96] ZÖCKLER M., STALLING D., HEGE H.-C.: Interactive visualization of 3D vector fields using illuminated stream lines. In *IEEE Visualization* (1996), pp. 107–113. 8