

Glyphs for Non-Linear Vector Field Singularities

Alexander Wiebel, Stefan Koch, Gerik Scheuermann

Abstract Glyphs are a widespread technique to depict local properties of different kinds of fields. In this paper we present a new glyph for singularities in non-linear vector fields. We do not simply show the properties of the derivative at the singularities as most previous methods do, but instead illustrate the behavior that goes beyond the local linear approximation. We improve the concept of linear neighborhoods to determine the size of the vicinity from which we derive the data for the glyph. To obtain data from outside this neighborhood we use integration in the vector field. The gathered information is used to depict convergence and divergence of the flow, and non-linear behavior in general. These properties are communicated by color, radius, the overall shape of the glyphs and streamlets on their surface. This way we achieve a depiction of the non-linear behavior of the flow around the singularities.

1 Introduction

Vector field singularities (also called critical points), i.e. zeros in the vector field, are a central ingredient to the analysis of vector field topology [10]. In two-dimensional as well as three-dimensional vector fields they are the locations at which separatrices (lines in 2D, surfaces in 3D) originate or end. These separatrices partition the field into areas with streamlines that have the same origin and destination (α and ω limit sets) and thus allow to divide the fields into areas with similar behavior. Unfortunately, this partition is only useful for time-independent vector fields or instantaneous snapshots of time-dependent vector fields. When considering complete time-dependent vector fields the critical points lose their importance concerning a

Alexander Wiebel
Zuse-Institut Berlin, Takustr. 7, D-14195 Berlin-Dahlem, Deutschland. e-mail: wiebel@zib.de

Stefan Koch and Gerik Scheuermann
Universität Leipzig, Institut für Informatik, PF 100920, 04009 Leipzig, Deutschland.
e-mail: {stefan.koch, scheuermann}@informatik.uni-leipzig.de

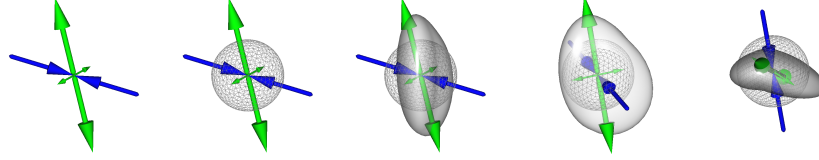


Fig. 1 From left to right: Saddle point, saddle Point with seeding sphere, saddle point with seeding sphere and deformed sphere illustrating the flow behavior, other perspective of previous image and another perspective of the same image. Note how the glyph (deformed sphere) illustrates the non-linear behavior around the saddle point; linear behavior would produce an ellipsoid (see 3.2).

partition of the space. Still, for time-dependent fields, they are important in the development of certain flow features. The usefulness of their tracking and visualization has for example been shown in the context of vortex breakdown [7].

The obvious importance of singularities for the overall flow behavior results in a need for the analysis of the flow in the vicinity of such singularities. In the past vector field singularities themselves have been visualized mainly by showing either only their location with a dot or a sphere, or with glyphs illustrating the nature of their linear approximation. A prominent example of the latter are the iconic representations of the repelling/attracting and rotating nature of the field around the singularities that can be found in a paper by Theisel et al. [21]. They go beyond the common visualization showing only arrows for the directions of the eigenvectors of the linear approximation (Jacobian matrix).

In this paper we want to go even further and illustrate also the non-linear behavior of the field around a singularity. Therefore we introduce glyphs (see figure 1) whose shape and coloring provide information about the behavior of the flow gathered during a short time integration in the field. In particular, the main contributions of this paper are

- an improved and robust computation of the *linear neighborhood* that was introduced to visualization by Schneider et al. [19],
- a new type of glyphs that is seeded according to this *linear neighborhood* and that describes the non-linear behavior of vector field singularities not only in the directions of the eigen-manifolds,
- and a demonstration of different additions to the glyphs (coloring, FTLE and streamlets) that emphasize special characteristics of the depicted flow.

As we will show throughout this paper, the introduced techniques can help to improve visualization, understanding and analysis of vector field singularities and their surrounding flow. Please note that we do not aim at specific new findings in application areas. Our method is simply intended to provide additional and complementary views on the vicinity of the singularities. We explicitly recommend a combination with other methods (e.g. [15] and [16]) described in section 2.

2 Related Work

Glyphs are very common in the visualization of vector fields. One of the first techniques uses arrows to depict the vectors themselves. It is not attributable a specific author. The same holds for the use of tripods to depict the linear behavior of critical points. A more advanced glyph shown by Theisel et al. [21] was already mentioned in the introduction. Weinkauff et al. [22] presented an icon for higher order vector field singularities. De Leeuw and van Wijk [3] combine a collection of iconic representations for divergence, rotation, shear, curvature, acceleration and the vector itself to form a glyph called *local flow probe*. All properties shown by the local flow probe are derived from the linear approximation of the field. Additionally, the display of the flow direction and speed, would not work for singular points as the field is zero there. Recently, Hentschel et al. [11] successfully demonstrated the use of glyphs in virtual reality to show the deformation of blood cells in a blood flow.

The technique with the most similarities to our work was reported by Löffelmann et al. [15]. They present abstract depictions of the eigen-manifolds of some types of singularities and streamlets on spheres around singularities to illustrate the dynamical system in the vicinity of the points. Löffelmann and Gröller improved this method by adding streamlets around the characteristic curves of the singularities [16]. In contrast to these methods which focus on the eigen-manifolds of the singularities, the glyphs produced by our method give an impression of a singularity's vicinity in all directions without the need of integral lines started on a sphere around the singularity. We consider the integral line alternative (e.g. the sphere tufts [15]) to yield significantly more cluttered images. Another important difference is that our method can illustrate the interaction of close singularities (see e.g. figure 7).

While all above methods dealing with singularities require them to be hyperbolic, hyperbolicity is irrelevant for the construction of the presented glyphs.

Recently glyphs have become very popular to illustrate results from diffusion weighted magnetic resonance imaging [12, 4]. In this context a large number of different glyph types that show the diffusion strengths in the different directions as deformed sphere surfaces has been developed [1, 5, 17]. Our approach is inspired by this type of visualization.

As we advect a surface around a singularity to obtain the glyph, flow volumes [2] by Becker et al. are also related. However, they do not pay special attention to the characteristics of singularities and their algorithm is much more complex than our basic advection because of the volumetric subdivision. For long integration times our method could also need refinement. The sphere can then be interpreted as a time surface and advected as described by Krishnan et al. [14]. Note, however, that long integration are not intended for the glyphs themselves.

3 Mathematical Foundations

3.1 Linear Neighborhood

For our algorithm we need the concept of a linear neighborhood around a critical point as introduced by Schneider et al. [19]. We define the *linear neighborhood* $U_L(\mathbf{x}_c)$ around a critical point $\mathbf{x}_c \in \mathbb{R}^3$ as the region for which a linear approximation of the vector field holds within a certain bound $C_L \in \mathbb{R}, C_L > 0$:

$$U_L(\mathbf{x}_c) = \left\{ \mathbf{y} \in \mathbb{R}^3 \mid \frac{\|\mathbf{v}(\mathbf{y}) - J(\mathbf{x}_c) \cdot (\mathbf{y} - \mathbf{x}_c)\|}{\|\mathbf{v}(\mathbf{y})\|} < C_L \right\} \quad (1)$$

where J denotes the Jacobian. We describe the computation of the neighborhood in section 4.

3.2 Note on Ellipsoids in Linear Vector Fields

We note that the application of the advection scheme we use to produce the glyphs leads to the well known ellipsoidal glyphs in linear vector fields with one isolated singularity¹. Thus, glyphs for such vector fields can directly be derived from their Jacobian matrix defining the whole field. The comparatively costly step of advecting the sphere is not necessary. This holds to some degree also for the field in the linear neighborhood around a singularity. For any sensible choice of the linearity threshold the difference to an ellipsoid will not be visible.

The fact that ellipsoids always transform to ellipsoids in linear vector fields may not be obvious. Therefore we provide a short discussion in the following. We consider the changes of an ellipsoid under an integral line approximation algorithm because this is what we use to advect the vertices of the sphere. In the following, we use the Euler approximation method for simplicity. It is given by $\mathbf{y}_{n+1} = \mathbf{y}_n + h \mathbf{v}(\mathbf{y}_n)$ where \mathbf{y}_0 is the start position, h the step size and \mathbf{v} the vector field. In a linear vector field this can be written as

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h A \mathbf{y}_n \quad (2)$$

where A is the matrix defining the linear vector field and its Jacobian matrix at the same time. Consider all \mathbf{y}_0 that make up the sphere (which is a special ellipsoid). If we add the positions of an ellipsoid to those of another ellipsoid the resulting positions will again fulfill the quadratic equation defining an ellipsoid, i.e. form an ellipsoid. The same holds for scaling all positions of an ellipsoid. Together this means, that we need to show that all $A \mathbf{y}_n$ lie on an ellipsoid if all \mathbf{y}_n lie on an ellipsoid

¹ Linear vector fields with one isolated singularity are defined by invertible matrices.

in order to show that equation (2) transforms another ellipsoid to an ellipsoid. Proofs for this statement are available in the literature, see e.g. Hyslop [13].

4 Linear Neighborhood Computation

In contrast to Schneider et al. [19] who only need to guarantee that their points are somewhere in the linear neighborhood for some of their operations, we are explicitly interested in the radius of the largest sphere lying completely in the linear neighborhood. Inside this sphere all positions adhere to the linearity condition. We call the radius of this sphere the radius of the linear neighborhood.

As vector fields are still mostly stored as discrete samples at vertices (building up a grid of cells for interpolation), we compute the sphere using a cell-based procedure outlined by algorithm 1.

```

input : Position of singularity: singularityPosition
output: Radius of linear neighborhood: radius
cell ← SearchCell (singularityPosition);
positions ← GetPositions (cell);
posFound ← false;
while not posFound do
  for pos ∈ positions do
    if CheckLinearity (pos) then
      add GetNeighboringPositions (pos) to newPositions;
      // no duplicates in newPositions
    else
      posFound ← true;
      outsidePositions ← pos;
    end
  end
  positions ← newPositions;
end
pos ← FindPositionWithMinDistToSing (outsidePositions,singPos);
minimalRadiusFound ← false;
while not minimalRadiusFound do
  radius ← GetMinLinearRadius (singularityPosition,pos);
  minimalRadiusFound ← true;
  for pos ∈ discretized ball with radius do
    if not CheckLinearity (pos) then
      minimalRadiusFound ← false;
      break;
    end
  end
end
return radius;

```

Algorithm 1: Computing the radius of the linear neighborhood.

The linearity test `CheckLinearity` essentially checks whether the condition in (1) holds for a certain position. `FindPositionWithMinDistToSing` selects the position from the given set which is closest to the singularity. After this call the grid vertex violating the linearity condition which is closest to the singularity has been found. The distance of this vertex to the singularity is an upper bound for the radius of the linear neighborhood. The second part of the algorithm is concerned with refining the radius. `GetMinLinearRadius` determines the specific point on a ray between the vertex and the singularity which first violates the linearity condition. Therefore a binary search² is performed. The search is confined to the cell which contains the vertex and which is crossed by the ray. Other cells crossed by the ray are guaranteed to lie completely in the linear neighborhood.

The procedure so far would already compute a reasonable approximation of the radius for most cases. To handle also all other cases we check whether the linearity condition holds for all vertices of a discretized sphere with the radius determined so far. If the check fails for the first time we use the current vertex to reiterate the binary search on a ray from this vertex to the singularity. Again, we get an approximation of the radius and check it on a discretized sphere. We repeat this until we have found a radius for which no vertex on the sphere violates the linearity condition. This radius is the radius of the linear neighborhood. The iteration is guaranteed to terminate, because the radius can only decrease in each step, each step uses a separate binary search and there is a natural minimum (i.e. zero).

5 The Glyphs

5.1 Glyph Generation

As a starting point for the glyph computation, we place spheres with a fixed radius around the singularities of the vector field. The centers of the spheres are exactly at the position of the singularities. The spheres are approximated by repeated subdivision of icosahedra, so that we obtain triangular discretizations for all spheres. The vertices of the triangulations of the start spheres serve as starting points for a integral line computation over a certain time interval. The endpoints of these particle integrations are stored in a so called flow map vector field, which means that the endpoint of every integral line is stored at the corresponding vertex. The simplest version of the glyphs, i.e. the spheres advected respectively deformed by the flow surrounding the singularities, can then be obtained by substituting all sphere vertices by the corresponding flow map entries.

² A binary search is suitable here because the error in the linearity condition varies monotonically along the ray. This is immediately clear for tetrahedral cells because linear interpolation is used for them. For other cells, i.e. trilinearly interpolated cells, the monotonicity is not given for the whole cell. However, as the line starts from the “non-linear” vertex closest to the singularity and as the extrema in a trilinear cell lie on the boundary, the ray can only cross monotonic regions.

Because the choice of the initial sphere radius and the integration time are crucial for easily interpretable and meaningful glyphs, we will now discuss different radius determination approaches.

5.1.1 Choice of Initial Sphere Radii and C_L

The simplest choice is a user defined radius for all glyphs. This gives the user maximum freedom but may be not the optimal solution concerning depiction of non-linearity.

If the user is not sure which initial sphere size is suitable for the flow field, the initial radius of each sphere may be determined semi-automatically. For this purpose we employ the linear neighborhood introduced in section 3.1, i.e. we use the linear neighborhood spheres as start spheres. Only the parameter C_L has to be given by the user. It indicates how non-linear the field has to be at the starting sphere. The resulting radii are meaningful for each glyph separately, because they will reflect exactly the degree of non-linear flow behavior the user aims to illustrate around each singularity. This is guaranteed as the advection then starts where a certain degree of non-linearity starts. According to the considerations in section 3.2 very small C_L will lead to nearly ellipsoidal glyphs. Note, that the semi-automatic choice results in separate starting radii for the different singularities.

5.1.2 Choice of Integration Time

Besides the choice of the radii for the initial spheres, the integration time for the integral line computations is very important to obtain meaningful glyphs. Again, users can either select the value manually or use an automatically computed integration length for each single initial sphere.

We determine the automatic integration time such that the vertices of a starting sphere are advected approximately a distance (arc length) that is equal to half the radius of that sphere. In fact, we determine the median integration time all vertices of the sphere need to travel this prescribed arc-length. This integration is then used to advect all vertices of the sphere to their final positions. Thus the glyphs are between half and 1.5 times the size of the starting sphere.

5.2 Rendering Styles

To elevate the characteristic of the glyphs we provide different color maps and a glyph evolution.

5.2.1 Color Maps

Color mapping on the glyphs can support the perception of the deformation of the initial sphere through the non-linear flow behavior and can be used to add additional information. We provide two color maps for the first and one color map for the latter task.

First, the value range of the radii can be mapped on our glyphs. As a result of this, we obtain depictions of large values at the positions on the glyphs which have a large distance to their singularity and vice versa we depict small values at the positions that are close to the singularity. Therefore, this color map emphasizes the perceptions of the glyph shape and supports its fast and easy interpretation by the user.

The second color map we provide makes a statement about the deformation of the glyph relative to the initial sphere. For that purpose, we subtract the radius at all positions of a glyph from their actual distance to the singularity. Thus, we obtain a positive value if a position moved outside the initial sphere and a negative value if the position moved to the inside. Hence, the user is able to get a quick overview of the glyph shapes relative to the initial spheres without the need to render the spheres.

Coloring glyphs with Lagrangian flow properties depicts divergence and convergence of the flow in the different directions around the singularity. Therefore, we adapt the surface FTLE technique introduced by Garth et al. [8]. In their paper they present FTLE computed for surfaces with a small offset from the boundary of flow embedded surface structures to depict separation of flows. We compute the surface FTLE for our starting spheres. The resulting scalar field on the glyph is mapped to color. An example of this version of our glyphs is shown in figure 2 in the right images. Large values indicate a strong stretching of the spherical surface.

Note that the computation of the FTLE causes only a very small amount of extra computation time because the most expensive part, i.e. the advection of the vertices of the starting sphere, is already done for the construction of the glyphs. Only the evaluation of the FTLE values from the flow map has to be performed additionally. For details on the FTLE computation on surfaces we refer the reader to the mentioned paper [8] and for more in depth introduction of FTLE itself to works by Shadden et al. [20] and Haller [9].

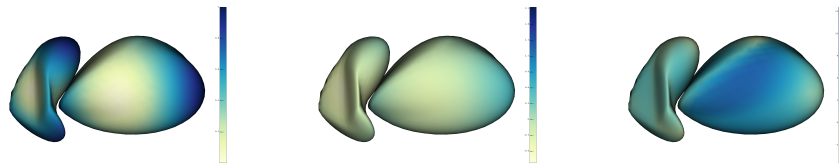


Fig. 2 The images show a selection of color maps on glyphs for two close singularities (from left to right): normalized distance to singularity, signed distance to start position and $FTLE^+$.



Fig. 3 Evolution of a glyph over time. Four steps between initial sphere and final integration time. The arrows indicate the linear nature of the singularity for reference. The right image is a high resolution version of the glyph in the left image with color mapping.

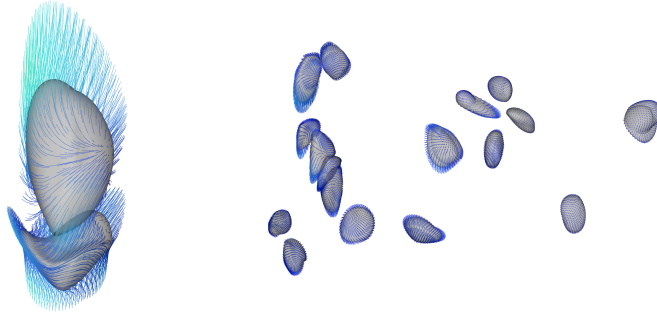


Fig. 4 Glyphs with streamlets emphasizing the flow direction. Left: Pair of singularity glyphs. Right: Overview of a groups of singularities in the gas furnace chamber data set.

5.2.2 Glyph Evolution

Besides the just introduced glyph shapes and color maps, another way to give a deeper insight into the non-linear flow is to illustrate the deformation of the initial sphere by showing the evolution of the glyphs over their integration time.

Therefore, we provide the option to interrupt the glyph integration at equidistant time steps and store the intermediate shapes. Displaying all intermediate shape with a color coding of the different advection times provides an instructive visualization. An example for this illustration is given in figure 3.

5.3 Streamlets

In general, our singularity glyphs do not represent rotational flow behavior well. The color map encoding the movement of the different vertices can give a first hint in this regard. However, we generate an additional cue for the flow direction. Therefore we use all glyph vertices as seeding positions for streamlets, i.e. short streamlines (or path lines in the unsteady case). As integration time we choose half the integration as for the glyph surface. The streamlets provide the desired directional cues without obstructing the visibility of the glyph surface. This is demonstrated in figure 4.

In contrast to Löffelmann et al. [15] who seed their streamlets stochastically on a sphere, we use the triangulation of the glyph for seeding. This strategy increases the streamlet density where the flow converges and decreases it where the flow diverges, thus providing additional information about the flow behavior.

5.4 *Steady vs. Unsteady Flow*

Up to now we described the technique without any focus on time-dependent or time-independent vector fields. Fortunately, the whole idea applies to time-dependent as well as time-independent fields without any major changes. The advection of the sphere vertices is done exactly the same way. The only difference is that their traces are path lines instead of streamlines in the unsteady case.

As we discussed earlier, singularities in time-dependent fields have a somewhat different meaning than in steady fields. Thus they deserve some special comments. First, we can simply use instantaneous vector field singularities as in the steady case. However, in the unsteady case there also exist other definitions of meaningful critical points. The most recent definition are the motion compensated critical points given by Fuchs et al. [6]. They are not the obvious singularities in the vector field, but vector field singularities in a specially chosen local frame of reference. As these singularities are also given as simple points, they fit well into our approach and can be used as centers for the start spheres. The advection of the points can then be done either in the originally given field or in the field in the local frame of reference. The choice depends on the desired meaning: the first simply depicts the flow around the critical point the second puts a special emphasis on the critical points influence on its neighborhood.

5.5 *Implementation Details*

The basis for the glyphs are spheres that have been approximated with the triangular discretization. As described above, the triangulation is derived by repeated subdivision of an icosahedron. Except where mentioned explicitly, all images in this paper use a subdivision depth of three yielding 1280 triangles for each sphere. This resolution was sufficient to capture all desired details in our experiments, while being coarse enough to be easily handled by any graphics card with simple 3D support.

5.6 *Special Cases of Glyphs*

Figure 5 shows glyphs for singularities close to the boundary. The radius of the start sphere of some of these glyphs (arrows) is larger than their distance to the bound-

ary. Thus some of their points lie outside the data domain. These points need special treatment because they cannot be advected at all. Two possibilities seem useful. One is discarding all triangles containing such a point. This leaves a holes where the problematic points are. The hole nicely indicates the fact of non-advectable points while producing no visual artifacts. Increasing the resolution of the starting spheres yields smaller holes. This looks nicer but very expensive regarding time and memory. The second possibility is to clip the mentioned triangles at the data domain boundary. This results in glyphs nicely attached to the boundary. Jagged parts in theses glyphs indicate shear flow close to the boundary (which is quite common). This is also quite costly in our naive implementation. However, this is only needed for producing final high quality images, and a much more effective implementation is easily achieved by using acceleration data structures like octrees for the intersection step.

6 Application Examples and Discussion

Most of the examples used in this paper are singularities from a vector field representing the flow in a furnace chamber of a central heating system as used in single-family homes. As can be seen in the overview of the dataset given in figure 6, the flow contains many (87) isolated singularities as it is very turbulent. The turbulence in this example is not a problem of the design, it is intended to ensure a good intermixture of air and gas for an effective combustion process.

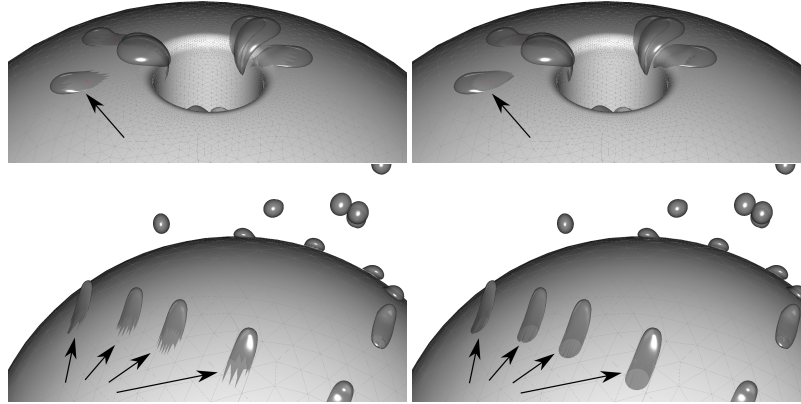


Fig. 5 Singularity glyphs with manually selected radius close to the boundary of a flow embedded ball with a hole. Some glyphs (some marked with arrows) are open on one side because the radius was chosen larger than their distance to the wall. The upper right image shows glyphs with higher resolution than in the upper left image. The lower right image shows the same glyphs as in the lower left image but with enabled clipping.

The large number of singularities in this data set leads to many very close singularity pairs. The interaction of such singularities is nicely visible in figure 7. The flow around the left singularity (a source) in the left image is repelling and the right singularity (a saddle) attracts the flow in one direction. The expansion of the left glyph causing a “dent” in the right glyph illustrates this effect. A similar illustration is shown in the right image where the attracting and repelling parts of two saddle points interact. Such an interaction is not representable with any other previous technique, like for example the eigenvector arrows shown in the same image.

Figure 5 shows glyphs advected in a time-dependent data set. It represents the flow around a ball with a hole through its center. We use illustrate instantaneous vector field singularities taken from a single time step. The special properties of the glyphs in figure 5 have been discussed in section 5.6. We compute all information for all rendering styles in one step before displaying the glyphs. This allows us to switch between the different renderings immediately. The computation time of the preprocessing step ranges from 5 to 30 seconds depending on the chosen glyph parameters for datasets with less than 100 singularities. Our naive special treatment of glyphs close to the boundary (mentioned above) is not included in these numbers. The measurements were performed on one core of an Intel(R) Xeon(R) CPU with 2.40 GHz.

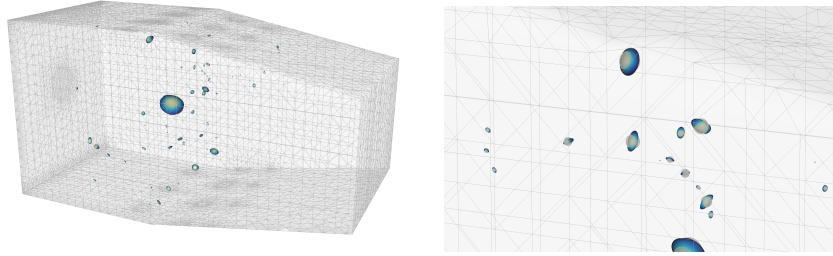


Fig. 6 Left: Overview of glyphs for all singularities of the turbulent furnace chamber dataset. Right: Zoomed in and slightly rotated version of the same visualization reveals the details. All images are produced with automatically determined radius and integration time.

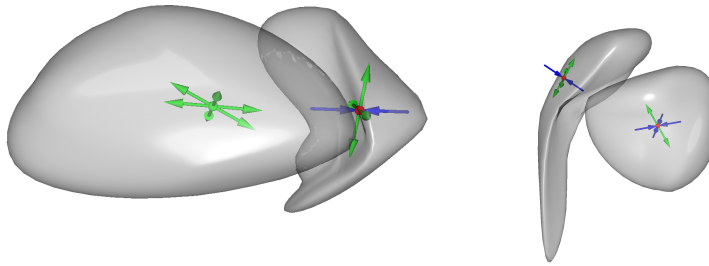


Fig. 7 Pairs of glyphs for close-by singularities with radius larger than linear neighborhood.

7 Conclusion

We have presented a new type of glyphs for vector field singularities that is capable to illustrate a number of the characteristics of the non-linear behavior of the field around the singularities. Users can determine the size and meaning of the glyphs by either prescribing an initial radius or a linearity threshold. Changing the latter they can explore the different degrees of non-linearity around the singularities. The glyphs are applicable for time-independent as well as time-dependent vector fields. The concept can be easily extended to 2D using circles instead of spheres.

In order to keep the flow behavior and its representation by the glyph shape consistent, some limitations of the overall technique are unavoidable. For example, we can not scale the glyphs to a size that would make it possible to see an overview of the whole field with all glyphs in sufficient detail. If we increase the size of the glyphs after the advection step, the consistence is lost. The glyphs would mislead the observer to believe that the flow in the volume covered by the glyph is represented by the glyph although only the flow in a small part of the volume would be responsible for the shape. Thus, we decided to show the glyphs in their original size (which is large enough to be noticed in an overview, see figure 6) and leave it to the user to zoom in to see the details of the local behavior. A future extension simplifying the analysis of the singularities in the context of the whole dataset could be an interface similar to the *interactive closeups* presented in the context of medical exploration by Ropinski et al. [18]. We recommend to show the starting spheres together with the glyphs in general (e.g. figure 1). This allows the user to put the deformation in relation to the original shape. In the paper we have omitted the spheres in most cases.

As future enhancement, we plan to depict the unsteadiness as introduced by Fuchs et al. [6] by a color map on the glyphs. We are already working on an extraction of the exact shape of the linear neighborhood (not as sphere).

Acknowledgements First of all the authors would like to thank the reviewers for their many constructive remarks and suggestions which greatly helped to improve the paper. The authors would like to thank Markus Rütten from DLR in Göttingen for providing the datasets. Thanks also go to Wieland Reich and Roxana Bujack for helpful discussions. Special thanks go to the *FAnToM* development group for providing the environment for the implementation of the presented work. This work was partially supported by DFG grant SCHE 663/3-8. During the course of this work Alexander Wiebel was hired by the Max Planck Institute for Human Cognitive and Brain Sciences in Leipzig. The first two authors contributed equally to this work.

References

1. D. C. Alexander, G. J. Barker, and S. R. Arridge. Detection and modeling of non-gaussian apparent diffusion coefficient profiles in human brain data. *MRM*, pages 331–340, 2002.
2. B. G. Becker, D. A. Lane, and N. L. Max. Unsteady Flow Volumes. In G. M. Nielson and D. Silver, editors, *Proceedings of the 6th Conference on Visualization '95*, pages 329 – 335,

- Washington, DC, USA, 1995. IEEE Computer Society.
3. W. C. de Leeuw and J. J. van Wijk. A probe for local flow field visualization. In *Proceedings of the 4th conference on Visualization '93, VIS '93*, pages 39–45, Washington, DC, USA, 1993. IEEE Computer Society.
 4. M. Domin, S. Langner, N. Hosten, and L. Linsen. Direct glyph-based visualization of diffusion mr data using deformed spheres. In L. Linsen, H. Hagen, and B. Hamann, editors, *Visualization in Medicine and Life Sciences*, pages 177–195, 321. Springer-Verlag, 2007.
 5. L. R. Frank. Characterization of anisotropy in high angular resolution diffusion-weighted MRI. *Magnetic Resonance in Medicine*, 47:1083–1099, 2002.
 6. R. Fuchs, J. Kemmler, B. Schindler, J. Waser, F. Sadlo, H. Hauser, and R. Peikert. Toward a lagrangian vector field topology. *Computer Graphics Forum*, 29(3):1163–1172, 2010.
 7. C. Garth, X. Tricoche, and G. Scheuermann. Tracking of Vector Field Singularities in Unstructured 3D Time-Dependent Datasets. In H. Rushmeier, G. Turk, and J. J. van Wijk, editors, *Proc. of the IEEE Visualization 2004 (VIS'04)*, pages 329 – 336. IEEE Computer Society, October 2004.
 8. C. Garth, A. Wiebel, X. Tricoche, K. Joy, and G. Scheuermann. Lagrangian visualization of flow-embedded surface structures. *Computer Graphics Forum*, 27(3):1007–1014, May 2008.
 9. G. Haller. Lagrangian structures and the rate of strain in a partition of two-dimensional turbulence. *Physics of Fluids*, 13(11), 2001.
 10. J. L. Helman and L. Hesselink. Surface Representations of Two- and Three-Dimensional Fluid Flow Topology. In *VIS '90: Proceedings of the 1st Conference on Visualization '90*, pages 6–13, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.
 11. B. Hentschel, I. Tedjo, M. Probst, M. Wolter, M. Behr, C. H. Bischof, and T. Kuhlen. Interactive blood damage analysis for ventricular assist devices. *IEEE TVCG*, 14(6):1515–1522, 2008.
 12. M. Hlawitschka and G. Scheuermann. Hot-lines: Tracking lines in higher order tensor fields. In *IEEE Visualization*, page 4. IEEE Computer Society, 2005.
 13. J. Hyslop. Linear transformations and geometry. *Edinburgh Math. Notes*, 25:iv–x, 1930.
 14. H. Krishnan, C. Garth, and K. Joy. Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets. In *Proc. of IEEE Visualization*, pages 1267–1274, 2009.
 15. H. Löffelmann, H. Doleisch, and E. Gröller. Visualizing dynamical systems near critical points. In *Proceedings of the Spring Conference on Computer Graphics and its Applications 1998*, pages 175–184, Budmerice, Slovakia, April 1998.
 16. H. Löffelmann and E. Gröller. Enhancing the visualization of characteristic structures in dynamical systems. In D. Bartz, editor, *Visualization in Scientific Computing '98*, pages 95–68. Eurographics, Springer-Verlag Wien New York, 1998.
 17. E. Özarslan and T. H. Mareci. Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magnetic Resonance in Medicine*, 50:955–965, 2003.
 18. T. Ropinski, I. Viola, M. Biermann, H. Hauser, and K. Hinrichs. Multimodal visualization with interactive closeups. In *EGUK Theory and Practice of Computer Graphics*, June 2009.
 19. D. Schneider, W. Reich, A. Wiebel, and G. Scheuermann. Topology aware stream surfaces. *Computer Graphics Forum*, 23(3):1153–1161, June 2010.
 20. 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*, 212:271–304, 2005.
 21. H. Theisel, T. Weinkauff, H.-C. Hege, and H.-P. Seidel. Saddle Connectors - An Approach to Visualizing the Topological Skeleton of Complex 3D Vector Fields. In *Proc. of IEEE Visualization*, pages 225–232, 2003.
 22. T. Weinkauff, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel. Extracting higher order critical points and topological simplification of 3D vector fields. In *Proc. IEEE Visualization 2005*, pages 559–566, Minneapolis, U.S.A., October 2005.