Recirculation Surfaces for Flow Visualization

Thomas Wilde, Christian Rössl, Holger Theisel

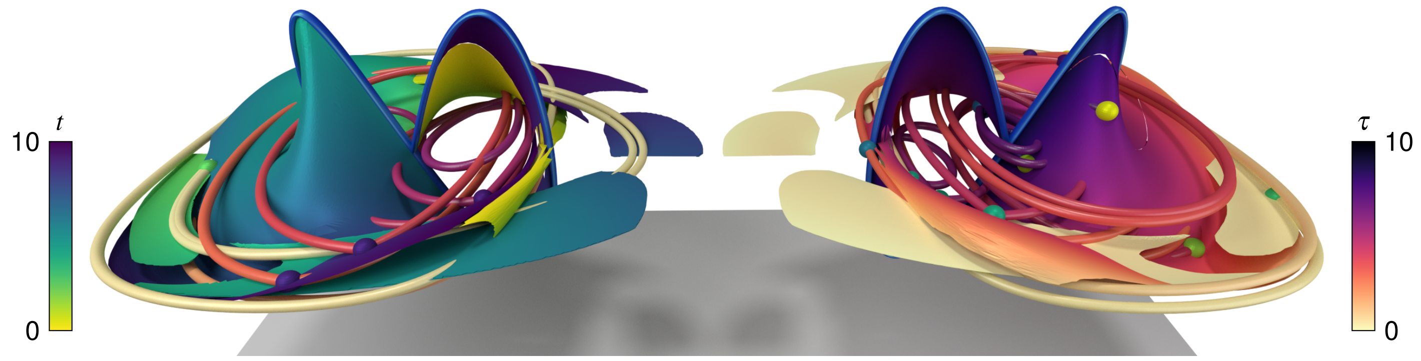

Fig. 1. Recirculation surfaces and recirculating path lines in the 3D Double Gyre data set. The color maps of the surfaces indicate start time t (left) or integration time τ (right). The tubes show a random selection of recirculating path lines with τ encoded as constant color; the spheres show their start points on the recirculation surface with color-coded t. The boundary curves of the surfaces are defined by the paths of critical points, which are depicted as blue tubes.

Abstract— We present a formal approach to the visual analysis of recirculation in flows by introducing recirculation surfaces for 3D unsteady flow fields. Recirculation surfaces are the loci where massless particle integration returns to its starting point after some variable, finite integration. We give a rigorous definition of recirculation surfaces as 2-manifolds embedded in 5D space and study their properties. Based on this we construct an algorithm for their extraction, which searches for intersections of a recirculation surface with lines defined in 3D. This reduces the problem to a repeated search for critical points in 3D vector fields. We provide a uniform sampling of the search space paired with a surface reconstruction and visualize results. This way, we present the first algorithm for a comprehensive feature extraction in the 5D flow map of a 3D flow. The problem of finding isolated closed orbits in steady vector fields occurs as a special case of recirculation surfaces. This includes isolated closed orbits with saddle behavior. We show recirculation surfaces for a number of artificial and real flow data sets.

Index Terms—Flow visualization, recirculation, unsteady flow

1 Introduction

Recirculation is a phenomenon that influences many effects in 3D unsteady flows. A number of approaches study recirculation itself or other phenomena in the presence of recirculation. Recirculation has been treated in rather different communities, ranging from biology, chemistry and physics to flow visualization.

There seems to be a common intuitive understanding about what recirculation is: The term may refer to locations where moving particles return to their starting locations after a certain time. Alternatively, recirculation describes regions of particles that do not leave a certain region (but may move within this region). Recirculation can also describe areas where particles move "backwards" or "against the main flow direction", which requires a proper definition of what "backwards" means. For steady flows, recirculation is related to closed orbits. Surprisingly, there seems to be no unique common definition of the concept of recirculation — a property that recirculation shares, e.g., with the concept of a vortex.

In this paper, we start with a rigorous definition of recirculation: We introduce *recirculation surfaces* as features in 3D unsteady flows. Then we study their properties and provide a stable numerical algorithm for their extraction. We consider particularly the special case of steady flows in which our approach provides a new extraction algorithm for isolated closed orbits. Finally, we demonstrate recirculation surfaces for a number of artificial and real flow examples.

The main contribution of this paper consists in the definition of recirculation surfaces and the construction of an algorithm for their extraction.

2 Related Work

The analysis and visualization of 3D flow data is an important and active area of research as shown, e.g., by recent contributions by Chern et al. [8] and Günther et al. [13]. Both approaches provide insight into given data by recovering and visualizing different flow phenomena. In this paper, we consider another, different property of flows: recirculation.

Recirculation is an important phenomenon that has been studied in various application domains. This section provides a brief review of related work on recirculation as well as on algorithms that our approach is based upon.

Recirculation in Physics and Computational Fluid Mechanics.

In physics and computational fluid mechanics, recirculation can be important for controlling the flow in certain manners. For instance, Gautier et al. [12] use a machine learning approach to reduce the recirculation zone of a backward- facing step flow, and Debien et al. [9]

- *Thomas Wilde is with Visual Computing group at the University of Magdeburg. E-mail: thomas@isg.cs.uni-magdeburg.de.*
- *Christian Rössl is with Visual Computing group at the University of Magdeburg. E-mail: roessl@isg.cs.uni-magdeburg.de.*
- *Holger Theisel is with Visual Computing group at the University of Magdeburg. E-mail: theisel@ovgu.de.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

apply genetic programming for a flow of the turbulent boundary layer downstream from a sharp edge ramp. A recent overview of turbulence control is given by Brunton and Noack [3]. Tadmor and Banaszuk [32] track a reference orbit in a combustor recirculation zone to control vortex motion. Chen et al. and Laramee et al. [6,19] identify recirculating flow behavior as a feature for describing combustion processes in an engine. Noack et al. [23] optimize coarse-scale mixing in a recirculation zone with a simple vortex model by Suh [31]. Shadden et al. [29] analyze aortic blood flows for predictions stenosis using Lagrangian Coherent Structures (LCS) that separate regions of recirculation flow. Mittasch et al. [22] introduce a method for influencing cytoplasmic streaming and identify intracellular flows that show recirculating behavior.

Recirculation in Flow Visualization. Recirculation and related phenomena have been studied in flow visualization. Peikert and Sadlo [25] study vortex rings in recirculating flows. Sanderson et al. [27] extract orbits for a magnetic field by exploiting a natural Poincaré section.

Isolated closed orbits for steady vector fields. For steady vector fields, recirculation is related to closed orbits. There exists a number of methods for finding closed orbits in 2D fields. Wischgoll and Scheuermann [40] provide the first approach to the detection of closed stream lines in planar flows. Theisel et al. [34] compute closed orbits in 2D by intersecting 3D stream surfaces. Morse decomposition provides a combinatorial approach to vector field topology and is used to detect closed stream lines in a series of articles by Chen et al. [6,7].

Existing 2D techniques make use of the fact that closed orbits have either a source or a sink behavior. Their extension to 3D is straightforward as long as the closed orbits show source or sink behavior [26]. In 3D, however, a closed orbit can also have a saddle behavior, i.e., almost all stream lines in the neighborhood of the closed orbit move away under both forward and backward integration [1,24]. For such orbits, the known 2D techniques do not carry over. The only existing solution so far is by Kasten et al. [15]: The intersections of the closed orbit with a plane are obtained by computing the intersection of finite-time Lyapunov exponent (FTLE) [14] ridges in forward and backward integration.

Extraction of isolated critical points. 3D steady vector fields generally have isolated critical points. For a piecewise linear field, they can be computed analytically as a solution system of three linear equations. For piecewise trilinear vector fields, an analytic solution does not exist. Mann and Rockwood [20] use an octree-like approach. An approach specifically designed for electric fields defined by a set of point charges was presented by Max and Weinkauf [21]. Weinkauf [37] uses a recursive algorithm: the values at all vertices spanning a trilinear cell are examined, and if all values shared the same sign a critical point cannot occur in the cell, and the recursion stops. Otherwise the cell is subdivided, and each subcell is examined recursively until a certain threshold is reached, i.e., the location of the critical point is bounded within the smallest subcell. This algorithm typically generates multiple "candidates" for the same root, which is resolved by a local clustering.

Tracking critical points. There exist several methods for tracking critical points. Tricoche et al. [35] consider piecewise linear 2D vector fields and compute and connect the critical points on the faces of a prism cell structure obtained by the tetrahedral grid. Garth et al. [11] give an extension to 3D flows defined piecewise on a tetrahedral partition with linear interpolation in both space and time. An adaptation to quadrilinear vector fields, i.e., linear interpolation in each spatial dimension and time, is straightforward. In this case, the restriction to cell boundaries of 4D hypercubes gives 3D trilinear vector fields. Then the extraction of critical points on hypercube faces can proceed as described above.

An alternative class of algorithms for tracking critical points uses feature flow fields [33], i.e., they construct space-time vector fields with tangent curves corresponding to the searched paths of the critical points. Klein et al. [17] use this approach to track critical points in scale space. Weinkauf et al. present an out-of-core version [38] and a stabilized version [39] of feature flow fields.

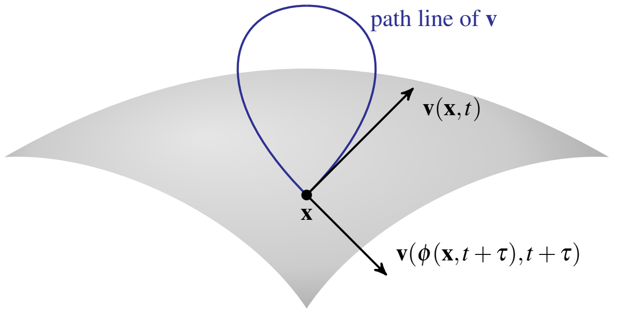
path line of $\mathbf{v}$
$\mathbf{v}(\mathbf{x},t)$
$\mathbf{x}$
$\mathbf{v}(\phi(\mathbf{x},t+\tau),t+\tau)$

Fig. 2. Illustration of a recirculation surface $\bar{\bar{Y}}$ (or Y in 3D): Starting a path line from a surface point $\mathbf{x}$ returns to $\mathbf{x}$ after a finite integration time.

3 Definition of recirculation surfaces

Throughout this paper we use the following notation and conventions. We write small bold letters for points and vectors, in particular, $\mathbf{0}$ denotes the null vector. We write capital bold letters for matrices. The capital letter Y denotes implicitly defined recirculation surfaces.

Given is the 3D time-dependent vector field $\mathbf{v}(\mathbf{x},t)$ such that $\mathbf{v} \in C^1$ is at least once continuously differentiable. The term *path line* refers to the trajectory of a massless particle that is advected under influence of $\mathbf{v}$ for a given start time t and time interval τ. A point $(\mathbf{x},t,\tau)$ on a path line has the vector $\mathbf{v}(\mathbf{x},t+\tau)$ as (spatial) tangent. In contrast, *stream lines* are tangent to the underlying vector field $\mathbf{v}(\mathbf{x},t_0)$ for a fixed time $t_0 = \text{const}$. If the vector field $\mathbf{v}$ vanishes at an isolated point $\mathbf{x}_0$, i.e., $\mathbf{v}(\mathbf{x}_0,t) = \mathbf{0}$, then $\mathbf{x}_0$ is called a *critical point*.

We use the concept of the flow map $\phi(\mathbf{x},t,\tau)$, which yields the location of a massless particle released at $(\mathbf{x},t)$ after integration over a period of time τ. Further, $\nabla\phi = \nabla_{\mathbf{x}}\phi$ denotes the spatial gradient of ϕ, and $\phi_t = \frac{\partial\phi}{\partial t}$ is its time derivative. In order to search for recirculation, we need to explore a 5D space $(\mathbf{x},t,\tau)^{\mathrm{T}}$ where $\mathbf{x}$ is the spatial location, t is the starting time and τ the time of a particle integration. The additional dimensions of points or vectors in this 5D space are indicated by double bars, e.g., $\bar{\bar{\mathbf{x}}} = (\mathbf{x},t,\tau)^{\mathrm{T}}$ or $\bar{\bar{\mathbf{v}}} \in \mathbb{R}^5$. Similarly, we distinguish between $Y \subset \mathbb{R}^3$ and $\bar{\bar{Y}} \subset \mathbb{R}^5$.

We define the recirculation surface implicitly as the set of all 5D points, for which the flow map gives the identity, i.e., the loci of particles that return to their starting point after a finite integration time.

Definition. *A* recirculation surface *is defined as the 5D surface*

$$\bar{\bar{Y}} = \left\{ (\mathbf{x},t,\tau)^{\mathrm{T}} \in \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R} : \frac{\phi(\mathbf{x},t,\tau) - \mathbf{x}}{\tau} = \mathbf{0} \wedge \tau \neq 0 \right\} .$$

The projection Y of $\bar{\bar{Y}}$ to the 3D space is given by

$$Y = \left\{ \mathbf{x} \in \mathbb{R}^3 : (\mathbf{x},t,\tau)^{\mathrm{T}} \in \bar{\bar{Y}} \right\} .$$

Figure 2 gives an illustration of a recirculation surface. Note that the dependence on the denominator τ is necessary to treat the case $\tau \to 0$ properly: Without the term $\frac{1}{\tau}$, any point in the 5D domain would be part of a recirculation "surface" for vanishing integration time $\tau \to 0$, which is clearly undesired. With the term included, there exists a well-defined limit surface for $\tau \to 0$ as will be explained in Section 3.2.

3.1 Properties

This and the following sections require some concepts like tangent space from differential geometry of manifolds (see, e.g., [10]) as well as concepts from linear algebra (e.g., null space and its basis). Furthermore, we apply multivariate calculus and provide nontrivial derivations in the appendix.

We use the abbreviations

$$\phi := \phi(\mathbf{x},t,\tau)\,, \quad \mathbf{v}_2 := \mathbf{v}(\phi,t+\tau) = \nabla\phi\cdot\mathbf{v}+\phi_t\,, \quad \mathbf{G} := \nabla\phi - \mathbf{I}\,.$$

Note that the definition of $\mathbf{v}_2$ uses a property of the flow map that is proven in the appendix. Furthermore, the definition of symbols $\mathbf{v}_2$ and $\mathbf{G}$ will be used literally as "abbreviations" to facilitate expressions and comparing coefficients/factors in expressions.
We consider the vector function

$$\mathbf{d}(\mathbf{x},t,\tau) = \frac{\phi-\mathbf{x}}{\tau} \tag{1}$$

whose zeros define the recirculation surface.

To study its properties, we consider its (spatial) directional derivative in direction $\mathbf{r}$, which is obtained using elementary differentiation rules:

$$\frac{\mathbf{dd}}{\mathbf{dr}} = \nabla\mathbf{d}\cdot\mathbf{r} = \frac{\nabla\phi\cdot\mathbf{r}-\mathbf{r}}{\tau} = \frac{\mathbf{G}\,\mathbf{r}}{\tau}\,.$$

Furthermore, the partial derivatives of $\mathbf{d}$ w.r.t. times t and τ are

$$\frac{\partial\mathbf{d}}{\partial t} = \frac{\phi_t}{\tau} \quad\text{and}\quad \frac{\partial\mathbf{d}}{\partial\tau} = \frac{\mathbf{v}_2-\mathbf{d}}{\tau}\,.$$

This allows us to assemble the 5D gradient $\nabla\mathbf{d}$ as the matrix

$$\mathbf{M} \;:=\; \nabla\mathbf{d} \;=\; \frac{1}{\tau}\Big[\;\mathbf{G}\;\big|\;\phi_t\;\big|\;\mathbf{v}_2-\mathbf{d}\;\Big] \in \mathbb{R}^{3\times 5}\,.$$

The recirculation surface $\bar{\bar{Y}}$ is defined implicitly as a point set. We study the neighborhood of a point $\bar{\bar{\mathbf{x}}}_0 \in \bar{\bar{Y}}$ by constructing the *tangent space* $\{\bar{\bar{\mathbf{x}}} : \mathbf{M}(\bar{\bar{\mathbf{x}}}-\bar{\bar{\mathbf{x}}}_0) = \mathbf{0}\}$, which provides a local linear approximation to $\bar{\bar{Y}}$. The null space of $\mathbf{M}$ provides a basis for this affine subspace of $\mathbb{R}^5$. We generally expect $\mathbf{M}$ to have full rank, i.e., rank$(\mathbf{M}) = 3$. Then the null space has dimension 5-3=2, and $\bar{\bar{Y}}$ is indeed locally a 2-manifold, which justifies the term recirculation *surface*. Local configurations where $\mathbf{M}$ has a smaller rank are structurally unstable: with a small perturbation, like adding some noise, all singular values will become nonzero and $\mathbf{M}$ is "regularized" to have full rank. Note that the dimension of the kernel can only grow with a rank deficit of $\mathbf{M}$, in particular, the dimension cannot become less than 2.

We remark that the QR-decomposition $\mathbf{M}^{\mathrm{T}} = \mathbf{QR}$ provides an efficient way to compute an orthonormal basis of the null space of $\mathbf{M}$: The first two columns of the orthogonal matrix $\mathbf{Q}$ span the null space and hence the tangent space of $\bar{\bar{Y}}$ at $\bar{\bar{\mathbf{x}}}_0$. While this is more efficient than a singular value decomposition of $\mathbf{M}$ (or likewise a spectral decomposition of $\mathbf{M}^{\mathrm{T}}\mathbf{M}$), it assumes full rank whereas singular values (or eigenvalues) would reveal a rank deficit directly.

The norm of the vector function $\mathbf{d}$ gives a pseudo-distance to the recirculation surface. This can be used for the "projection" of points onto the surface using a minimization approach. The gradient of the squared distance can be expressed as

$$\frac{1}{2}\nabla||\mathbf{d}||^2 \;=\; \frac{1}{2}\Big((\nabla\mathbf{d})^{\mathrm{T}}\mathbf{d}+\mathbf{d}^{\mathrm{T}}(\nabla\mathbf{d})\Big) \;=\; \nabla\mathbf{d}^{\mathrm{T}}\mathbf{d} \;=\; \mathbf{M}^{\mathrm{T}}\mathbf{d}\,. \tag{2}$$

This allows to move points $\bar{\bar{\mathbf{z}}}_0 \in \mathbb{R}^5$ that are close to the surface towards and onto the surface, in the simplest case by a gradient descent, or using a more sophisticated nonlinear solver to minimize, e.g., $||\mathbf{d}(\bar{\bar{\mathbf{z}}})||^2 + \mu||\bar{\bar{\mathbf{z}}}-\bar{\bar{\mathbf{z}}}_0||^2$, where the weighted term accounts for finding the surface point with the shortest (Euclidean) distance to $\bar{\bar{\mathbf{z}}}$, i.e., get close to a projection. Standard solvers can make use of the gradient vector $\nabla||\mathbf{d}||^2$, and the "closed form" (2) can be used instead of a finite differences approximation. We note that the computation of $\nabla\phi$ and hence $\mathbf{d}$ typically involves finite differences in 3D.

3.2 The special case $\tau \to 0$

The case $\tau = 0$ was excluded from the definition of recirculation surfaces. This is due to the denominator τ in the definition. Without this term any domain point would be part of a recirculation "surface"

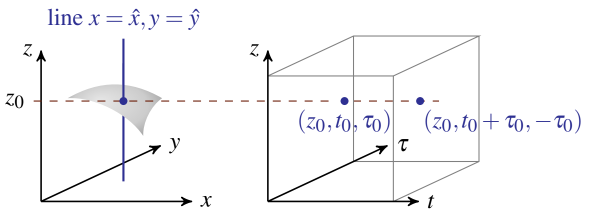

Fig. 3. In order to find an intersection of $\bar{\bar{Y}}$ with the line $\{(x,y,z)^{\mathrm{T}} : x = \hat{x}, y = \hat{y}\}$ (left), we search for isolated critical points (z_0,t_0,τ_0) and $(z_0,t_0+\tau,-\tau)$ of the 3D vector field $\mathbf{d}_{\hat{x},\hat{y}}(z,t,\tau)$ (right).

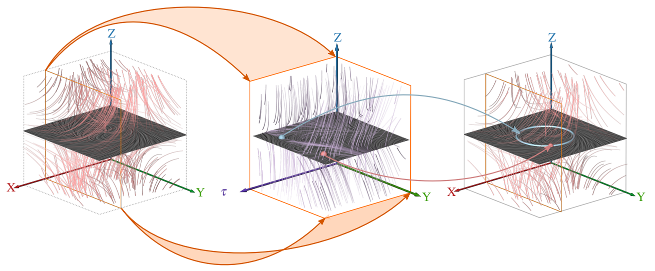

Fig. 4. Extraction of closed streamlines for the CLOSED ORBIT flow (5). Left: Place a fixed plane in $x = \hat{x}$ (orange rectangle). Center: Compute $\mathbf{d}(\hat{x},y,z,\tau)$ for this plane and extract critical points. Right: Reproject the critical points back into the plane $x = \hat{x}$ and integrate closed stream line (right).

for $\tau \to 0$. Also, a numerical root finding may fail for small τ as the absolute difference $\phi(\mathbf{x},t,\tau)-\mathbf{x}$ becomes arbitrarily small.

For the given definition of recirculation surfaces based on the scaled distance (1) there is a well-defined limit for $\tau \to 0$ as

$$\lim_{\tau\to 0}\mathbf{d} \;=\; \mathbf{v} \quad\text{and}\quad \lim_{\tau\to 0}\frac{\mathbf{G}}{\tau} \;=\; \nabla\mathbf{v} \tag{3}$$

(see appendix for details), which gives

$$\lim_{\tau\to 0}\mathbf{M} \;=\; \Big[\;\nabla\mathbf{v}\;\big|\;\mathbf{v}_t\;\big|\;\nabla\mathbf{v}\cdot\mathbf{v}+\mathbf{v}_t\;\Big]\,.$$

Equation (3) states that isolated critical points of the vector field $\mathbf{v}$ are on the recirculation surface Y, i.e.,

$$\mathbf{v}(\mathbf{x},t) = \mathbf{0} \quad\Rightarrow\quad (\mathbf{x},t,0)^{\mathrm{T}} \in \bar{\bar{Y}}\,. \tag{4}$$

3.3 Properties of the 3D surface Y

We have shown that $\bar{\bar{Y}}$ is in general a closed implicit surface in 5D. However, its projection $Y \subset \mathbb{R}^3$ is an *open* surface that is circumscribed by boundary curves. To show this, consider that if $(\mathbf{x},t,\tau)^{\mathrm{T}} \in \bar{\bar{Y}}$, then also the point $(\mathbf{x},t+\tau,-\tau)^{\mathrm{T}} \in \bar{\bar{Y}}$: both points must be on the surface. This means that almost every point on Y corresponds to *two* original points on $\bar{\bar{Y}}$. The points on $\bar{\bar{Y}}$ where both originals collapse in a single point build the *boundary curves* of Y. Since this occurs for $\tau = 0$, the paths of critical points of $\mathbf{v}$ are the boundary curves of Y due to (4).

In general, $Y \in \mathbb{R}^3$ is locally manifold. However, the projection $\mathbb{R}^5 \to \mathbb{R}^3$ may result in *self-intersecting* surfaces and possibly "degenerate" (locally) to a curve (please refer Figure 5 right).

4 Extraction of Recirculation Surfaces

In this section, we describe an algorithm for the extraction of recirculation surfaces. Using the properties of $\bar{\bar{Y}}$, there are several starting points for constructing such an algorithm.

One possible approach is using a *brute force sampling* and trying to move from random seed points in 5D onto $\bar{\bar{Y}}$ using $||\mathbf{d}||^2$ and its gradient (2). This works reasonably only for seeds that are close enough to the surface, the generation of seeds should be adapted, e.g., using some sort of importance sampling. The collected surface points can be triangulated or rendered as splats (similar to a particle based extraction of 3D ridge surfaces presented in [16]).

A different approach would start from a seed point $\bar{\bar{\mathbf{x}}}_0 \in \bar{\bar{Y}}$ and apply a *surface growing* algorithm that propagates a front over the surface by "moving" in the tangent space followed by a back-projection onto the surface. The propagating front would expand, e.g., a triangulation, and possibly split and merge.

Unfortunately, both approaches become numerically unstable especially for longer integration times because both rely on the flow map gradients: It is known that for increasing τ the gradient of the flow map tends to have an "extreme behavior" in a sense that its norm grows exponentially (see, e.g., [18]).

For this reason, we introduce an extractor that requires only the flow map but *not its gradient*. The algorithm is based on a repeated search of isolated critical points in steady 3D vector fields, which is a standard problem with well-established and robust numerical solutions.

4.1 Algorithm

The core algorithm aims at finding all intersections of Y with straight lines parallel to the coordinate axes. For instance, we search for the intersections of Y with the line $\{(x,y,z)^{\mathrm{T}} : x = \hat{x},\, y = \hat{y}\}$ that is parallel to the z-axis for fixed $\hat{x}, \hat{y}$. For this restriction, we extract isolated critical points of the 3D vector field $\mathbf{d}_{\hat{x},\hat{y}}(z,t,\tau) := \mathbf{d}(\hat{x},\hat{y},z,t,\tau)$. If we find a critical point $\mathbf{d}_{\hat{x},\hat{y}}(z_0,t_0,\tau_0) = \mathbf{0}$, then Y intersects the line in $(\hat{x},\hat{y},z_0)^{\mathrm{T}}$. Figure 3 gives an illustration. In a similar way, intersections of Y with lines parallel to the x- or the y-axis can be computed. We decompose the spatial domain using a uniform grid and apply this search along all grid lines.

Summary of the algorithm.

The extraction of Y from $\mathbf{v}$ proceeds in the following steps:

(i) Sample $\mathbf{d}$ on a regular grid and assume a piecewise quintilinear (i.e., linear in every of the five dimensions) interpolation inside the grid cells in 5D.

(ii) Extract the boundary curves of Y by tracking the critical points of $\mathbf{v}(\mathbf{x},t)$.

(iii) Find the intersections of Y with all 3D grid lines in x-, y-, and z-directions.

(iv) Visualize the sampled surface Y.

This algorithm needs further explanation.

Step (i): A complete sampling of $\mathbf{d}$ in 5D is challenging, both in terms of computation time and memory requirements. The choice of the grid resolution should reflect a trade-off between accuracy and performance. A reasonable starting point for a suitable grid resolution is the original grid on which $\mathbf{v}$ is given. However, a denser sampling grid for $\mathbf{d}$ gives better results because $\mathbf{d}$ collects information over several grid cells of $\mathbf{v}$. In order to save memory, we use an on-the-fly computation of $\mathbf{d}$ for step (iii) instead of a precomputation of $\mathbf{d}$ on the whole 5D grid.

Step (ii): This is a solved standard problem in flow visualization. We use an approach similar to [11] for a piecewise quadri-linear vector field. We consider each 4D hypercube $[x_i,x_{i+1}] \times [y_j,y_{j+1}] \times [z_k,z_{k+1}] \times [t_\ell,t_{l+1}]$ in which a quadri-linear interpolation of $\mathbf{v}$ is assumed. In order to search for the intersections with the paths of critical points, we search the isolated critical points in the 8 cubes that build the boundary cubes of the hypercube:

$$\begin{array}{ll}
[y_j,y_{j+1}] \times [z_k,z_{k+1}] \times [t_\ell,t_{l+1}] & \text{at} \quad x = x_i \\
[y_j,y_{j+1}] \times [z_k,z_{k+1}] \times [t_\ell,t_{l+1}] & \text{at} \quad x = x_{i+1} \\
[x_i,x_{i+1}] \times [z_k,z_{k+1}] \times [t_\ell,t_{l+1}] & \text{at} \quad y = y_j \\
[x_i,x_{i+1}] \times [z_k,z_{k+1}] \times [t_\ell,t_{l+1}] & \text{at} \quad y = y_{j+1} \\
[x_i,x_{i+1}] \times [y_j,y_{j+1}] \times [t_\ell,t_{l+1}] & \text{at} \quad z = z_k \\
[x_i,x_{i+1}] \times [y_j,y_{j+1}] \times [t_\ell,t_{l+1}] & \text{at} \quad z = z_{k+1} \\
[x_i,x_{i+1}] \times [y_j,y_{j+1}] \times [z_k,z_{k+1}] & \text{at} \quad t = t_\ell \\
[x_i,x_{i+1}] \times [y_j,y_{j+1}] \times [z_k,z_{k+1}] & \text{at} \quad t = t_{l+1}
\end{array}$$

We find the critical points of $\mathbf{v}$ in each subcube by the same algorithm as in step (iii). Then we connect the points by a simple heuristic preferring spatially close points to be connected.

Step (iii): We assume a multilinear interpolation in step (i). Then the restriction to grid lines $x = \hat{x}, y = \hat{y}$ yields $\mathbf{d}_{\hat{x},\hat{y}}(z,t,\tau) \in \mathbb{R}^3$ as a piecewise trilinear vector field over a regular grid. Finding its isolated critical points is another solved standard problem in Flow Visualization. We use the approach described in [37] that was briefly explained in Section 2. Since Y is double folded in 3D, we can restrict the search to positive integration times τ.

We restrict the search space to grid lines that are parallel to coordinate axes. The algorithm would work similarly with a restriction to *any* line $(1-\alpha)\mathbf{x}_1 + \alpha\mathbf{x}_2$ in 3D: the spacial coordinate (here z) of critical points is to be replaced by the parameter α.

Step (iv): The previous step generates point samples on $\bar{\bar{Y}}$. We triangulate their projection to 3D space. The resulting triangle mesh is a piecewise linear approximation to Y, which can be rendered with an piecewise additional color coding of either t or τ.

4.2 Surface reconstruction

Surface reconstruction from point samples is a nontrivial problem that is well-studied in Computer Graphics. However, there are no methods readily available for the triangulation of $\bar{\bar{Y}}$, a closed 2-manifold in 5D. Likewise, reconstruction algorithms for closed or open surfaces in 3D typically assume 2-manifolds, i.e., they cannot deal with self-intersections and degenerate situations that may occur for the projection $Y \in \mathbb{R}^3$. Also an assumption on either the sampling rate, e.g., in terms of Nyquist frequency, or equivalently the minimum local feature size is required for guaranteeing an accurate reconstruction. Unfortunately, we cannot guarantee a sufficient sampling of neither $\bar{\bar{Y}}$ nor Y. Ideally, we would like to work in the original sampling space and reconstruct by triangulation of the samples. However, we note that the projection to a 3D triangle mesh that approximates Y may locally degenerate due to topology changes, which e.g., may lead to overlapping triangles, and due to the unacceptable "distortion" from the projection.

We leave surface reconstruction of $\bar{\bar{Y}}$ and Y with guarantees as an open problem, and do not claim any novel contribution. In our experiments we used the *ball pivoting* algorithm [2]. This standard surface reconstruction method served well for our purpose for two reasons: First, it can handle surfaces with boundaries. Second, it propagates a "front" on the surface, which allows for partial processing of surface patches defined by specific t and τ intervals. This is a simple way to deal with self-intersections (or likewise surface sheets that get very close to each other): they are avoided by generating the intersecting patches sequentially and combining them into a single surface. While this works fairly well in practice, there is of course no guarantee. Our results show that – as expected – the reconstruction suffers from local undersampling, e.g., in regions of high curvature. Figure 9 shows an extreme example. We note that, ball pivoting may be a promising candidate for extension to 5D, however, it is unclear whether a denser sampling is required.

The algorithm as is does not provide an adaptive sampling. While possible in principle, a true adaptation is nontrivial as it must adapt

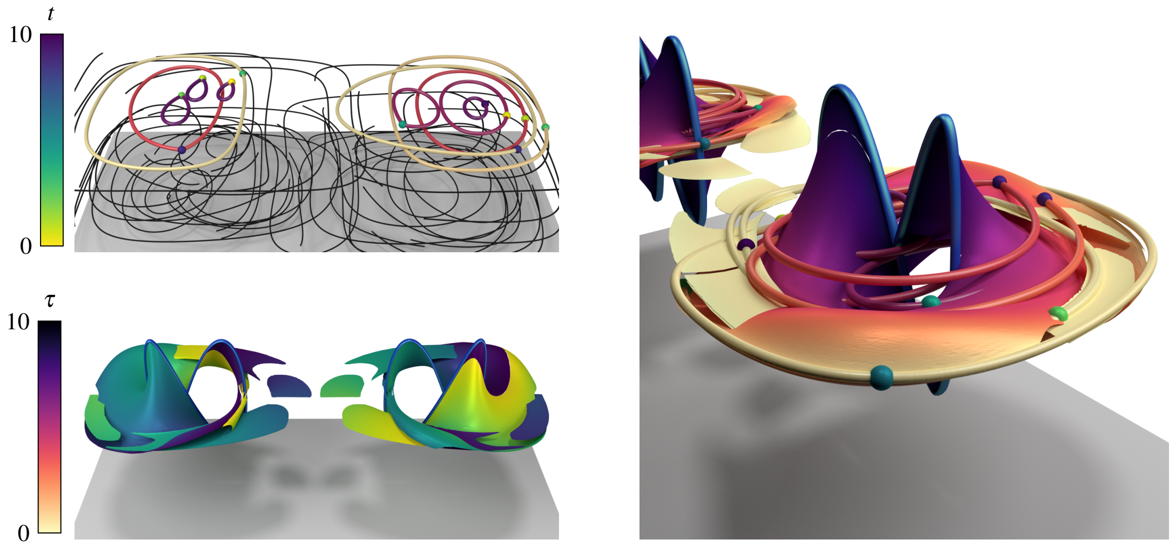

Fig. 5. 3D Double Gyre. Top-left: Random path lines including path lines with recirculation. Bottom Left: Recirculation surfaces. Close-up right: Recirculation surfaces and recirculating lines. The color map indicates start time t (bottom left) or integration time τ (right). Recirculating path lines (constant color τ) are shown with their start points (spheres of color t). Boundary curves are depicted in blue color.

to the differential geometry of a 2-manifold embedded in 5D. Nevertheless, a naïve dense sampling of the search space is too expensive, which makes the use of a multiresolution approach mandatory. We apply a simple modification to the algorithm that introduces two fixed sampling resolutions: An initial, coarser sampling grid is used to detect the presence of recirculation. Whenever a candidate sample is found on an edge of this grid, we sample a local region at finer resolution. In our experiments the size of the local region is twice the cell size in every dimension, and the sampling rate is quintupled in space and doubled in time dimensions.

5 Isolated closed stream lines in 3D steady vector fields

For 3D steady vector fields $\mathbf{v}(\mathbf{x},t) = \mathbf{v}(\mathbf{x})$, isolated closed stream lines are a special (and degenerate) case of recirculation surfaces. In the steady case, the flow map simplifies to $\phi(\mathbf{x},\tau)$, and the vector field (1) writes as $\mathbf{d}(\mathbf{x},\tau)$.

We provide an algorithm for finding isolated closed stream lines:

(i) Specify a plane perpendicular to one of the coordinate axes, e.g., the plane $x = \hat{x}$ for some constant $\hat{x}$. The next step searches for the intersections of a closed stream line with the plane.

(ii) Search for the isolated critical points in the 3D vector field $\mathbf{d}_{\hat{x}}(y,z,\tau) = \mathbf{d}(\hat{x},\hat{y},z,\tau)$. If $(y_0,z_0,\tau_0)^{\mathrm{T}}$ is such a point, the point $(\hat{x},y_0,z_0)^{\mathrm{T}}$ lies on the closed stream line of $\mathbf{v}$. Note that in this case, the closed stream line has at least two intersections with the plane.

(iii) Integrate $\mathbf{v}$ from $(x_0,y_0,z_0)^{\mathrm{T}}$ to get the complete closed stream line. — Repeat steps (i) and (ii) to find more seeds on the closed streamline if the integration time is too large for a "stable" integration.

In the first step, one has to make sure that the considered "search" planes are setup densely enough to intersect all closed orbits. This plane selection problem is shared with other algorithms for 3D closed orbit extraction that are based on FTLE [15] or the Poincaré map [26]. We do not provide a new contribution in this part of the algorithm.

We sample $\mathbf{d}_{\hat{x}}(y,z,\tau)$ on a regular hexahedral grid and apply a piecewise linear interpolation, such that the search for critical points is a standard problem (see, e.g., [37]). The problem of multiple intersections of a closed orbit with the plane is also shared with existing approaches. Also this part does not provide a new contribution.

We apply our algorithm to a simple Closed Orbit test data set

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} \frac{1-r}{r}x - \frac{1}{2}y \\ \frac{1-r}{r}y + \frac{1}{2}x \\ z \end{pmatrix} \qquad (5)$$

with $r = \sqrt{x^2+y^2}$. This test data has an isolated closed orbit with a saddle behavior: the unit circle in the (x,y)-plane. An extraction approach based on the Poincaré map fails because the Poincaré map on a plane does not converge to its invariants depicting the closed orbit. Also the FTLE-based approach by Kasten et al. [15] fails in this simple example because for a positive integration time τ, FTLE of (5) gives a constant value due to the linear third component. This means that it does not build the ridge structures that are required in the algorithm by Kasten et al.

Our algorithm extracts the closed orbit, as shown in figure 4. This result is a *new contribution*: to the best of our knowledge, no existing algorithm is able to handle this simple example. Our algorithm handles cases that so far cannot be handled otherwise.

In Section 6 we present isolated closed orbits with a saddle characteristic in a real data set: a time slice of a blood flow simulation in a cerebral aneurysm (Figure 8). This is – to the best of our knowledge – the first saddle closed orbit on real data shown in the Visualization community, which comes as a side-product of our general approach.

6 Results and Discussion

We show results for one synthetic data set and three simulated flows. All figures use the same color ramp for visualizing t and τ (see figure 5, left), the particular mapping depends on the extrema. All curves are rendered with integration time τ mapped to a constant color, boundary curves are depicted in blue color (only in figure 5, right and bottom left). Points are rendered as spheres, the color of starting points of recirculating path lines encode the start time t. For the sampling we give the resolution of the initial sampling grid (see Section 4.2) that is locally refined. Our algorithm was run in parallel on a virtual machine, that could access 18 Intel Xeon E5-2690v3 CPUs at 2.6GHz with 128GB RAM.

The 3D Double Gyre is a synthetic data set, which was introduced by Shadden et al. [30]. In the original form it represents a periodic 2D

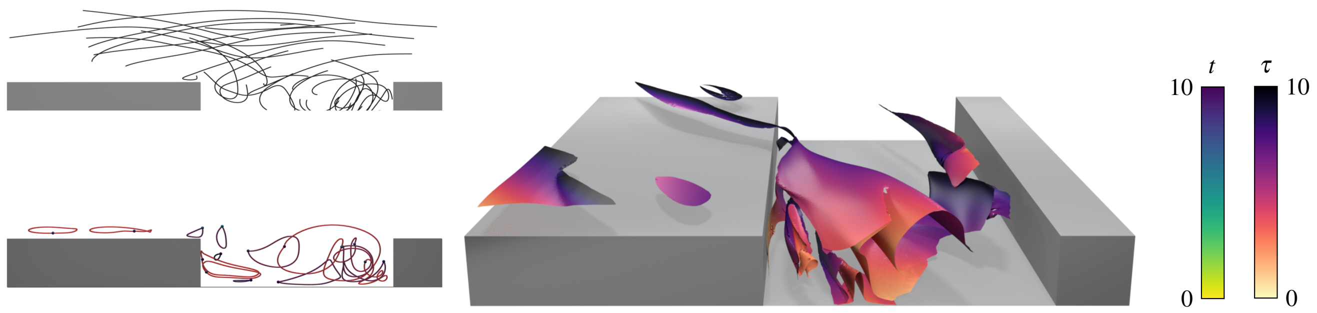

Fig. 6. CAVITY. Top left: Random path lines in the 2D cavity flow. Seeded particles move from right to left. Bottom left: Collection of recirculating path lines in and behind the cavity. The spheres mark the seed, color encodes integration time τ (curves) and t (spheres). Right: Recirculation surfaces extended to in x-y-t-space. The surface color encodes the integration time τ.

unsteady vector field. We add a third component and define

$$\mathbf{v}(\mathbf{x},t) = \begin{pmatrix} -\frac{\pi}{10}\sin(\pi f(x,t))\cos\pi y \\ \frac{\pi}{10}\cos(\pi f(x,t))\sin(\pi y)\,\frac{\mathrm{d}}{\mathrm{d}x}f(x,t) \\ \frac{1}{5}\,z\,(1-z)\,(z-\frac{1}{4}\sin\frac{2}{5}\pi t-\frac{1}{2}) \end{pmatrix} \quad \text{with}$$

$$f(x,t) = a(t)x^2 + b(t)x\ ,$$

$$a(t) = \tfrac{1}{4}\sin\tfrac{\pi}{5}t\ ,\ \ b(t) = 1-\tfrac{1}{2}\sin\tfrac{\pi}{5}t\ .$$

Particles advected in the domain $[0,2]\times[0,1]\times[0,1]$ never leave this domain.

Figure 5 (top left) gives an overview of the data by visualizing random path lines in gray color. This includes few examples of path lines that show recirculation. Their color indicates the integration time τ, the starting points are depicted as spheres, and their color encodes the start time t. Note that generally only limited insight of the data can be obtained from rendering path lines, simply because of the sheer amount of existing path lines. The bottom left figure shows a reconstruction of recirculation surfaces with t mapped as color, the boundary curves are depicted in blue color. The close-up (right) maps integration time τ as color and includes few recirculating path lines (t and τ encoded as for top left figure). We can also see the paths of critical points of $\mathbf{v}$ (blue curves) as boundaries of the recirculation surfaces, see step (ii) in the algorithm.

The search space was restricted to $(\mathbf{x},t,\tau)^{\mathrm{T}} \in [0,2]\times[0,1]\times[0,1]\times[0,10]\times[0,10]$. The size of the initial sampling grid was $200\times100\times100\times50\times50$, and 22 667 candidates were found (254 minutes). Local refinement (plus 92 minutes) generated a total of 445 062 samples. Nearly 90% of the computation time was spent on path line integration.

The ANEURYSM data set is a 3D unsteady blood flow simulation in a geometric model of a cerebral blood vessel containing a strongly developed aneurysm. The research question behind the simulation is rupture prediction. Rupture of an aneurysm is associated with a high mortality rate; depending on the rupture prediction decisions about possible surgeries are to be made. Rupture prediction of cerebral aneurysms is still largely unsolved [41]. However, it is known that rupture risk is related to the question whether the flow in the aneurysm is "cut-off" and disconnected from the main flow in the vessel. The (un-)connectedness of the flows in aneurysm and mail vessel is related to recirculation. If the main flow mainly passes through the aneurysm, few or no areas of recirculation are within the aneurysm.

Our approach finds a number of recirculation surfaces that cover large areas within the aneurysm. On the other hand, there are also large areas without recirculation surfaces, indicating that the main flow and the aneurysm flow are partly connected. While this does certainly not solve the rupture prediction problem, the systematic study of recirculation in many aneurysm data sets may reveal correlations between presence, shape and size of recirculation surfaces and likely rupture events.

Figure 7 shows results: Few random path lines illustrate the flow, which enters the domain from the left side and leaves to the right. A collection of recirculating path lines indicate the presence of recirculation. The spheres indicate the start positions on the recirculation surfaces with color encoding start time t. The colors of the curves encode the integration time τ. The same color codes are used for the generated recirculation surfaces. Part of the occluding surface has been removed in the bottom right image to clear the view on the recirculation surface in the center of the domain.

In addition, we examined a single time slice of the ANEURYSM data and searched for isolated closed stream lines in the steady flow. We found two close stream lines, both with *saddle behavior*! Figure 8 shows the stream lines together with a visualization of the Poincaré maps. For this visualization, planes through the red and blue points orthogonal to the velocity direction are considered. In each plane, a small closed seed curve – a circle – is constructed, see the left column of Figure 8 (right). The points on this circle are integrated both in forward and backward direction until they intersect the plane again. The resulting intersection curves are the closed curves in the center and right columns of Figure 8 (right). The relation of these integrated curves and the seed circle characterize the closed orbit: Since the curves are partially inside and partially outside the seed circle, we have a closed orbit with saddle behavior.

The search space was restricted to $(\mathbf{x},t,\tau)^{\mathrm{T}} \in [-0.0155,-0.025]\times[0.2245,0.2355]\times[-0.1765,-0.1645]\times[0.0,0.75.]\times[0.0,0.25.]$ and the size of the initial sampling grid was $[130\times110\times120\times150\times50]$. 29 228 candidates were found (124 hours, 98% path line integration) which resulted in 458 362 total samples after refinement (46 hours).

The CAVITY data set is a vector field describing the flow over a 2D cavity. This data set was kindly provided by M. and E. Caraballo [5] and B. Noack and I. Pelivan. 1 000 time steps were simulated using the compressible Navier-Stokes equations. The flow exhibits a non-zero divergence inside the cavity, while outside the cavity the flow tends to have a quasi-divergence-free behavior. The data is almost (but not perfect) periodic, with a period of about 100 time steps in length, and only the first 100 time steps are shown.

Cavity flows (i.e., laminar flows passing over an open cavity) are of interest in many applications in engineering, ranging from the small cavities due to gaps in the body work of vehicles, the shapes of river channel beds, via cargo bays on aircraft, to the large scale flows in urban street canyons.

The main analysis question for the data set is: Do particles in the cavity always move out after a while, and if so, what is the maximal dwell time in the cavity? Clearly, this question is strongly related to

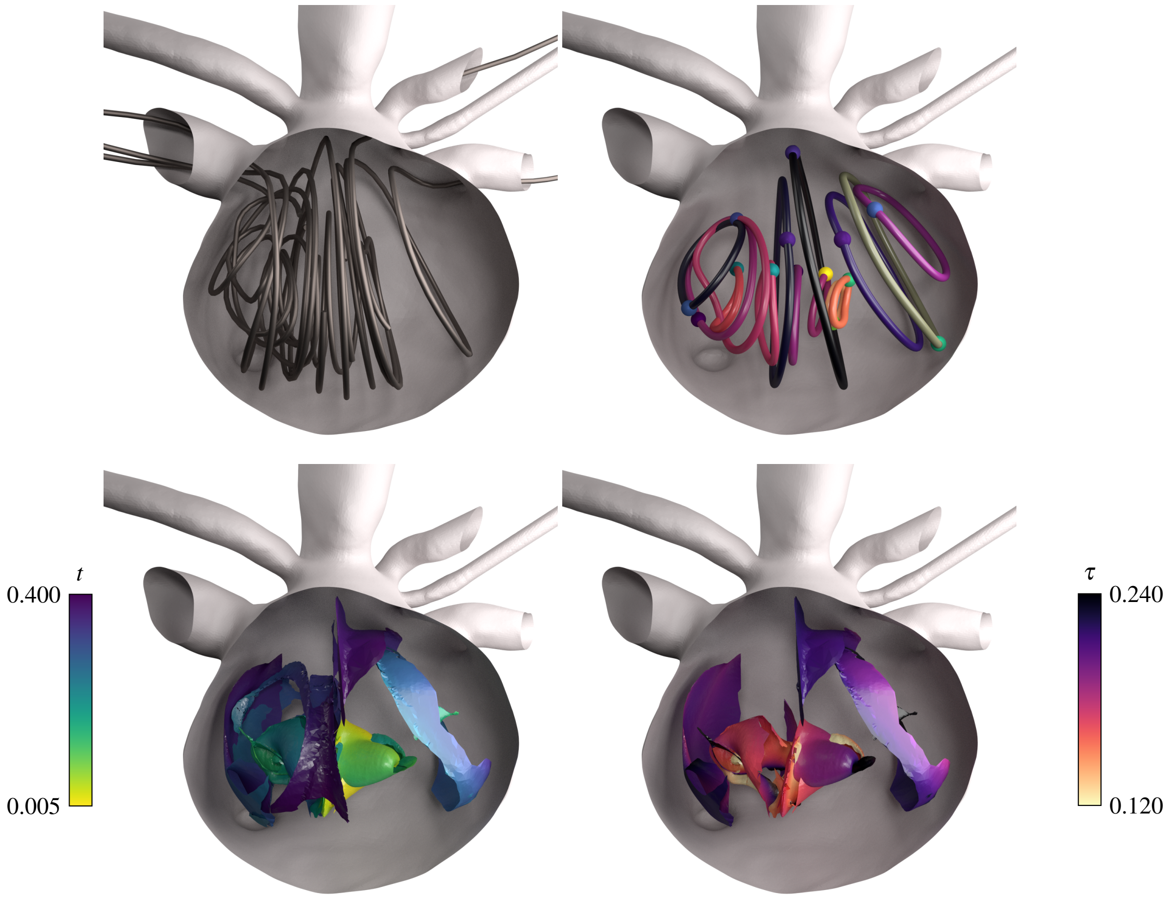

Fig. 7. ANEURYSM. Top left: Random path lines in the Aneurysm flow. The flow passes from left (inflow) to right (outflow). Top right: Some recirculation path lines. The spheres mark the seed, color encodes total integration time τ (curves) and t at start (spheres). The bottom shows recirculation surfaces, color encodes t (left) and τ (right). The center surfaces was cut out in the right image to clear the view on the surfaces in the center region.

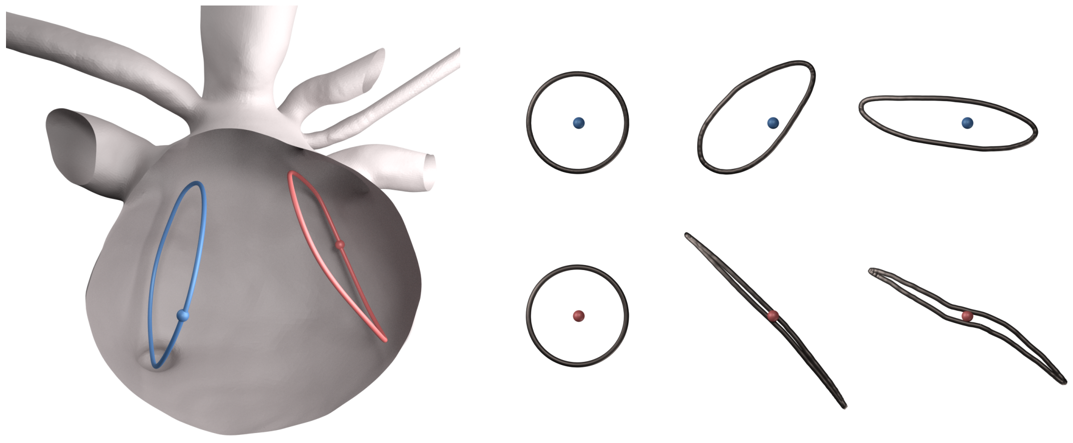

Fig. 8. Single time slice of ANEURYSM. Left: The single time slice defines a steady vector field, which contains two *isolated closed stream lines*. The spheres denote the start positions. Right: The planar curves in each row visualize the Poincaré maps for both isolated closed stream lines at the start positions (colored spheres). The columns show (from left) the circular seed structure, the seed advected by one turn in forward and in backward direction. Both isolated closed stream lines show saddle behavior.

recirculation within the cavity. As the this is a 2D time-dependent flow, recirculation surfaces are defined in x-y-t-τ-space. We found a rather dense set of recirculation surfaces with rather long τ values inside the cavity, which indicate a high number of particles that remain in the cavity for a longer period of time. Further, we found more recirculation surfaces downstream outside the cavity, which indicate the impact of the cavity even in a certain distance. Figure 6 shows results: random collections of path lines, recirculating path lines same style as the previous figures. We use the start time t as third dimension and extend the recirculation curves to surfaces in x-y-t-space.

The search space was restricted to $(\mathbf{x}, t, \tau)^{\mathrm{T}} \in [-1.0, 8.0] \times [-1.0, 1.5] \times [0.0, 10.0] \times [0, 10.]$ and the size of the initial sampling grid was $[900 \times 250 \times 50 \times 50]$ (There is no z-direction). 257 080 samples were found (10.7 hours, 60% path line integration) and no refinement was done, because the initial result was sufficient.

The SQUARE CYLINDER data set describes the simulated flow around a 3D square cylinder and is based on the Navier-Stokes simulation by Camarri et al. [4]. The uniformly resampled version of this vector field sequence was provided by Tino Weinkauf (see [36]). Figure 9 shows few random path lines (top left) and recirculating path lines (bottom left, same style as for previous figures) in a region close behind the obstacle. Figure 9 (center) shows recirculation surfaces in the same region of the domain. The recirculation surfaces near the obstacle show high curvature, and the reconstruction algorithm (see Section 4.2) is unable to generate a consistent surface mesh due to undersampling. The recirculation surfaces at higher times show smaller curvature and we thus obtain a more sufficient sampling and a better reconstruction.

The search space was restricted to $(\mathbf{x}, t, \tau)^{\mathrm{T}} \in [0.5, 2.5] \times [-0.65, 0.65] \times [0, 6] \times [0, 6] \times [0, 6]$. The size of the initial sampling grid was $[400 \times 270 \times 1200 \times 30 \times 30]$, and 269 187 candidates were found (43 hours). Local refinement (plus 25 hours) generated a total of 2 217 204 samples. About 75% of the computation time was spent on path line integration.

In the following we discuss the value of *recirculation surfaces* for flow visualization and discuss their algorithmic reconstruction.

Relation to other Flow Visualization techniques

There is a variety of visualization techniques that locally focus on the velocity field. In recent years, integration based techniques moved into the focus of research, i.e., techniques that apply a visual analysis of the flow map. We claim that our technique is the *first comprehensive feature extraction approach incorporating the complete flow map*. Analyzing the flow map bears three challenges:

- It is a high-dimensional (5D) continuous field.
- It is expensive to compute.
- It contains large gradients, which accounts for extremely dense sampling, especially in the interesting areas where flow separation occurs.

In addition, our technique opts for a *strong abstraction*: We search for a steady 3D surface representing the relevant information of a complete 5D field. Domain experts would like to see a "still image" – that can be physically printed as a figure in an article – that tells as much as possibly of the story of a 3D unsteady flow. While there exists a number of solutions for steady flows (see, e.g., [28] and the references therein), we are not aware of any similar approaches for unsteady flows. In this sense this work could be considered as the first one in this direction.

Because of the high computational complexity, existing methods for flow map analysis focus on subsets of ϕ. For instance, the well-known FTLE visualization techniques usually pick a particular t and τ to either volume render or do feature extraction on the resulting 3D scalar field. Further, different t or different τ can be shown via animations. We are not aware of any approach to systematically computing and visualizing FTLE fields for all t and all τ values. We are neither aware of any other technique that systematically explore the whole 5D flow map.

Degeneration of recirculation surfaces

Recircualtion surfaces can degenerate to points, lines and volumes, but these cases are structurally unstable: adding noise to the data will break the points/lines/volumes to a finite number of surfaces. An exception is the case of steady flows where – as shown in Section 5 – recirculation surfaces degenerate to structurally stable isolated closed stream lines.

In order to illustrate the idea of structural stability, consider algorithms for isosurface extraction in 3D scalar fields. Such isosurfaces can collapse to a single point (for instance the scalar field $s(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{x} = 0$ in the origin) or to a volume (for instance $s(\mathbf{x}) \equiv 0$). These cases are usually excluded from an isosurface extraction algorithm because of their structural instability. We do the same for recirculation surfaces.

Recirculation as a phenomenon

We are not sure if recirculation surfaces are the best approach to treat recirculation as a phenomenon. Neither exists a common definition of recirculation, nor are there ground truth data or other approaches to compare with. Thus, a final answer can only be given in the future by test of time.

However, there are points that give some evidence on the usefulness of our approach:

User feedback. — We presented the approach to domain experts that are involved in numerical flow simulation. They acknowledged the usefulness and found the detected structures interesting and informative. However, we consider this positive feedback as anecdotic. A formal user study was neither done, nor are we aware of a proper approach to the verification – or falsification – of recirculation surfaces by a user study.

Mixture of expected and surprising results. — The found recirculation surfaces mainly correspond with the expectation. For instance, for the SQUARE CYLINDER data we expected recirculation behind the cylinder, and our approach confirms this. On the other hand, there were surprising results: For the cavity data set, we found recirculation surfaces both inside the cavity and outside downstream the flow.

Closed orbits. — We interpret the fact that a special case of our algorithm solves the problem of extracting isolated closed orbits with saddle behavior as a point of evidence for the usefulness of recirculation surfaces.

Computation time

The computation time is long. The reason is in the nature of flow maps described above. We are not aware of any faster algorithm doing a systematic analysis of the whole flow map.

Further, our approach can be seen as *preprocessing step*: Recirculation surfaces are computed once and stored along with the flow data. Then recirculation surfaces can be explored by interactively seeding path lines on them and observing their behavior (see the accompanying video).

Parameters

The presented algorithm comes with a number of parameters that can be divided into two classes: first, parameters of the existing standard algorithms that our algorithm is based upon, and second, new parameters inherent to the new algorithm.

The first group includes the choice of ODE solver and its parameters required for numerical path line integration. For instance, we use a fourth order Runge-Kutta solver with adaptive step size that is controlled, e.g., by error tolerance and maximum step size. Furthermore there is the initial grid resolution for finding isolated critical points and the maximum recursion depth to bound subdivision as well as a strategy for clustering multiply detected points. Finally, a heuristic is required for tracking critical points that decides when to locally connect extracted critical points at different time steps. We do not further discuss these parameters here because they are well-discussed in the original publications. Instead, we consider the algorithms as black-boxes for our approach.

The parameters of the second group reduce to the grid resolution for sampling $\mathbf{d}$. Its variation does not show any surprising behavior: a

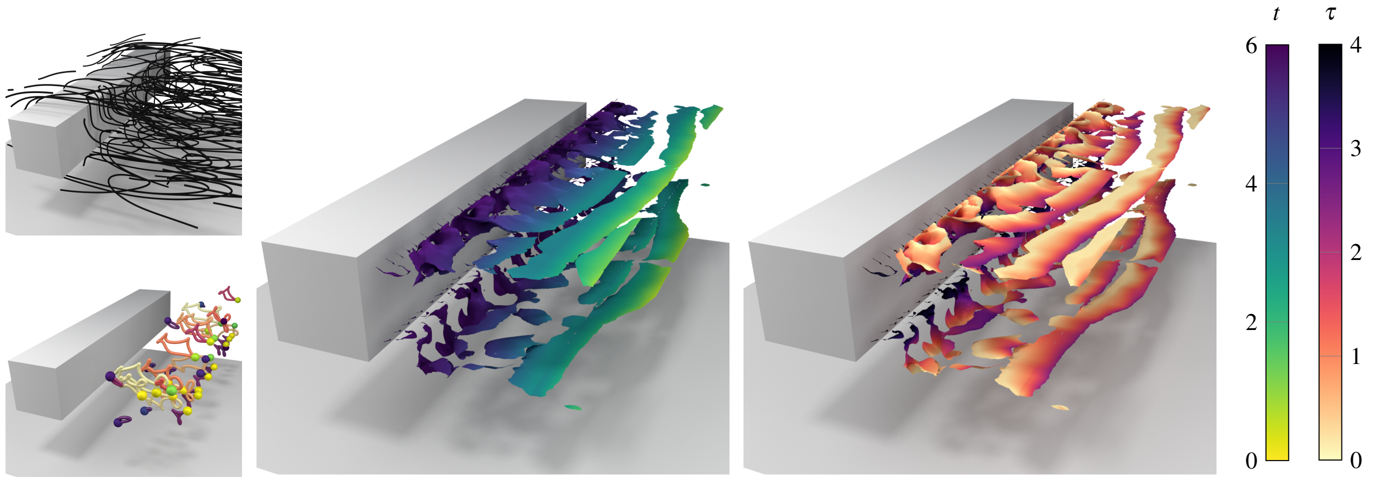

Fig. 9. Square Cylinder. Top left: Random path lines close behind the obstacle. Bottom left: Recirculating path lines close behind the obstacle, color encodes total integration time τ (curves) and t at start (spheres). Center: Recirculation surfaces behind the obstacle, color encodes t. Right: Same with color-coded τ. The surface shows a high spatial curvature directly behind the cylinder. The sampling was not dense enough to capture the recirculation everywhere sufficiently. Therefore the reconstructed surface has jagged borders and some holes.

higher sampling rate increases accuracy and computation time. In theory, the sampling rate should not fall below the Nyquist frequency to capture the recirculation surfaces. It can be bounded by using assumptions on the local feature size or curvature of the surface. However, we see currently no way to determine the (local) sampling rate reliably. The situation is significantly more difficult as for surface reconstruction in 3D – which is a difficult problem already: A sampling and reconstruction must take into account curvature of a 2-manifold embedded in a 5D space. We know that this manifold has regions of high curvature. Moreover, the differential geometry in 5D does not carry over to the projection to 3D, the ultimate visualization space. For instance, the projection $Y \in \mathbb{R}^3$ is in general a 2-manifold. However, configurations in the projection can lead to different topologies of the projected surface in 3D: First, points $(\mathbf{x}, t+\tau, -\tau)^{\mathrm{T}} \in \bar{\bar{Y}}$ build boundary curves of Y (see Section 3.3). Second, closed stream lines in steady flows $\mathbf{v}(\mathbf{x},t) = \mathbf{v}(\mathbf{x})$ constitute a degenerate recirculation surface (see Section 5).

In this work, we chose the simplest approach and use regular sampling. The disadvantages of this approach are potential undersampling, i.e., loss of information, or oversampling, i.e., waste of computation time. Both are likely to occur at least locally: the grid provides no means of adaptation. This property, however, is shared with many very successful algorithms that restrict themselves to operating on uniform grids. The simplicity of this approach is at the same time its biggest advantage. In our setting, it allows for relying on fairly simple and proven standard methods and leads to an only moderately complex implementation.

7 Conclusions and Future Work

Recirculation is an important phenomenon that is studied in many application domains. Recirculation can be explained and understood intuitively. Surprisingly there existed no formal definition so far. In section 3 we gave a formal definition as well as properties of *recirculation surfaces* and showed how to compute them in section 4. We search the surface in 5D by intersections with lines. This way we are able to project the problem to a 3D search for critical points in constant vector fields. The search for isolated closed stream lines is a special case of our approach, which was explained in section 5. In section 6 we showed results followed by a discussion.

Our approach provides points of the recirculation surface on the edges of a regular grid. On the one hand, this ensures a certain minimum density of samples points (within in the range covered by the grid), on the other hand it hinders a higher density of the samples, leading to artifacts in the point based rendering. We see two ways to deal with this: a modification of the algorithm to adaptive grids, or the reconstruction of a triangle mesh from sample points. For the latter option, surface extraction could be carried out by an algorithm similar to Marching Cubes. However, in addition to the standard Marching Cubes configurations, few more cases have to be considered: The 3D surface Y can have self-intersection and therefore it can have more than one intersection point with an edge of a cell. Moreover, in the presence of boundary curves, Y may enter a cell but does not necessarily leave it but end in the boundary curve instead.

Another current limitation is performance. The obvious solution for future research is adaptive sampling in 5D, which clearly has the potential to speed up the algorithm. On the other hand, no extreme improvement of the performance can be expected since our algorithm does an exhaustive feature extraction in an expensive 5D space.

Appendix

We show

$$\nabla\phi\ \mathbf{v} + \phi_t = \mathbf{v}(\phi, t+\tau)$$

for $\mathbf{v} = \mathbf{v}(\mathbf{x},t)$ and $\phi = \phi(\mathbf{x},t,\tau)$.

We consider the 3D time-dependent velocity field $\mathbf{v}$ as 4D steady velocity field $\bar{\mathbf{p}} = (\mathbf{v}, 1)^{\mathrm{T}}$. This gives the 4D flow map for $\bar{\mathbf{p}}$ as $\bar{\phi} = (\phi, t-\tau)^{\mathrm{T}}$. Computing its (space-time) gradient gives $\nabla\bar{\phi} = \begin{bmatrix} \nabla\phi & \phi_t \\ \mathbf{0} & 1 \end{bmatrix}$. Consider a point $(\mathbf{x}_0, t_0)$ under integration of $\bar{\mathbf{p}}$. Further, consider a point $(\mathbf{x}_1, t_1)$ in a small linear neighborhood of $(\mathbf{x}_0, t_0)$: $(\mathbf{x}_1, t_1) = (\mathbf{x}_0, t_0) + \varepsilon\bar{\mathbf{r}}$ for a small ε. Then the definition of the flow map gradient gives $\frac{1}{\varepsilon}\left(\bar{\phi}(\mathbf{x}_1, t_1, \tau) - \bar{\phi}(\mathbf{x}_0, t_0, \tau)\right) = \nabla\bar{\phi}(\mathbf{x}_0, t_0, \tau)\ \bar{\mathbf{r}}$. If we place $(\mathbf{x}_1, t_1)$ on the path line through $(\mathbf{x}_0, t_0)$, i.e., $\bar{\mathbf{r}} = \bar{\mathbf{p}}(\mathbf{x}_0, t_0)$, it remains on the same path line during integration. This gives $\frac{1}{\varepsilon}\left(\bar{\phi}(\mathbf{x}_1, t_1, \tau) - \bar{\phi}(\mathbf{x}_0, t_0, \tau)\right) = \bar{\mathbf{p}}(\bar{\phi}(\mathbf{x}_0, t_0, \tau)) = \nabla\bar{\phi}(\mathbf{x}_0, t_0, \tau)\ \ \bar{\mathbf{r}}$. Rewriting this condition in the spatial coordinates gives the postulated equation. □

Assuming a smooth vector field $\mathbf{v} \in C^1$, we derive the limits

$$\lim_{\tau\to 0} \mathbf{d} = \lim_{\tau\to 0} \frac{\phi - \mathbf{x}}{\tau} = \lim_{\tau\to 0} \frac{\phi(\mathbf{x},t,\tau) - \phi(\mathbf{x},t,0)}{\tau} = \frac{\partial\phi}{\partial\tau} = \mathbf{v}(\mathbf{x},t)\ ,$$

$$\lim_{\tau\to 0} \frac{\mathbf{G}}{\tau} = \lim_{\tau\to 0} \frac{\nabla\phi - \mathbf{I}}{\tau} = \lim_{\tau\to 0} \frac{\nabla\phi(\mathbf{x},t,\tau) - \nabla\phi(\mathbf{x},t,0)}{\tau} = \nabla\mathbf{v}(\mathbf{x},t)\ .$$

Acknowledgments

We thank Tino Weinkauf, John McLaughlin and Anke Friederici for their contributions to earlier versions of the paper. The work was partially supported by DFG grant TH 692/8-1.

References

[1] D. Asimov. Notes on the topology of vector fields and flows. Technical report, NASA Ames Research Center, 1993. RNR-93-003.

[2] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[3] S. L. Brunton and B. R. Noack. Closed-loop turbulence control: progress and challenges. *Applied Mechanics Reviews*, 67(5):050801, 2015.

[4] S. Camarri, M. V. Salvetti, M. Buffoni, and A. Iollo. Simulation of the three-dimensional flow around a square cylinder between parallel walls at moderate reynolds numbers. In *XVII Congresso Di Meccanica Teorica Ed Applicata*, 2005.

[5] E. Caraballo, M. Samimy, and J. DeBonis. Low dimensional modeling of flow for closed-loop flow control. *AIAA 2003-59*, 2003.

[6] G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang. Vector field editing and periodic orbit extraction using morse decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):769–785, July 2007. doi: 10.1109/TVCG.2007.1021

[7] G. Chen, K. Mischaikow, R. S. Laramee, and E. Zhang. Efficient morse decompositions of vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):848–862, 2008.

[8] A. Chern, F. Knöppel, U. Pinkall, and P. Schröder. Inside fluids: Clebsch maps for visualization and processing. *ACM Trans. Graph.*, 36(4):142:1–142:11, 2017.

[9] A. Debien, K. A. F. F. von Krbek, N. Mazellier, T. Duriez, L. Cordier, B. R. Noack, M. W. Abel, and A. Kourta. Closed-loop separation control over a sharp edge ramp using genetic programming. *Experiments in Fluids*, 57(3):40, 2016.

[10] M. P. Do Carmo. *Differential Geometry of Curves and Surfaces*. 1976.

[11] C. Garth, X. Tricoche, and G. Scheuermann. Tracking of vector field singularities in unstructured 3d time-dependent datasets. In *Pro. IEEE Visualization*, pp. 329–336. IEEE, 2004.

[12] N. Gautier, J.-L. Aider, T. Duriez, B. R. Noack, M. Segond, and M. Abel. Closed-loop separation control using machine learning. *Journal of Fluid Mechanics*, 770:442–457, 2015.

[13] T. Günther, M. Gross, and H. Theisel. Generic objective vortices for flow visualization. *ACM Trans. Graph.*, 36(4):141:1–141:11, 2017.

[14] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Phys. D*, 149(4):248–277, Mar. 2001.

[15] J. Kasten, J. Reininghaus, W. Reich, and G. Scheuermann. Toward the extraction of saddle periodic orbits. In *Topological Methods in Data Analysis and Visualization III*, pp. 55–69. Springer, 2014.

[16] G. L. Kindlmann, R. S. J. Estépar, S. M. Smith, and C.-F. Westin. Sampling and visualizing creases with scale-space particles. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1415–1424, 2009.

[17] T. Klein and T. Ertl. Scale-space tracking of critical points in 3d vector fields. In *Topology-Based Methods in Visualization*, pp. 35–49. Springer, 2007.

[18] A. Kuhn, C. Rössl, T. Weinkauf, and H. Theisel. A benchmark for evaluating ftle computations. In *Proc. IEEE Pacific Visualization Symposium*, PacificVis, pp. 121–128, 2012.

[19] R. Laramee, D. Weiskopf, J. Schneider, and H. Hauser. Investigating swirl and tumble flow with a comparison of visualization techniques. In *IEEE Visualization 2004*, pp. 51–58. IEEE Comput. Soc. doi: 10.1109/VISUAL.2004.59

[20] S. Mann and A. Rockwood. Computing singularities of 3D vector fields with geometric algebra. In *Proc. IEEE Visualization*, pp. 283–289, 2002.

[21] N. Max and T. Weinkauf. Critical points of the electric field from a collection of point charges. In H.-C. Hege, K. Polthier, and G. Scheuermann, eds., *Topology-Based Methods in Visualization II*, Mathematics and Visualization, pp. 101–114. Springer, 2009.

[22] M. Mittasch, P. Gross, M. Nestler, A. W. Fritsch, C. Iserman, M. Kar, M. Munder, A. Voigt, S. Alberti, S. W. Grill, and M. Kreysing. Noninvasive perturbations of intracellular flow reveal physical principles of cell organization. *Nature Cell Biology*, 20(3):344–351, mar 2018. doi: 10.1038/s41556-017-0032-9

[23] B. R. Noack, I. Mezić, G. Tadmor, and A. Banaszuk. Optimal mixing in recirculation zones. *Physics of Fluids*, 16(4):867–888, 2004.

[24] R. Peikert and F. Sadlo. Topology-guided visualization of constrained vector fields. In H. Hauser, H. Hagen, and H. Theisel, eds., *Topology-based Methods in Visualization*, Mathematics and Visualization, pp. 21–34. Springer, 2007.

[25] R. Peikert and F. Sadlo. Flow topology beyond skeletons: Visualization of features in recirculating flow. In *Topology-Based Methods in Visualization II*, pp. 145–160. Springer, 2009.

[26] W. Reich, D. Schneider, C. Heine, A. Wiebel, G. Chen, and G. Scheuermann. Combinatorial vector field topology in three dimensions. In *Topological Methods in Data Analysis and Visualization II*, pp. 47–59. Springer, 2012.

[27] A. Sanderson, G. Chen, X. Tricoche, and E. Cohen. Understanding quasi-periodic fieldlines and their topology in toroidal magnetic fields. In R. Peikert, H. Hauser, H. Carr, and R. Fuchs, eds., *Topological Methods in Data Analysis and Visualization II*, pp. 125–140. Springer, 2012.

[28] M. Schulze, J. Martinez Esturo, T. Günther, C. Rössl, H.-P. Seidel, and T. Weinkauf. Sets of globally optimal stream surfaces for flow visualization. *Comput. Graph. Forum (Proc. EuroVis)*, 33(3):1–10, 2014.

[29] S. C. Shadden, M. Astorino, and J.-F. Gerbeau. Computational analysis of an aortic valve jet with lagrangian coherent structures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1):017512, 2010.

[30] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212(7), 2005.

[31] Y. K. Suh. Periodic motion of a point vortex in a corner subject to a potential flow. *Journal of the Physical Society of Japan*, 62(10):3441–3445, 1993.

[32] G. Tadmor and A. Banaszuk. Observer-based control of vortex motion in a combustor recirculation region. *IEEE Transactions on Control Systems Technology*, 10(5):749–755, Sep 2002.

[33] H. Theisel and H.-P. Seidel. Feature flow fields. In *VisSym*, vol. 3, pp. 141–148, 2003.

[34] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Grid-independent detection of closed stream lines in 2d vector fields. In *VMV*, vol. 4, pp. 421–428, 2004.

[35] X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computers & Graphics*, 26(2):249–257, 2002.

[36] W. von Funck, T. Weinkauf, H. Theisel, and H.-P. Seidel. Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. *IEEE Transactions on Visualization and Computer Graphics (Proc. Visualization)*, 14(6):1396–1403, 2008.

[37] T. Weinkauf. *Extraction of Topological Structures in 2D and 3D Vector Fields*. PhD thesis, University Magdeburg, 2008.

[38] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Feature flow fields in out-of-core settings. In H. Hauser, H. Hagen, and H. Theisel, eds., *Topology-based Methods in Visualization*, Mathematics and Visualization, pp. 51–64. Springer, 2007.

[39] T. Weinkauf, H. Theisel, A. Van Gelder, and A. Pang. Stable feature flow fields. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):770–780, 2011.

[40] T. Wischgoll and G. Scheuermann. Detection and visualization of closed streamlines in planar flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001.

[41] J. Xiang, V. Tutino, K. Snyder, and H. Meng. Cfd: Computational fluid dynamics or confounding factor dissemination? the role of hemodynamics in intracranial aneurysm rupture risk assessment. *American Journal of Neuroradiology*, 35(10):1849–1857, 2014. doi: 10.3174/ajnr.A3710