Stream Volume Segmentation of Grid-Less Flow Simulation

Harald Obermaier[1], Jörg Kuhnert[2], Martin Hering-Bertram[2], and Hans Hagen[1]

[1] University of Kaiserslautern, Germany
[2] Fraunhofer ITWM Kaiserslautern

Abstract. We present a novel algorithm for the geometric extraction of stream volume segmentation for visualization of grid-less flow simulation. Our goal is the segmentation of different paths through a mixing tube where the flow is represented by scattered point sets approximated with moving least squares. The key challenges are the watertight construction of boundary representations from separatrices. These are obtained by integrating and intersecting stream surfaces starting at separation and attachment lines at boundaries of flow obstacles. A major challenge is the robust integration of stream lines at boundaries with no-slip condition such that closed volume segments are obtained. Our results show the segmentation of volumes taking consistent paths through a mixing tube with six partitioning blades. Slicing these volumes provides valuable insight into the quality of the mixing process.

1 Motivation

Topological features of flow fields such as separatrices are of specific interest in vector field visualization, since they partition the domain into connected segments with consistent limit behavior of stream lines, classifying (stationary) flow regions based on their sources and targets. In planar vector fields, critical points often determine these flow targets besides cycles and boundary segments [5]. In three-dimensional incompressible flow critical points are rather rare. Sinks and sources do not exist, due to conservation of mass, and only saddles of different types may occur. Thus, separatrices mostly emerge from separation and attachment lines on boundary surfaces from where they may be integrated with sophisticated stream surface extraction.

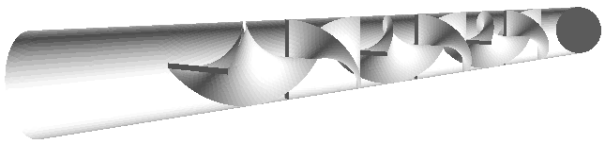

Fig. 1. Cylindrical mixing tube with six blades.

In the present work we contribute a novel algorithm for the extraction of watertight boundary representations for stream volumes. The algorithm is used to analyze a three-dimensional vector field describing a mixing process simulated by the *Finite Pointset Method* (FPM) [13]. The underlying application is concerned with flow of liquid glass at high temperature through a cylindrical mixing tube (see Fig. 1). Six twisted blades located sequentially inside the tube partition the flow into 64 different paths, each associated with a stream volume composed of all stream lines taking the same path. Both, the granularity and the geometric shape of these separating structures are indicators of the quality of the mixing process, i.e. maintaining homogeneous optical properties of the glass before it is casted into its final shape.
Our application data set describes a stationary velocity field of viscous incompressible flow, represented by a finite point set with associated field attributes. Due to the absence of a computational grid, the data needs to be approximated with a local method like *Moving Least Squares* (MLS) which is also used in the simulation code. A major challenge is the integration of stream lines along surfaces with no-slip boundary, since the approximated vector field may not get exactly zero on the boundary and may even reverse its direction. To obtain a valid segmentation, stream lines on boundaries are projected on the surface and released on intersection with a separation line. The framework composed of such separating boundary stream lines, separation, and attachment lines connects the material boundaries to the complex of inner separatrices that need to be constructed carefully by adaptive integration due to varying complexity of the vector field.
We present a novel algorithm for the adaptive construction of watertight stream volumes by mutual intersection of separatrices in three-dimensional flow, used for quality analysis of mixing processes. These are the main challenges arising during stream-volume construction:

- **No neighborhood relation between data points.** The vector field is approximated by a mesh-free approximation technique called "Moving Least Squares". For each evaluation, a local point set is defined using weighting and visibility queries.
- **No-slip boundary and flow obstacles.** To create watertight stream volumes, one needs to be capable of integrating stream lines along geometry. Additionally, flow obstacles have an impact on the construction of stream surfaces due to their splitting behavior.
- **Intersection of separatrices.** Intersected stream surfaces generated by separation or attachment lines on flow obstacles yield parts of separatrices, that are recombined to define boundaries of stream volumes.

In Sect. 2 we summarize fundamentals and refer to related work that has been done in the field of vector field approximation and stream surface construction. Sect. 3 describes our approaches to stream volume construction and visualization. We provide numerical examples of stream volume visualizations in Sect. 4, which as well contains an analysis of the test data set and an outlook on future work.

2 Fundamentals and Related Work

2.1 Moving Least Squares Approximation

Let S be a field of scattered points $x_i \in \Omega \subseteq \mathbb{R}^n$ with function values $f_i \in \Pi \subseteq \mathbb{R}^m$. A method suitable to approximate such grid-less data is the MLS approach [7, 8]. MLS is a weighted, local generalization of the well-known "Least Squares" technique, which fits a polynomial of given degree to a set of points while minimizing the squared distance to corresponding field values.
For a large set of points, it is not feasible to find a globally defined polynomial f that still provides a fairly small minimum overall error, as obtained by a standard Least Squares approximation. To construct a locally supported function $f_{x'}$ for arbitrary fixed $x' \in \Omega$, the standard LS equation is altered by introducing a compactly supported weighing function. The approximating polynomial $f_{x'}$ must then satisfy (1).

$$\sum_i \omega(x', x_i) \, ||f_i - f_{x'}||^2 \rightarrow \min \quad \text{with } \omega(x', x_i) = e^{-\alpha \frac{||x'-x_i||^2}{r^2}} - e^{-\alpha} \tag{1}$$

where r is given by the simulation and defines the radius of influence or "smoothing length". Only points whose distance to x' does not exceed r are used for evaluation. MLS uses (1) to construct an approximating function f by moving x' over the domain of S. Therefore f is not defined by a single fixed $x' \in \Omega$, but needs to be evaluated at every $x' = x \in \Omega$ separately. This construction creates f as a composition of multiple $f_{x'}$. Let $m = 1$ and $f_{x'}$ be a polynomial of the general form $f_{x'}(x) = \boldsymbol{a}_{x'}^T \cdot \boldsymbol{v}(x)$ with $\boldsymbol{a}_{x'}$ an unknown vector of coefficients and $\boldsymbol{v}(x)$ a given polynomial base vector of degree d. For $n = 2$, $m = 1$, and $d = 1$, (2) needs to be solved for $\boldsymbol{a}$ to obtain an approximating polynomial.

$$\boldsymbol{a} \sum_i \omega(\mathbf{x}', \mathbf{x_i}) \begin{pmatrix} 1 & x & y \\ x & x^2 & xy \\ y & xy & y^2 \end{pmatrix} = \sum_i \omega(\mathbf{x}', \mathbf{x_i}) \begin{pmatrix} 1 \\ x \\ y \end{pmatrix} f_i \tag{2}$$

2.2 Line, Surface, and Volume Integration

Stream lines provide a simple way of visualizing particle traces in stationary fields [11]. Their computational complexity depends on the method used for vector field evaluation as well as on the integration method and adaptivity scheme. When integrating stream lines, appropriate measures have to be taken to guarantee a given accuracy. We use an embedded Runge-Kutta integration scheme of 4th/5th order to generate adaptive stream lines. Hereby a comparison of the two different results with respect to angular deviation is used to scale the step size for a fifth order Runge-Kutta integration.
Stream surfaces represent a generalization of the uni-variate stream lines. They are of special importance to the analysis of mixing processes as they provide the basis for constructing three-dimensional separatrices. The introduction of stream

surfaces necessitates new concepts of adaptivity. Such a concept was presented by Hultquist et al. [6] and has been refined in the context of grid-based data sets by different authors such as Scheuermann et al. [10] and Garth et al. [4]. The underlying stream surface generation algorithm used in this paper is based on an extended version of this ribbon-based approach. Stream surfaces generated by methods discussed in this paper represent separating structures as proposed by Wiebel et al. [14].
Traditional approaches of stream volume generation use a closed polygon as rake for stream surfaces and are hardly more than an adopted version of the stream surface algorithms introduced by Hultquist. However, there are more sophisticated methods based on scalar field generation [15] or tetrahedral volumes [2]. This work introduces a novel surface-based algorithm to create watertight stream volumes from parts of separatrices, making knowledge about the starting and ending position of the volume unnecessary.

3 Algorithm

3.1 Outline

These are the basic steps of our algorithm:

1. Compute separation and attachment lines on flow obstacles.
2. Generate three-dimensional separatrices by construction of stream surfaces starting at separation and attachment lines (forwards and backwards, resp.).
3. Intersect separatrices and split them into multiple surface segments.
4. Compose stream volumes of parts of separatrices.

In the following sections, these steps are explained in detail.

3.2 Methods

Computation of Separation Lines. Separation and attachment lines define locations on the two-dimensional projection of the vector field onto the boundary object where flow separates from an object, or attaches to it. In the following *separation line* will be used to denote both types.
For the given mixing simulation we assume, that all significant separation lines are located on the triangulation of obstacles. Such triangles separate incoming flow in one of two ways:

Points with Flow Parallel to Eigenvectors. A method proposed by Kenwright et al. [9] finds points of separation lines on edges of triangles. This is done by eigenvector analysis of the Jacobian of the two-dimensional projected vector field. Points on a separation line are classified by flow parallel to the direction of one of the Jacobian's real eigenvectors. If this eigenvector corresponds to the smallest real eigenvalue, the point is on an attachment line. If it corresponds to the greatest eigenvalue, a point on a separation line was found.

Separating Edges. Convex edges between triangles of obstacles may have separating properties. The two variables listed in the following determine, whether a point p on the edge between two adjacent triangles Δ_i and Δ_j separates the flow.

$$d_1 = ((\Delta_i.p_1 - p) \times \Delta_i.n) \cdot f(p)$$
$$d_2 = ((\Delta_j.p_1 - p) \times \Delta_j.n) \cdot f(p)$$

If d_1 and d_2 do have the same sign as illustrated by Fig. 2, point p has the desired separating properties. This examination allows classification of edges by the separating behavior of their adjacent corner vectors.

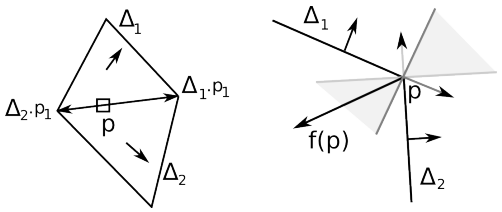

Fig. 2. Flow separation at convex edge. Regions, where d_1 and d_2 have matching signs are marked in gray.

The separation lines of our test data are located near the sharp edges of the mixing blades. These separation lines will be used as rakes for stream surfaces representing three-dimensional separatrices. As our data set does not include saddles, cycles and conventional sinks and sources, separatrices originate from separation lines only.

Integration of Stream Lines. Stream line integration to construct surfaces in grid-less flow simulations yields certain challenges:

Visibility of Data Points. In order to avoid inclusion of points during the MLS approximation of the vector field that are not visible from the position of evaluation, a visibility check needs to be implemented. Grid-based approaches to vector field visualization generally avoid such calculations, as boundary elements are integrated into computational meshes. Figure 3 shows the effects of visibility on stream line integration. To check for a visibility block between points x and x', we find intersections of the line x-x' with triangles of the boundary object.

Integration along Boundary Geometry. Stream lines may start on geometry or cause intersections with it due to numerical inaccuracies of the MLS approximation. Both cases require the ability to integrate stream lines on geometry.
To guarantee watertight surface construction, stream lines that are integrated on the boundary object are forced to stay on geometry until they meet with

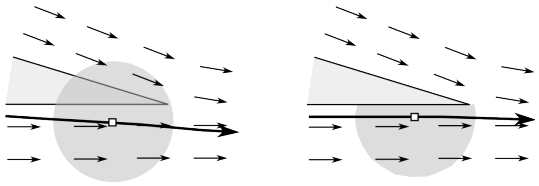

Fig. 3. Impact of visibility on stream lines.

a separation line. A basic approach to stream line integration on triangulated geometry assumes, that every position on a stream line is located on exactly one triangle t. A new triangle is entered as soon as the stream line leaves its current triangle i.e. the stream line crosses one of the edges of t. The appropriate neighboring triangle is chosen as new plane of projection. While this simple method works in most common cases, it however ignores some of the more complex situations illustrated in Fig. 4. Stream line integration on triangles of two-dimensional

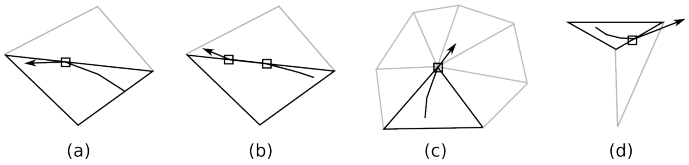

Fig. 4. Situations a) and b) show tangential behavior at triangle edges. In situation c), a stream line crosses one of the corners of a triangle, continuing on only one of multiple neighboring triangles. Situation d) depicts a stream line leaving its current triangle, being released on a convex edge. There are numerous similar situations, where the simple approach mentioned above falls short and oscillation might occur.

linear vector fields has been presented before by Battke et al. [1] in 1997. Our projective approach with focus on edge cases is presented in the following. Let p be the position of a stream line on geometry, with a list $T = \{t_i\}$ of triangles directly adjacent to this position. This method repeats the following steps:

1. **Evaluation of the vector field.** The vector field is evaluated at point p, being located on at least one triangle. Due to the approximating properties of MLS, impact of the no-slip condition is reduced, as the vector field does not evaluate to zero at its boundary.
2. **Projection.** The resulting vector v is projected onto all triangles $t_i \in T$.
3. **Choice of Next Point and Determining Triangles.** There are three possible cases. Either p is located on exactly one triangle, on an edge between two triangles, or on multiple triangles. p is advanced to $p + v_i$ for the appropriate triangle t_i or the intersection of this line with any of the edges of t_i and appended to the stream line. T is updated accordingly.

Degree reduction. If a stream line reaches a point where the number of neighboring data points is not sufficient to provide a uniquely solvable LSE for the evaluation of MLS, the polynomial degree is reduced, as lower degrees require fewer non-coplanar data points. These situations occur especially close to boundaries of the data set. When integrating on or near geometry, extrapolation may flip the vector field's orientation, what can be avoided by decreasing the degree of integration to zero. With geometric integration being unsteady and restricted to a low order due to the low number of available points in the neighborhood, this represents a valid and fast alternative to complex point-in-volume checks.

Construction of Stream Surfaces. One special case that arises, when constructing stream surfaces in vector fields with obstacles is the event of surface splitting. If neighboring stream lines diverge to different sides of an obstacle, triangulation of this stream line pair is canceled. This results in two different fronts of the stream surface. To take advantage of caching strategies, those two fronts are advanced separately and sequentially (see Fig. 5). If at least one of the two affected stream lines is not integrated on the boundary object, a new stream line is inserted on the boundary geometry to maintain a closed representation of the stream surface.
For volume creation, stream surfaces are generated at every separation line of the data set. Resulting separatrices are combined to stream volumes as explained in the following.

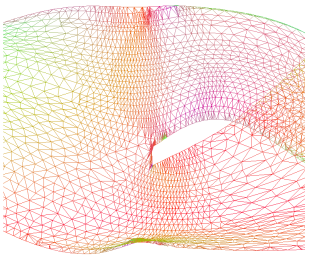

Fig. 5. Obstacle splitting surface. Fronts are advanced separately.

Intersecting Separatrices. An overall look at the division of space is provided by the generation of separatrices as mentioned earlier. To observe one distinct volume at a time, these separatrices need to be split and reorganized to form closed boundaries of stream volumes. As illustrated in Fig. 6, in directed vector fields such as the mixing process considered in this work, intersections resulting in the splitting of stream surfaces can be of two different types:
Intersections caused by separatrices of opposing directions are **crossings** of sep-

aratrices, which cannot occur between separatrices of identical direction[3], as stream surfaces are not able to cross each other if their rakes do not.
The remaining intersections are **T-intersections** rather than x-crossings. They occur whenever an outer stream line of a separatrix, that is integrated along the boundary geometry meets with the rake of a separatrix of the same direction.

Surface Crossings. If two surfaces are crossing, so do their triangulations. Hence it is possible to operate on their triangulations when determining cuts of surfaces, rather than generating and inserting new stream lines into both surfaces to represent the cut. While the former method is less accurate than the latter, intersection of triangulations is computationally far less expensive than according stream line integration and insertion into the complex structure of stream surfaces. In this paper, former method was chosen to create cuts between surfaces of opposing directions. As soon as the triangle-based intersection is computed and affected surfaces are retriangulated, they are split into multiple parts along the trace of their cut. Surface crossings generally divide two surfaces into a total of four parts (see Fig. 7).

T-Intersections The second type of cut is produced between two surfaces of the same direction, where the left- or rightmost stream line s_{outer} of a surface S_1 leaves geometry at the rake of another surface S_2, whose rake represents a separation line.
This type of intersection cannot be handled by a straightforward cutting of the two triangulations. One reason for this is that the roots of S_2 do not yield a one-to-one representation of the original separation line due to rake discretization during the process of surface creation. As the starting points of s_{outer} and stream lines of S_2 usually differ, it is impossible to guarantee a continuous intersection. An advantage over the crossing situation is the knowledge about the meeting point of s_{outer} with the rake of S_2. This point is identical to the position, where s_{outer} leaves geometry. Insertion of a stream line starting at this point that directly represents the cutting trace on S_2 becomes feasible in this case. This newly generated stream line keeps all data about the surface intersection that is needed for retriangulation and splitting of the affected surface.

Composition of Stream Volumes. Previous work on stream volumes is not suitable to create the volumes desired in the context of this work, as no data about the starting or ending regions of volumes is available. Therefore a new approach of volume composition is introduced in the following.
Surface parts that originate from a common intersection yield a certain relation. As shown in Fig. 7, this relationship is governed by the normals of two adjacent triangles that are part of the intersection. These normals allow classification of the newly created surfaces (in this case four) by orientation of their normals, resulting in a neighborhood structure. A back-back neighborhood between two surfaces s_1 and s_2 with triangles Δ_1 and Δ_2 is for example indicated by:

[3] backwards or forwards in the vector field

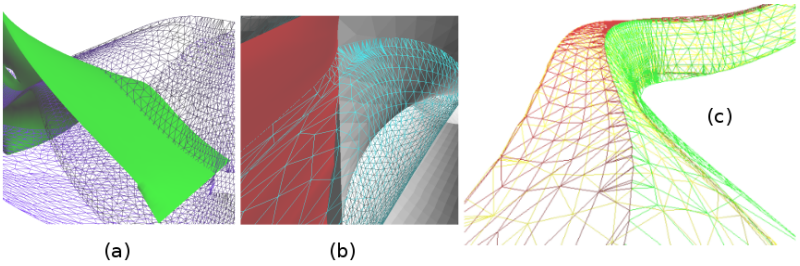

Fig. 6. Crossing behavior of stream surfaces (a), and t-intersection (b) between outer stream line and red surface. Stream volume composed of parts of three separatrices (c). Retriangulation during surface splitting guarantees a closed volume.

$$d_1 = \Delta_1.n \cdot (\Delta_2.o - \Delta_1.o) < 0$$
$$d_2 = \Delta_2.n \cdot (\Delta_2.o - \Delta_1.o) > 0$$

Where $\Delta_1.o$ and $\Delta_2.o$ denote triangle centers. Assembling of a stream volume follows these steps:

1. Choose an arbitrary stream surface.
2. Choose whether the surface faces the inside or the outside of the volume to be generated. Therefore indicating, whether the normal points into the volume or not.
3. Save a list of all relevant intersections of this surface.
4. Add all surfaces to the volume that share any of the relevant stored intersections and face in the correct direction as defined by d_1 and d_2. Intersections of the new surface are processed and added to the list of intersections.
5. Repeat steps 3-4 until no more surfaces are added.

After all five steps are completed, every surface contributing to the volume is assigned at most two volume indices.

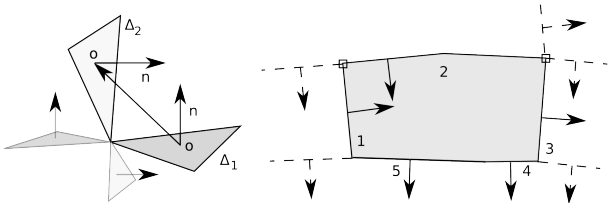

Fig. 7. Neighboring triangles of different surfaces (left) and composition with intersected surfaces (right).

Constructing Slicing Planes. Slicing planes are traditionally used in volume visualization of scalar data sets. Such textured planes cut through a data set and are colored by the scalar values associated with the data. A visualization of the volume is obtained by placing parallel, transparent planes throughout the data set. A similar method is used in this work to avoid occlusion when visualizing stream volumes. The boundary and surface intersections with slicing planes are found and projected onto the plane. While the boundary object of the test data set may produce multiple outlines on the plane, intersections between the slicing plane and volumes consist of cuts with all surfaces of a certain stream volume, resulting in closed area representations.
During visualization slicing planes are rendered as RGBA textured quads with outlines and cuts mapped and plotted using Bresenham's line algorithm [3]. This produces a slicing plane displaying all intersections with geometry. To avoid ambiguity, the interior area of a volume cut is colored in a distinct color.

4 Results

4.1 Application

As expected due to the number of obstacles, $2^6 = 64$ volumes are created in our data set. These volumes consist of a total of 264 surfaces, implying that every separatrix was on average divided into 12 parts.
Figure 9 depicts visualizations of a single stream volume and the complete set of 64 volumes. Inspection of multiple volumes, as depicted in Fig. 8 provides more general information on the segmentation of flow. Figure 10 displays slicing planes through volumes of the data set. Figure 11 illustrates the use of transparent slicing planes for volume visualization. All slicing planes are aligned perpendicular to the main direction of flow.

4.2 Discussion

Stream volumes provide a general overview of the quality of a mixing process, as their form and start/end positions contain information about sets of particle traces. Slicing planes simplify analysis by extracting local information about stream volumes. This way multiple observations can be made using the visualization techniques presented in this paper: The simulation has a symmetrical rotating behavior, incoming groups of stream lines are rotated by at least 90°. The cuts in Fig. 10 reveal a shuffling or squeezing motion caused by the vector field, resulting in volume deformations. Additionally, a comparison between cuts through either end of the simulation suggests a mixing property, as neighboring stream volumes describe differing paths. These results suggest, that the mixing process is of good quality. As is seen by analysis of flow obstacles, the mixing character is directly influenced by the number of obstacles. The desired number of obstacles depends on the concrete application of the mixing process.

Future work in this direction may include parallelization of intersection operations, analysis of volume divergence and extension of the work to non-stationary fields.

Fig. 8. Visualization of several stream volumes. A stretching, rotating and diverging motion between adjacent volumes can be observed.

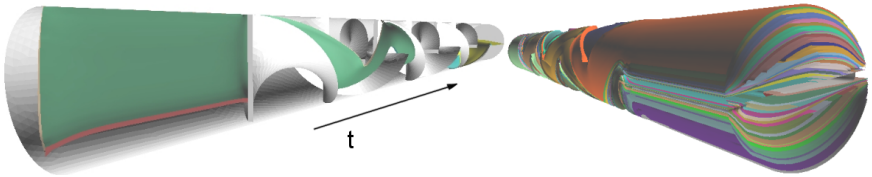

Fig. 9. Visualizations of a single twisted and stretched volume and all volumes of the data set. Colors identify adjacent stream volumes. Occlusion of volumes hides important information about the mixing process, which can be visualized by slicing.

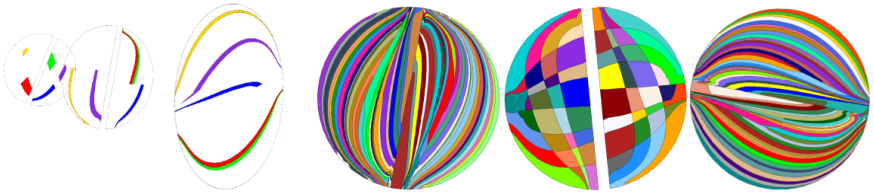

Fig. 10. Slicing planes through five (left) and all volumes of the data set (right) at positions $t = 0.3, 0.6, 1.0, 0, 0.5, 1.0$. It is clearly visible, how neighboring volumes take different paths through the data set. Comparison between individual slices allows observation of stretching, rotational, and mixing behavior. The distribution of volume colors indicate a good mixing process.

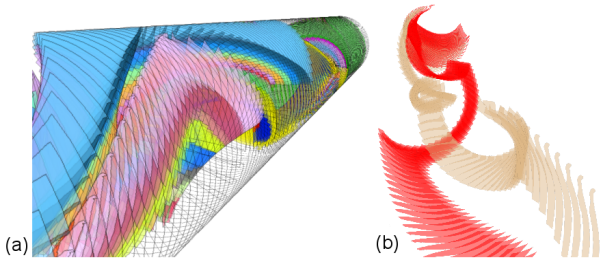

Fig. 11. Volume visualization of a selection of stream volumes by slicing planes (a) and volume visualization of two stream volumes (b). This type of visualization lessens the effect of volume occlusion by the use of transparency.

Acknowledgments This work was supported by the German Research Foundation (IRTG 1131) and by the Center for Mathematical and Computational Modeling $(CM)^2$.

References

1. H. Battke, D. Stalling, H-C. Hege: *Fast line integral convolution for arbitrary surfaces in 3D.* Visualization and mathematics: experiments, simulations and environments, 1997, 181–ff
2. B. G. Becker, D. A. Lane, N. L. Max: *Unsteady Flow Volumes.* Proceedings of Visualization 1995, 329
3. J. E. Bresenham: *Algorithm for Computer Control of a Digital Plotter.* IBM Systems Journal, Vol. 4, No. 1, 25–30 (1965)
4. C. Garth, H. Krishnan, X. Tricoche, T. Bobach, K. I. Joy. *Generation of Accurate Integral Surfaces in Time-Dependent Vector Fields.* IEEE TVCG, Vol. 14, No. 6, 1404–1411 (2008)
5. A. Globus, C. Levit, T. Lasinski: *A Tool for Visualizing the Topology of Three-Dimensional Vector Fields.* NASA Ames Research Center, Computer Sciences Corporation
6. J. P. M. Hultquist: *Constructing stream surfaces in steady 3D vector fields.* Proceedings of Visualization 1992, 173–175
7. D. Levin: *The Approximation Power of Moving Least-Squares.* Mathematics of Computation, Vol. 67, No. 224. October 1998, 1517–1531
8. A. Nealen: *An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Data Approximation and Interpolation.* Technical Report, Discrete Geometric Modeling Group, TU Darmstadt
9. D. N. Kenwright, C. Henze, C. Levit: *Feature Extraction of Separation and Attachment Lines.* IEEE TVCG 5, 2 1999, 135–144
10. G. Scheuermann, T. Bobach, H. Hagen, K. Mahrous, B. Hamann, K. I. Joy, W. Kollmann: *A tetrahedra-based stream surface algorithm.* Proceedings of IEEE Visualization 2001, 151–158
11. S. U. Sikorski, C. K. Ma: *Efficient Streamline, Streamribbon, and Streamtube Constructions on Unstructured Grids.* IEEE TVCG, Volume 2, No. 2, 100–110 (1996)
12. M. Spiegelman: *Myths and Methods in Modeling.* LDEO, Columbia University. chapter 4 (2000)
13. S. Tiwari, J. Kuhnert: *A numerical scheme for solving incompressible and low mach number flows by Finite Pointset method.* Technical Report, Fraunhofer ITWM, Kaiserslautern
14. A. Wiebel, X. Tricoche, G. Scheuermann: *Extraction of Separation Manifolds using Topological Structures in Flow Cross Sections.* Topology-Based Methods in Visualization II, Springer (2009)
15. J. J. van Wijk: *Implicit Stream Surfaces.* Proceedings of the 4th Conference on Visualization 1993, 245–252