Enhancing the Visualization of Characteristic Structures in Dynamical Systems*

Helwig Löffelmann and Eduard Gröller**

Institute of Computer Graphics, Vienna University of Technology,
Karlsplatz 13/186/2, A-1040 Wien, Austria,
`http://www.cg.tuwien.ac.at/home/`

Abstract. We present a thread of streamlets as a new technique to visualize dynamical systems in three-dimensional space. A trade-off is made between solely visualizing a mathematical abstraction through lower-dimensional manifolds, i.e., characteristic structures such as fixed points, separatrices, etc., and directly encoding the flow through stream lines or stream surfaces. Bundles of streamlets are selectively placed near characteristic trajectories. An over-population of phase space with occlusion problems as a consequence is omitted. On the other hand, information loss is minimized since characteristic structures of the flow are still illustrated in the visualization.
Keywords: visualization, dynamical systems.

1 Introduction

Visualization [14] has become an established field of science during the past years. Dynamical systems, for example, flow fields, are an important topic concerning research in this area [2, 16]. Dynamical systems provide a mathematical framework to deal with the dynamics of a set of variables. They are used to model real world phenomena such as, e.g., the stock market, chemical reactions, or food chains.

A *dynamical system* is usually given by a vector of state variables which change over time [3]. If the formulas which describe the dynamics of the system are varying over time, a dynamical system is called *time-dependent*. If the rules guiding the dynamics are static over time, the dynamical system is called *steady* (time-independent). Usually a *continuous* dynamical system (also called *flow*) is specified by a set of ordinary differential equations (ODEs – $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{p}, t)$) together with a set of parameters ($\mathbf{p}$). Often continuous dynamical systems are visualized in *phase space*, which is defined by associating each of the n state variables to one axis of an n-dimensional Cartesian coordinate system. In this

* `http://www.cg.tuwien.ac.at/research/vis/dynsys/KnitDS97/`

** `mailto:helwig@cg.tuwien.ac.at, mailto:groeller@cg.tuwien.ac.at` paper we will concentrate on 3D continuous dynamical systems which are steady, i.e., function $\mathbf{f}$ does not depend on time t.

Several approaches to the visualization of dynamical system can be distinguished [11]. One class of techniques deals with the visualization of *characteristic elements* such as, e.g., fixed points, cycles, or separatrices. A structure of lower-dimensional objects is composed in phase space to describe the key features of the system's behavior [1]. For example, a separatrix is visualized to indicate two subsets of phase space with qualitatively different dynamics. A brief overview of the relation between local linearization and characteristic structures can be found in the Appendix.

Another class of approaches deals with the *direct visualization* of the system behavior. Integral curves visualize the evolution of specific initial settings which change according to the dynamics of the underlying flow. Many techniques are already available for the 2D case. Spot noise [18] and line integral convolution (LIC) [5], for example, provide an overview of 2D dynamics within a 2D domain. In 3D, however, direct visualization is difficult. Rendered images tend to be overloaded when entire portions of flow in 3D space are simultaneously visualized. Some attempts into this direction are illuminated stream lines [19] and volume-rendered 3D flow [7].

In addition to the visualization of characteristic elements and direct visualization, a third class of techniques deals with the representation of local properties [12]. Glyphs [6] represent certain quantities derived from the Jacobian matrix (local linearization of the flow) such as, e.g., acceleration, rotation, or divergence. Another approach [17] transforms a polygon positioned perpendicular to a trajectory to represent local information.

In this paper we present a technique which to a certain extent belongs to all of the three classes mentioned above. It was inspired by the concept of modeling knit-wear as yarn with a complex micro-structure [8]. We visualize the vicinity of characteristic trajectories, for example, the stream lines emanating from fixed points. A great number of short integral curves (streamlets) is used to directly code the system's behavior near the characteristic trajectory. By this approach of selectively placing streamlets we omit distracting image cluttering while still providing direct cues to the (local) system behavior. Visualizing the vicinity of characteristic stream lines enhances the abstract representation of the system's behavior by local cues of direct visualization.

2 A thread of streamlets

To come up with a useful technique of locally enhanced stream lines, we propose a model for the generation of a *thread of streamlets.* Near a predescribed stream line $\mathcal{T}$ (the *base trajectory*) many short streamlets are placed. Thereby a continuous representation of the system's behavior in the vicinity of the base trajectory is approximated.

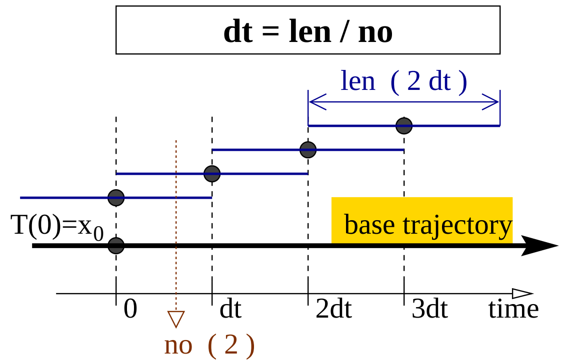


Fig. 1. Relation between streamlet density (no), streamlet integration length (len), and streamlet instantiation interval (dt)

Using constant flow as a reference model – stream lines are straight lines in this case – the thread of streamlets $\{\mathcal{T}_i\}_{i\in\mathbf{N}}$ is defined as follows: Any cross-section perpendicular to base trajectory $\mathcal{T}$ is pierced by a constant number (no) of streamlets. Using integration time t as parameterization of base trajectory $\mathcal{T}$ ($\mathcal{T}(0) = x_0 =$ seed point of $\mathcal{T}$), streamlets $\mathcal{T}_i$ are instantiated at time $t_i = i \cdot dt$ and integrated over the time interval $[i \cdot dt \pm \frac{len}{2}]$. See Fig. 1 for an illustration of the relationship between no, dt, and len, i.e., $dt = len/no$. Seed points $\mathcal{T}_i(i \cdot dt)$ of newly instantiated streamlets are randomly chosen within a perpendicular cross-section through $\mathcal{T}(i \cdot dt)$ corresponding to a probability distribution function (PDF) $d(\alpha, r)$ (see Eq. 1 and Fig. 2). In other words,

- many streamlets are arranged around a certain base trajectory $\mathcal{T}$ in a circular fashion. Thus, polar coordinates (r and α) were used to describe the seed states of the streamlets.
- Through PDF d the generated streamlet distribution is uniform within a certain radius (qR) and fades out linearly outside radius qR. This way of instantiating streamlets emphasizes the flow near base trajectory $\mathcal{T}$.

$$d(\alpha, r) = \begin{cases} D & \text{if } 0 < r \le qR \\ \frac{R-r}{R-qR} D & \text{if } qR < r \le R \\ 0 & \text{if } R < r \end{cases} \tag{1}$$

PDF $d(\alpha, r)$ is defined by parameters R (the maximal distance between $\mathcal{T}(i \cdot dt)$ and $\mathcal{T}_i(i \cdot dt)$) and $q \in [0, 1)$. The latter parameter is used to define PDF d as a truncated cone. This shape provides the fade-out characteristic of the streamlet placement procedure with respect to the distance from $\mathcal{T}$. To guarantee that d is a PDF $\int d(\alpha, r)\, d\alpha\, dr$ must equal 1, i.e., the volume of the truncated cone must be 1. This constraint can be expressed as specification for parameter D:

$$D = \frac{3}{(1 + q + q^2) R^2 \pi}$$

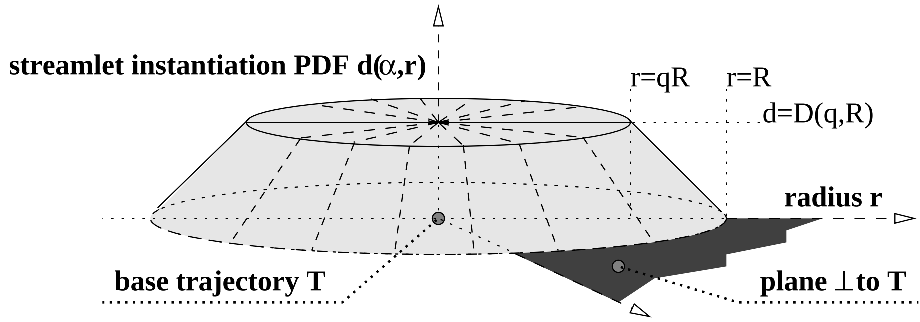


Fig. 2. Probability density function $d(\alpha, r)$ for the instantiation of streamlets based on a perpendicular cross-section through the base trajectory.

Computing a thread of streamlets for the reference model ($\dot{\mathbf{x}} = \mathbf{const}.$), a bunch of line segments (streamlets – $\{\mathcal{T}_i\}_{i\in\mathbf{N}}$) of equal length ($len \cdot |\dot{\mathbf{x}}|$) is generated. It this case of constant flow the streamlets are parallel to the base trajectory which is a straight line itself. The initial positions of streamlets $\{\mathcal{T}_i(i \cdot dt)\}_{i\in\mathbf{N}}$ are determined according to the PDF $d(\alpha, r)$. For any time t the cross-section perpendicular through $\mathcal{T}(t)$ is pierced by exactly $no = len/dt$ streamlets.

Applying this model to real (non-constant) flow data, local flow characteristics are visualized through the following variations from the constant flow reference setup:

- the **shape of the streamlets** directly visualizes the flow locally to the base trajectory. Local convergence/divergence or rotational behavior with respect to the base trajectory is intuitively depicted. Since local variations are significant in the area of (partial) degeneracies of the flow, characteristic trajectories are especially well suited to be chosen as base trajectories.
- the **streamlet length** is a direct visualization of flow velocities near the base trajectory. Due to this, the flow velocity can be depicted very well. Compared to color coding which is often used for velocity visualization the use of streamlets is more effective.

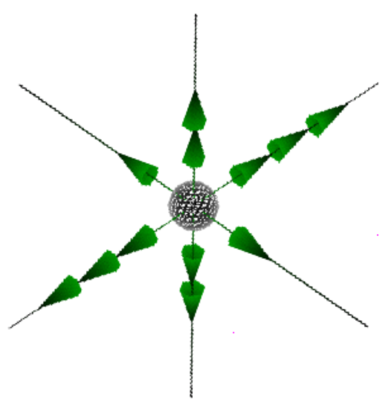
Taking a linear node repellor, i.e., a linear source, with eigenvalues 1, 10, and 100, for example, the flow characteristics in the vicinity of this fixed point can be visualized in different ways (see Fig. 3). Using threads of streamlets for a visualization of the characteristic trajectories – those which are aligned with the eigenvectors of the fixed point's Jacobian matrix – a dense and intuitive representation of the 3D flow near the fixed point is generated. Through the threads of streamlets (Fig. 3b) the flow next to the characteristic trajectories is visualized. A purely abstract notation (Fig. 3a) encodes the eigenvectors of the Jacobian matrix and the magnitudes of the associated eigenvalues. No information about the vicinity of the characteristic trajectories is provided.

3 Rendering

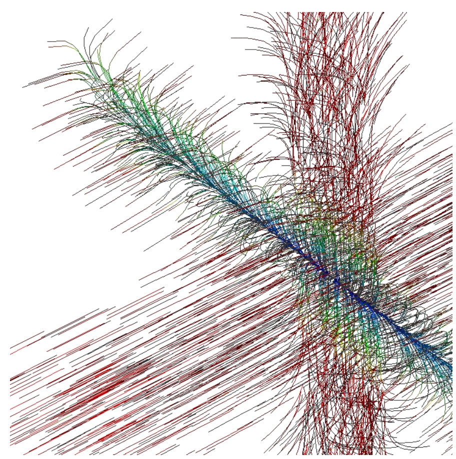
Drawing 1D objects poses several problems in the rendering stage. Shading, for example, improves the visual cues concerning the spatial arrangement of objects, but shading is usually defined on the basis of a surface (normal). Lines and curves have an infinite number of normals in each of their points. Therefore typical models such as Phong shading [15] can not be applied directly to 1D objects in 3D.

In 1989 Kajiya presented an "ad hoc" approach to deal with the problem of line shading in 3D which is based on an integration of all reflected intensities [9]. In 1996 Zöckler et al. described an efficient computation scheme for line shading in 3D which generates comparable results to the technique proposed by Kajiya [19]. A general framework for the task of shading k-dimensional manifolds in n-dimensional space was worked out by Banks in 1994 [4]. In addition to a consistent framework for the shading problem with arbitrary codimensions Banks also dealt with the problem of excess brightness-compensation which becomes an important topic if manifolds with codimension higher than 1 are shaded.

Another problem associated with line shading in 3D is (self-)shadowing. Normally, if shading 2D manifolds in 3D space, we (implicitly) deal with this aspect by assuming all surface points in (self-)shadow, where the outward normal $\mathbf{n}$ points away from the light vector $\mathbf{l}$, i.e., $\mathbf{n} \cdot \mathbf{l} < 0$. Furthermore we (implicitly) consider shadow rays before we compute surface shading. Both aspects are difficult with line shading in 3D. One approach to deal with these aspects comes from volume rendering: lines populating certain regions of 3D space can

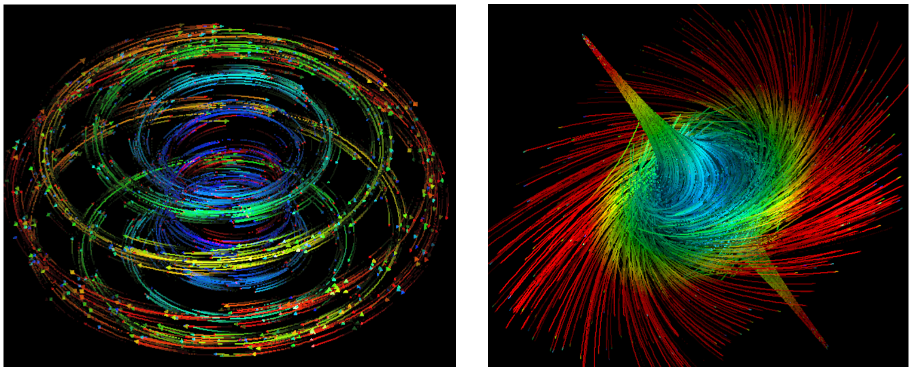


(a)



(b)

Fig. 3. Visualizing the flow near a linear node repellor in 3D: eigenvectors and eigenvalues (1, 10, and 100) (a), characteristic trajectories plus threads of streamlets (b).



(a) (b)

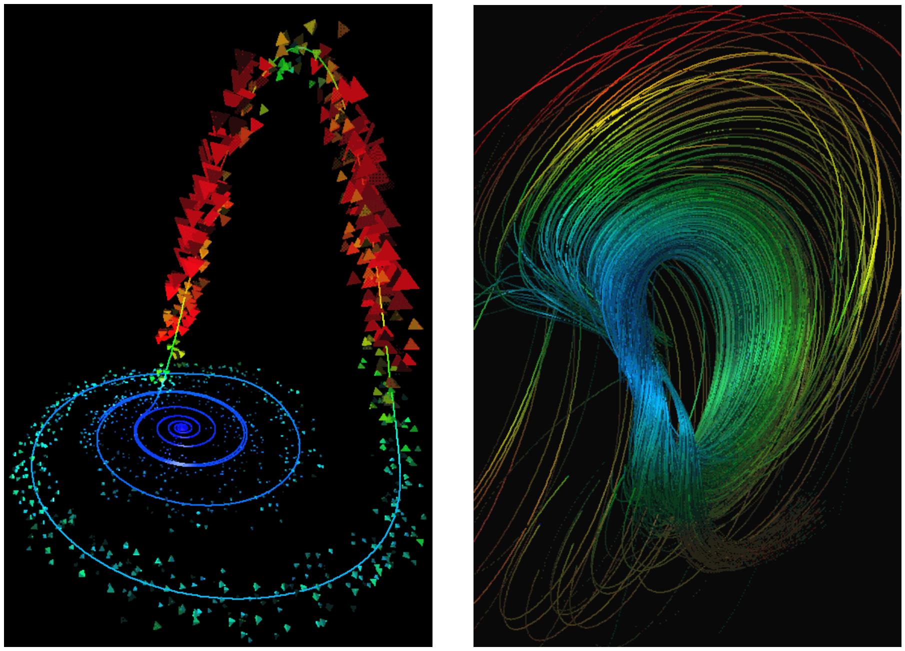
Fig. 4. A thread of streamlets visualizing the flow near a torus in 3D space (a); flow near a 3D focus visualized using two threads of streamlets (b).

be considered as volume opacity of a certain density. This assumption yields an exponential brightness attenuation for light passing through such a region. A paper by Max in 1995 compiles a comprehensive list of diverse models dealing with this effect [13].

For our implementation we chose the shading model used by Zöckler for shading the streamlets. Additionally we used depth cueing as a rough approximation of shadowing to enhance the spatial perceptability of the streamlets in 3D space. See Fig. 4a for an example. The heads of the streamlets have been pointed out by small arrow-heads to indicate the orientation of the flow. Furthermore color has been used to encode the flow velocity (blue $\leftrightarrow$ slow, red $\leftrightarrow$ fast). Line shading and depth cueing has been applied as described above.

4 Results

To test the newly proposed technique we firstly applied it to a simple cases, i.e., the fixed point of a linear dynamical system. Depending on the Jacobian matrix evaluated at this point, different results are obtained. Fig. 3b, for example, shows six threads of streamlets applied to the characteristic trajectories emanating from the fixed point. In this case the eigenvalues of the Jacobian matrix at the fixed point are 1, 10, and 100. The new visualization technique allows to easily depict the slow, medium, and fast directions of flow. Moreover, an impression is conveyed, how system states are repelled from the plane defined by the slow and medium direction (eigenvalues 1 and 10). Within that plane states are repelled from the slow direction which itself is therefore extremely instable in this setup. These flow characteristics typical for a dynamical system near a fixed point cannot be communicated by either showing an abstraction only (Fig. 3a) or a complete set of stream lines.



(a) (b)

Fig. 5. Visualizing the flow velocity near a stream line of the Roessler system (a); visualizing the dynamics of a periodic dynamical system exhibiting a twisted torus (b).

Fig. 4b is generated by using two threads of streamlets for the visualization of a 3D focus, also within a linear dynamical system. The Jacobian matrix of this system exhibits one negative eigenvalue and two conjugate complex eigenvalues with positive real parts. System states are attracted along an instable 1D manifold – a line in the case of a linear system – and repelled into a stable 2D manifold (plane) perpendicular to the instable set. Applying the threads to both instable trajectories the dynamics near this fixed point are meaningful visualized. As in Fig. 4a color was used to encode flow velocity.

There is no need for applying the new technique only to characteristic trajectories. Fig. 5 shows two examples where different results were produced with this technique. The left image shows a thread of streamlets through the Roessler system. Instead of the streamlets themselves just arrow-heads at the end of each streamlets are displayed. Using size and color according to the velocity of the flow slow and fast areas within this system are intuitively visualized. The right image depicts the dynamics of a periodic flow near a twisted torus. Color coding indicates the velocity along the streamlets. As in Fig. 5a and 5b no characteristic trajectories were used, the evolution of the streamlets is more or less aligned with the base trajectory. Regions of local convergence/divergence are implicitly shown as areas with more/less streamlets.

5 Implementation

The technique presented in this paper was implemented within DynSys3D, a visualization system concerned with analytically specified dynamical systems in 3D space [10]. According to the modular concept of this system the new visualization technique is independent of the dynamical system and the numerical integrator specification. An AVS module is generated by linking the implementation of a specific dynamical system – basically two evaluation functions for calculating the flow vector and the Jacobian matrix at a specific system state – and a specific numerical integrator, for example, a Runge-Kutta integration scheme, to the thread of streamlets implementation. The module generates one thread of streamlets for a specific dynamical system by using a specific numerical integrator.

Parameters for the module are the starting location of the base trajectory ($\mathcal{T}(0)$) and its length (either temporal or spatial), the number of streamlets per cross-section (no), their length (len), the maximum distance of their seed-points (R) together with the fade-out parameter (q). The performance of this technique is between interactive and moderate (up to one or two minutes), depending on how many steamlets are computed. However, if parameter no temporarily is set to some small number, the visualization can be adjusted interactively.

6 Conclusions

We present a new technique for the visualization of dynamical systems, namely the use of a thread of streamlets for characteristic trajectories. This is useful, since a trade-off is made between only displaying structural information such as, e.g., fixed points and separatrices, and directly visualizing the system dynamics by the use of stream lines or stream surfaces. Since an abstract denotation of the dynamics caused by a dynamical system are very hard to understand for most users, enhancing this information by locally adding cues of direct visualization helps to communicate the crucial aspects of the system behavior.

Contrary to surface based stream line visualization techniques like the stream tube of sweep base trajectory representations threads of streamlets visualize the flow continuously in the vicinity of a stream line. Furthermore, using a thread of streamlets instead of entirely populating 3D phase space with stream lines, has the advantage of reducing occlusion. Although quite a number of papers deal with densely visualizing flow in 3D space, it seems to be necessary to place visual cues selectively to reduce occlusion problems. For high-quality versions of the images presented in this paper please visit the web page at URL `http://www.cg.tuwien.ac.at/research/vis/dynsys/KnitDS97/`.

Acknowledgements. The authors thank Markus Götzinger and Helmut Doleisch for their help in preparing this paper.

References

1. R. H. Abraham and C. D. Shaw. *Dynamics – The Geometry of Behavior*. Addison-Wesley, 2nd edition, 1992.
2. H. Aref, R. D. Charles, and T. T. Elvins. Scientific visualization of fluid flow. In C. A. Pickover and S. K. Tewksbury, editors, *Frontiers of Scientific Visualization*, pages 7–43. Wiley Interscience, 1993.
3. D. K. Arrowsmith and C. A. Place. *An Introduction to Dynamical Systems*. Cambridge University Press, 1990.
4. D. C. Banks. Illumination in diverse codimensions. *Computer Graphics*, 28(Annual Conference Series):327–334, 1994.
5. B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. *Computer Graphics*, 27(Annual Conference Series):263–270, 1993.
6. W. C. de Leeuw and J. J. van Wijk. A probe for local flow field visualization. In *Proceedings of IEEE Visualization '93*, pages 117–123, October 1994.
7. T. Frühauf. Raycasting vector fields. In *Proceedings of IEEE Visualization '96*, pages 115–120, 1996.
8. E. Gröller, R. T. Rau, and W. Straßer. Modeling and visualization of knitwear. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):302–310, December 1995.
9. J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. *Computer Graphics*, 23(Annual Conference Series):271–280, July 1989.
10. H. Löffelmann and E. Gröller. DynSys3D: A workbench for developing advanced visualization techniques in the field of three-dimensional dynamical systems. In *Proceedings of The Fifth International Conference in Central Europe on Computer Graphics and Visualization '97*, pages 301–310, Plzen, Czech Republic, February 1997.
11. H. Löffelmann, E. Gröller, R. Wegenkittl, and W. Purgathofer. Classifying the visualization of analytically specified dynamical systems. *Machine* GRAPHICS *&* VISION, 5(4):533–550, 1996.
12. H. Löffelmann, Z. Szalavári, and E. Gröller. Local analysis of dynamical systems – concepts and interpretation. In *Proceedings of The Fourth International Conference in Central Europe on Computer Graphics and Visualization '96*, pages 170–180, Plzen, Czech Republic, February 1996.
13. N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, June 1995.
14. G. M. Nielson and B. Shriver. *Visualization in Scientific Computing*. IEEE Computer Society Press, 1990.
15. B.-T. Phong. Illumination for computer generated pictures. *CACM June 1975*, 18(6):311–317, 1975.
16. F. H. Post and T. van Walsum. Fluid flow visualization. In H. Hagen, H. Müller, and G. M. Nielson, editors, *Focus on Scientific Visualization*, pages 1–40. Springer, 1993.
17. W. J. Schröder, C. R. Volpe, and W. E. Lorensen. The stream polygon: A technique for 3D vector field visualization. In *Proceedings of IEEE Visualization '91*, pages 126–132, October 1991.
18. J. J. van Wijk. Spot noise – texture synthesis for data visualization. *Computer Graphics*, 25(Annual Conference Series):309–318, July 1991.
19. M. Zöckler, D. Stalling, and H.-C. Hege. Interactive visualization of 3D-vector fields using illuminated streamlines. In *Proceedings of IEEE Visualization '96*, pages 107–113, October 1996.

Appendix: Fixed Points and Characteristic Trajectories

Assuming $\dot{\mathbf{x}} = \mathbf{f_p}(\mathbf{x})$ to be continuous and steady dynamical system in 3D space, the fixed points $\mathbf{o}_i$ of $\mathbf{f_p}$ are given by

$$\dot{\mathbf{o}}_i = \mathbf{f_p}(\mathbf{o}_i) = 0$$

Using the Taylor expansion of $\mathbf{f_p}$ in the vicinity of a fixed point $\mathbf{o}_i$ together with local linearization a linear ODE in terms of $\mathit{\Delta} = \mathbf{x} - \mathbf{o}_i$ can be derived:

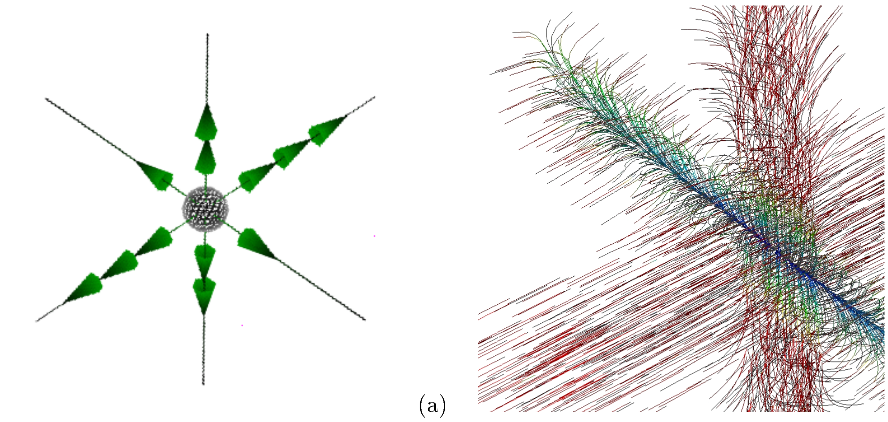
$$\mathbf{f_p}(\mathbf{o}_i + \mathit{\Delta}) = \sum_{k=0}^{\infty} \frac{1}{k!} \, (\mathit{\Delta} \cdot \nabla)^k * \mathbf{f_p}\big|_{\mathbf{o}_i} \approx \underbrace{\mathbf{f_p}(\mathbf{o}_i)}_{=0} + \nabla \mathbf{f_p}\big|_{\mathbf{o}_i} \cdot \mathit{\Delta}$$

$$\underbrace{\dot{\mathbf{o}}_i}_{=0} + \dot{\mathit{\Delta}} = \frac{d(\mathbf{o}_i + \mathit{\Delta})}{dt} = \mathbf{f_p}(\mathbf{o}_i + \mathit{\Delta})$$

$$\Longrightarrow \qquad \dot{\mathit{\Delta}} = \nabla \mathbf{f_p}\big|_{\mathbf{o}_i} \cdot \mathit{\Delta}$$

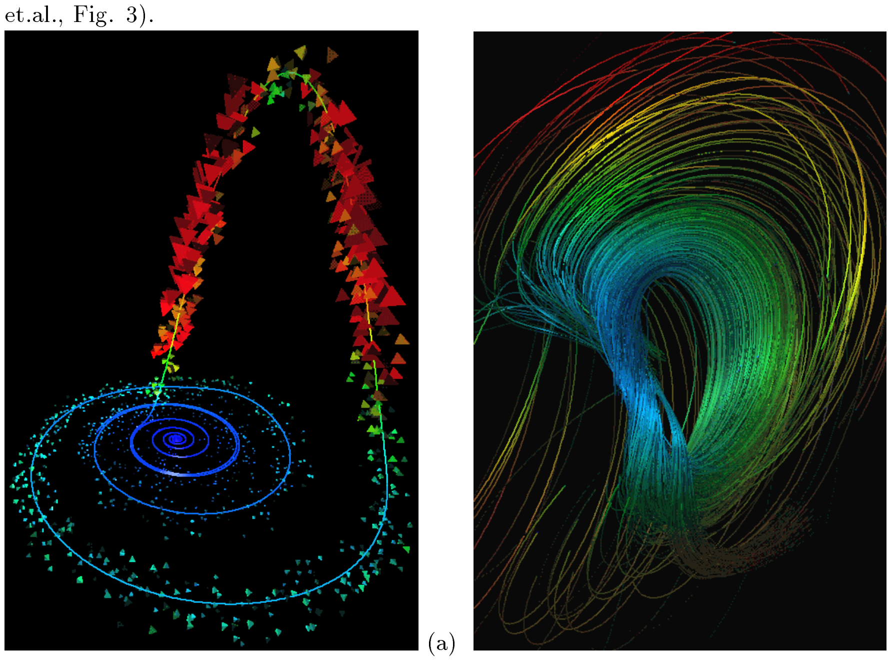
This linear dynamical system can be investigated by analyzing $\mathbf{f_p}$'s Jacobian matrix $\nabla \mathbf{f_p}\big|_{\mathbf{o}_i}$ at the fixed point $\mathbf{o}_i$. One possibility is to determine the eigenvalues and eigenvectors of the Jacobian. They completely describe the dynamics of a linear dynamical system [1].

Transferring the results from local linearization to the original system, we facilitate the fact that (in the hyperbolic case) manifolds spanned by the eigenvectors of $\nabla \mathbf{f_p}\big|_{\mathbf{o}_i}$ are coplanar with $\mathbf{f_p}$'s characteristic manifolds through $\mathbf{o}_i$. Characteristic stream lines, for example, are trajectories which are attracted to a saddle fixed point $\mathbf{o}_i$ while all the other stream lines near the characteristic trajectory (finally) are repelled from $\mathbf{o}_i$.



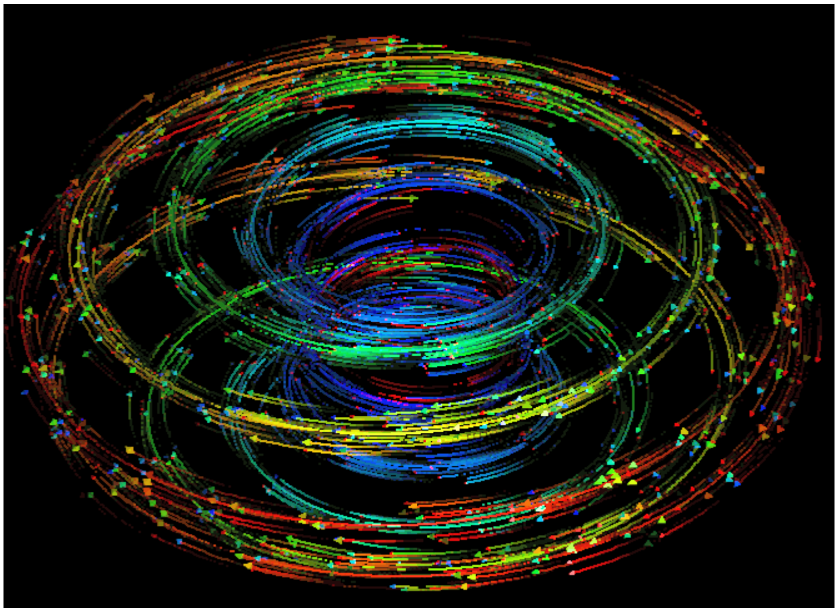
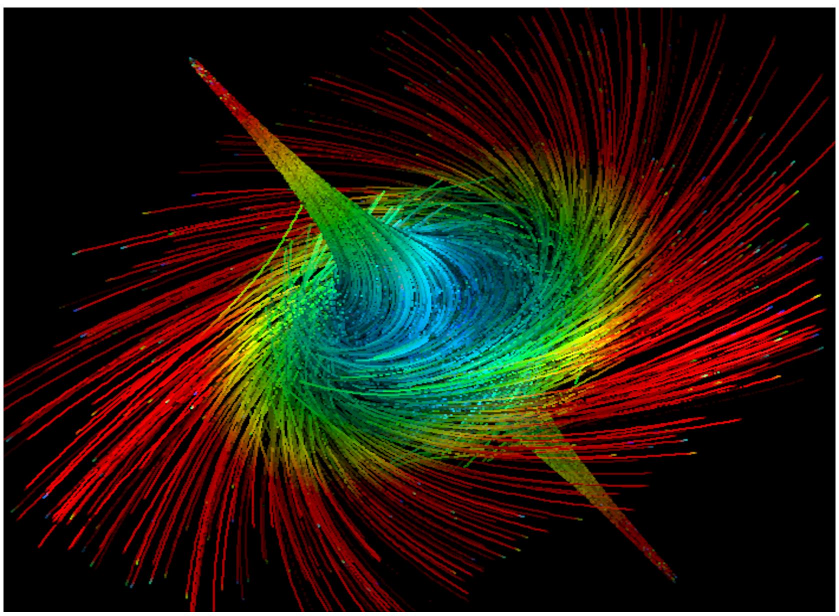
(a) (b)

Visualizing the flow near a linear node repellor in 3D: eigenvectors and eigenvalues (1, 10, and 100) (a), characteristic trajectories plus threads of streamlets (b) (Löffelmann et.al., Fig. 3).



(a) (b)

Visualizing the flow velocity near a stream line of the Roessler system (a); visualizing the dynamics of a periodic dynamical system exhibiting a twisted torus (b) (Löffelmann et.al., Fig. 4).


(a)

(b)
A thread of streamlets visualizing the flow near a torus in 3D space (a); flow near a 3D focus visualized using two threads of streamlets (b) (Löffelmann et.al., Fig. 5).