

Visualization of Dynamical Systems

Eduard Gröller and Helwig Löffelman and Rainer Wegenkittl

Vienna University of Technology, A-1040 Vienna, Austria

Abstract

The visualization of analytically defined dynamical systems is important for a thorough understanding of the underlying system behavior. An introduction to analytically defined dynamical systems is given. Various visualization techniques for dynamical systems are discussed. Several current research directions concerning the visualization of dynamical systems are treated in more detail. These are: texture based techniques, visualization of high-dimensional dynamical systems, advanced streamsurface representations, local analysis - Poincaré sections, visualizing econometric models.

Keywords: dynamical system, visualization, texture, multidimensional data, parallel coordinates, streamsurface, econometric models, Poincaré section

1 Analytically Defined Dynamical Systems

A dynamical system is a system whose temporal evolution from some initial state is dictated by a set of rules. Dynamical systems are found in many areas of research and application. Examples are fluid flow analysis, economic processes (e.g., stock market models), physics, medicine, and population growth models [2]. Dynamical systems are either given as an analytical specification or as sampled data. In the following we will concentrate on analytically defined dynamical systems.

Dynamical systems are either *continuous* or *discrete*. Continuous systems (also called flows or vector fields) are given by a set of differential equations $\dot{x} = \nu(x)$. Vector $\nu(x)$ describes the direction, orientation and velocity of the flow at position x . Discrete dynamical systems (often called maps) are specified by a set of difference equations $x_{n+1} = \nu(x_n)$.

The behavior of dynamical systems can be investigated in *phase space*, where each state variable (i.e., each coordinate component of x) corresponds to a coordinate axis. A point in phase space completely describes the state of the

system at one point in time. The temporal evolution from an initial state is called *trajectory* in case of a continuous system and *orbit* in case of a discrete system. Starting from an initial state x_0 the solution (i.e., trajectory) of a continuous dynamical system is given as curve $x(t)$ in phase space (see Equation (1)).

$$x(t) = x_0 + \int_0^t \nu(x(u)) du \quad (1)$$

Equation (1) is an integral equation which can be solved analytically only in very simple cases. Typically numerical integration is used to determine the trajectory of such a dynamical system [13].

Certain topological structures within phase space are of special interest. Locations \bar{x} with $\dot{\bar{x}} = \nu(\bar{x}) = 0$ are called fixed points, critical points or equilibrium points of the dynamical system. When all trajectories close to an equilibrium point \bar{x} converge to that point, \bar{x} is called an attractor. If all trajectories close to \bar{x} diverge \bar{x} is called a repeller. A trajectory $x(t)$ with $x(t) = x(t + T) \forall t$ is called a periodic trajectory, a cycle or an oscillation. Cycles again can be attracting or repelling. Attracting points or cycles are called *limit sets* of a dynamical system. In addition to limit points and limit cycles other limit sets occur in systems with dimension greater than two. For example in a three-dimensional system a torus can occur as a limit set. Limit sets of high-dimensional systems may have complex and sometimes even chaotic behavior.

A region within phase space where all trajectories converge to a limit set A is called the *inset* of A. When reversing the flow orientation in a dynamical system (e.g., integration is done backwards in time) the property of attraction and repulsion changes. The inset of a limit set A under reverse integration is called the *outset* of A. The insets of different limit sets are separated by *separatrices*. Separatrices segregate regions of phase space with vastly different dynamic behavior. The topology of a dynamical system can be described by the position and behavior of its limit sets. Analytically the behavior close to a limit set can often be determined by linearizing the dynamical system and analyzing the resulting Jacobian matrix.

The visualization of dynamical systems provides insight into the often intricate behavior of such systems. Many techniques result from the field of experimental fluid flow research (e.g., particle injection or dye advection). There are techniques to visualize entire classes of dynamical systems (e.g., bifurcation diagrams [1]). Systems within a class have different system parameters. Taking a specific set of parameters allows to investigate a single system. The entire phase space may be visualized (e.g., hedge-hog method [11], spot noise [14], LIC [3], topological representation [6]). The visual analysis may also

be directed only to a (small) subset of phase space. Techniques are either direct visualizations of the flow (e.g., LIC) or they show derived quantities, like topological structures. Topological representations show for example limit sets, separatrices, saddle-connections, and homo-clinic orbits. These objects are often quite difficult to calculate but suffice to illustrate the qualitative behavior of the flow.

The hedge-hog method displays the tangential directions of the flow at selected positions (e.g., regular grid) of phase space. The length of the vectors encodes flow velocity. Such plots give information about the vector field but do not show important structures, e.g., vortices of the underlying dynamical system. Although quite feasible for 2D dynamical systems the images for higher dimensions tend to become crowded. Occlusion is a problem in such images which are therefore difficult to analyse.

Often scientists want to investigate a dynamical system at some specific point in phase space. They are interested in local aspects as, for example, velocity, acceleration, and divergence. A vortex or a point near an obstacle may be such a point of special interest. Local properties are often determined by analyzing the Jacobian matrix at some point of the flow. Visualization techniques were developed that help to investigate these local flow properties. Glyphs are a prominent example for this type of visualization. A glyph is a geometric object whose properties (e.g., length, shape, color) encodes underlying flow properties like acceleration, shear, curvature, torsion, convergance and divergence. Such a glyph can be an enhanced three-dimensional arrow which is positioned interactively or automatically, e.g., along consecutive points of a trajectory [4].

Streamlines, streaklines and pathlines [11] illustrate the temporal evolution of an initial position in phase space. A streamline visualizes a trajectory of a dynamical system. It also describes the path of a single particle in a time-independent flow. A pathline describes the path of a single particle in a time-dependent flow. A streakline visualizes the path of a sequence of particles which are introduced into a time-dependent flow at a fixed spatial position but regularly distributed over time. Timelines on the other hand result by introducing particles at a fixed moment in time but regularly distributed in phase space. Starting with more general objects like lines and circles produces streamribbons, streamsurfaces, and streamtube.

A bifurcation diagram is the most common method for visualizing an entire class of dynamical systems. Such a diagram is constructed by extending the plot of the dynamical system's long-term behavior by additional coordinate axes. These axes corresponding to various system parameters. Variations along the axes represent modifications to the model and thus several systems can be illustrated within one image. At certain parameter values the behavior

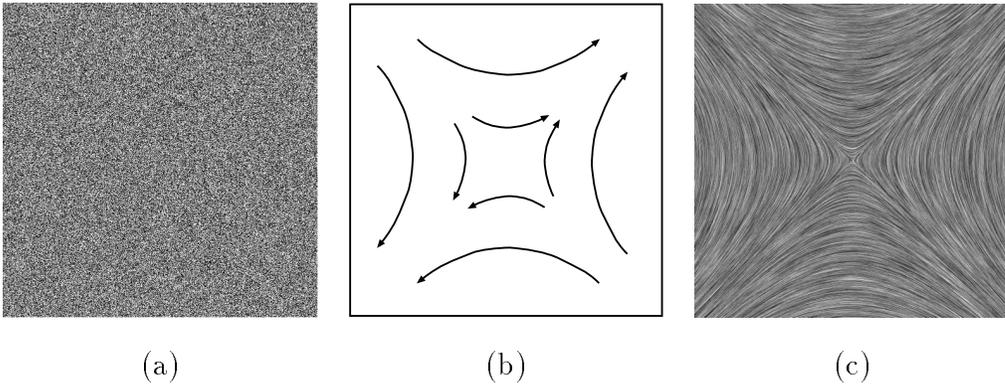


Fig. 1. white noise texture (a) – 2D vector field (b) – LIC: white noise texture filtered according to the vector field (c)

may change qualitatively (bifurcation scenario) therefore these plots are called bifurcation diagrams [1].

In the following several recent approaches to visualize dynamical systems are described in more detail. These techniques include: texture based flow visualization, visualizing high-dimensional dynamical systems, advanced streamsurface representation, local analysis - visualizing Poincaré sections, and visualizing econometric models.

2 Texture Based Techniques

Texture based techniques for the visualization of flow fields have been investigated in detail in recent years. Typically a high frequency texture (Figure 1(a)) is filtered according to an underlying vector field (1(b)) to produce a global overview of the entire flow (1(c)).

Line Integral Convolution (LIC) as illustrated in Figure 1 smoothes a white noise input texture along (curved) streamline segments [3]. LIC uses one-dimensional filter kernels which are determined by integrating the underlying vector field. The intensity $I(x_0)$ at an arbitrary position x_0 of the output image is calculated by

$$I(x_0) = \int_{s_0-s_l}^{s_0+s_l} k(s-s_0)T(\sigma(s)) ds, \quad (2)$$

where T is the input texture, $\sigma(s)$ is the parameterized streamline through x_0 ($x_0 = \sigma(s_0)$) and $k()$ describes the convolution kernel. s_l specifies the length of the streamline segment in the filter operation. The texture values along the streamline segment $\sigma(s)$, ($s_0 - s_l \leq s \leq s_0 + s_l$), are weighted with

the corresponding kernel values $k(s - s_0)$. They are accumulated to give the intensity $I(x_0)$ at position x_0 . Various kernel functions $k()$ can be used in the filter operation. For single images a constant filter kernel gives a good impression of the flow direction. Taking periodic low-pass filter kernels and phase shifting these kernels in successive images allows to animate the flow field. The animation shows flowing ripples which also encode the orientation of the flow.

LIC images encode flow direction and velocity magnitude, but they do not show the orientation of the flow in still images. Flow orientation can be illustrated through animation. But there are cases where only still images are available or necessary, e.g., reproduction of vector fields in books or journals. Furthermore LIC images are characterized by high spatial frequencies normal to the flow. This gives a good impression of the overall vector field, but is susceptible to aliasing artefacts in case an image has to be manipulated, e.g., it is resized or printed. Oriented Line Integral convolution (OLIC) [15] was designed to show the orientation of a flow even in still images and it is not as much prone to aliasing effects as LIC. There are two major differences between LIC and OLIC. LIC images typically use dense noise textures whereas OLIC utilizes only sparse textures. A sparse texture can be thought of as a set of ink droplets which are thinly distributed on a sheet of paper. The vector field smears these ink droplets but the ink droplets are so far apart from each other that blurred traces of droplets usually do not overlap. The second difference between LIC and OLIC is that OLIC uses asymmetric convolution kernels. A ramp-like kernel as in Figure 2 produces traces of droplets with intensity varying along the streamline. As a sparse texture is taken traces do not overlap very much and the orientation of the flow is visible in still images.

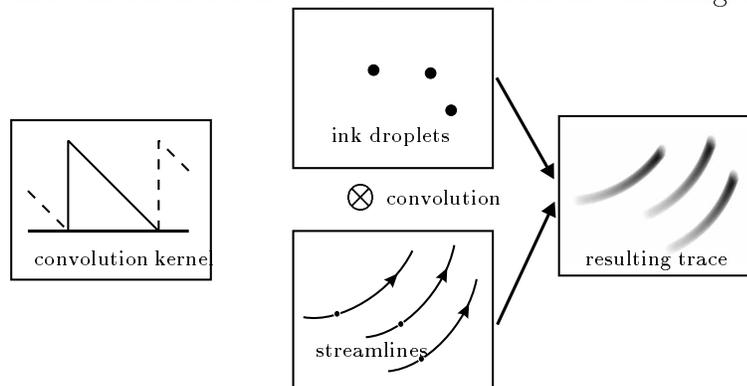


Fig. 2. OLIC: LIC with sparse texture and ramp-like kernel-function

In Figure 3 the difference between LIC and OLIC is clearly visible. Figure 3(a) shows the LIC image of a circular flow. In this image it is not recognizable if the flow is in clockwise or counterclockwise orientation. Figure 3(b) shows the OLIC image of a circular clockwise flow and Figure 3(c) shows the OLIC image of a circular counterclockwise flow. The additional information in the

OLIC image is gained at the expense of spatial resolution.

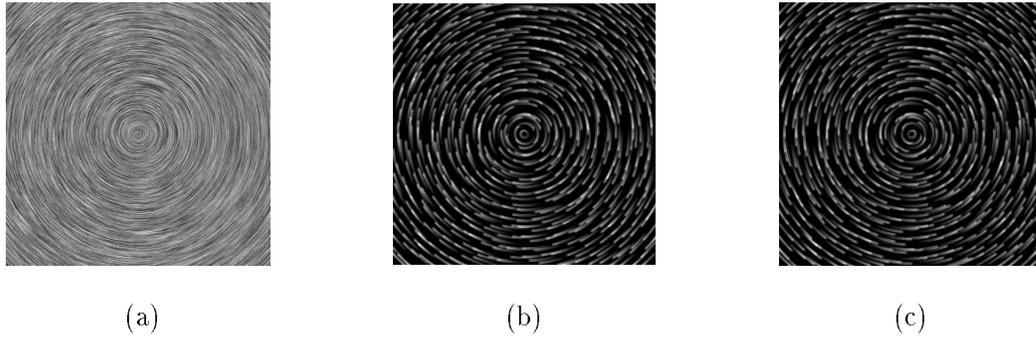


Fig. 3. LIC image of circular flow (a), OLIC image with clockwise flow (b), OLIC image with counterclockwise flow (c)

The initial positions of the droplets in the sparse texture must be selected carefully to avoid the formation of undesirable macroscopic patterns in the OLIC image. For efficiency reasons individual droplet traces can be approximated by a set of small disks with varying intensity [15]. This approach avoids the costly convolution operation and allows an easy calculation of animation sequences.

3 Visualizing High-Dimensional Dynamical Systems

In recent years scientific visualization has been driven by the need to visualize high-dimensional data sets within high-dimensional spaces. Most of these visualization methods are designed to visualize point sets. Typically these methods show statistical features like correlations, clustering or outliers.

Several visualization methods for high-dimensional data can be distinguished. Attribute mapping uses one or two-dimensional lattices to define some simple geometric primitives, e.g., contours or planes. The attributes of these geometric primitives can be used to visualize the remaining variables. The most often used attribute is the color of the geometric primitive (color coding).

Geometric coding maps high-dimensional data to distinct geometric objects, e.g., glyphs or icons. A glyph is a graphical entity whose shape or appearance is modified by mapping data values to some of its graphical attributes. An interactively positioned glyph adapts its appearance according to the underlying data. Variables can be mapped to the length, shape, angle, color and transparency of the glyph. Examples of this kind of visualization are given in [4,12].

Another method to visualize high-dimensional data sets is the reduction of dimension. This can be done by either focusing, where only part of the whole

data set is shown, or by linking, where some focused parts are linked together to represent the whole data set. Focusing techniques may involve selecting subsets, reduction of dimension by projection, or some more general manipulation of the information layout on the screen (e.g., nonlinear zooming). Examples for subset selection techniques are panning, zooming, and slicing.

Parallel coordinates [7] represent dimensions on parallel axes. Each axis represents one coordinate component. All axes are arranged orthogonal to a horizontal line uniformly spaced on the display. An n -dimensional point of the data set is displayed as a polyline that intersects the parallel coordinate axes at the corresponding coordinate values of the data point. By interactively brushing through the data set statistical characteristics like outliers and clusters can be recognized easily.

Displaying an n -dimensional trajectory is an important task to allow a direct global visualization of the behavior of a dynamical system. Extruded parallel coordinates [17] are based on parallel coordinates. They represent an n -dimensional trajectory as surface in 3D. With parallel coordinates a trajectory is sampled at discrete points in time $\{x(t_0), x(t_1), x(t_2), \dots\}$ and its coordinates are inserted as polylines in a parallel coordinate system (see left side of Figure 4). Instead of using the same coordinate system for each sample we now move the parallel coordinate system along the third spatial axis. The polylines of the samples can be viewed as cross sections of a moving plane with a complex surface which defines the trajectory. The right side of Figure 4 shows this surface and the moving parallel coordinate system at the end of the surface.

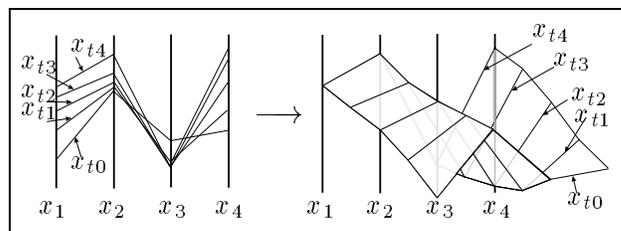


Fig. 4. A discrete sampled trajectory in parallel coordinates (left) and a three-dimensional extruded surface defining the same trajectory (right)

In Figure 5 extruded parallel coordinates illustrate a trajectory of a chaotic attractor (five-dimensional system). To show the chaotic behavior of the attractor, the starting point of the third dimension is slightly jittered (point A) and three different trajectories (shown with different colors) are superimposed. The tiny differences of the starting coordinates produce considerable differences after a few integration steps (points B).

Linking with Wings [17] is a new method of linking data (Figure 6). Two arbitrary dimensions of the high-dimensional system are selected and displayed as a two-dimensional trajectory within a base plane. This means that the high-

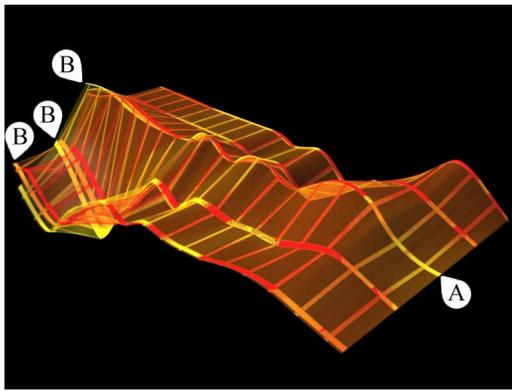


Fig. 5. Extruded parallel coordinates illustrate three trajectories of a five-dimensional system

dimensional trajectory is projected into a two-dimensional subspace. The third dimension (along the z -axis) can now be used to display additional variables over the base trajectory. If the resulting three-dimensional trajectory is connected with the base trajectory this connection can be thought of as a wing on the base trajectory. This wing can be tilted at each point within a plane normal to the base trajectory. When different tilting angles are used several additional dimensions can be linked to the base trajectory on separate wings.

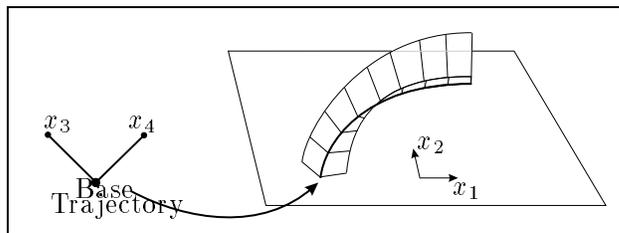


Fig. 6. Two-dimensional base trajectory with two wings (for the third and fourth dimension) linked to it

Theoretically any number of wings can be added to display high-dimensional trajectories. As the number of wings increases occlusion might become a severe problem. To avoid this the wings can be rendered transparently with opaque tubes at the top. The wings can also be textured with a grid texture allowing an exact measurement of the wing dimensions. To overcome the problem of occlusion texture can be used to modulate the transparency of the wings.

In Figure 7(a) wings are applied to a base trajectory of the four-dimensional Wonderland model [16], which describes the interactions of population, economy, environment, and pollution. Population and economy define the base trajectory. Environment and pollution are shown as wings. In this case the green trajectory (pollution) declines in a monotonic fashion, whereas the blue trajectory (describing the environment) collapses at point A and regenerates at point B.

Figure 7(b) shows a hedge-hog visualization of a four-dimensional data set.

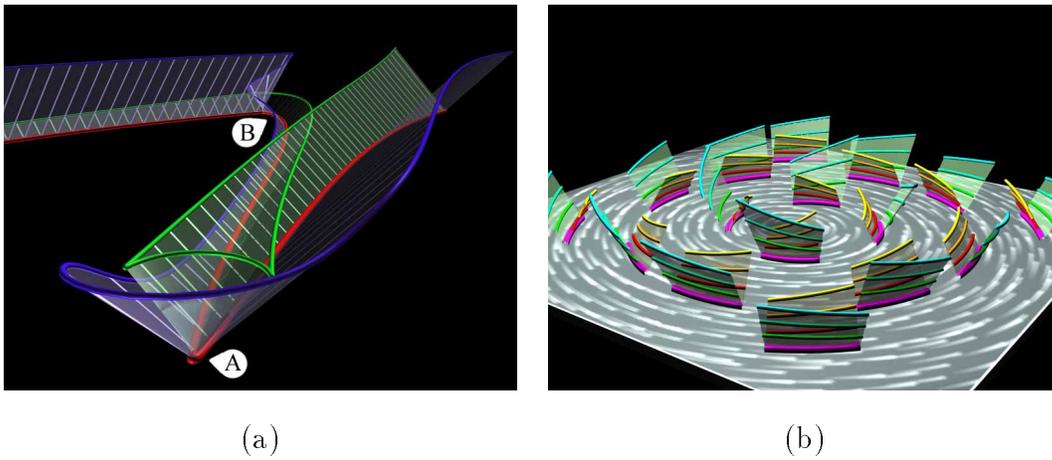


Fig. 7. Linking with wings of an econometric model (a) – and hedge-hog visualization of a four-dimensional data set (b)

On a regular four-dimensional grid flow directions are calculated. The visualization shows a cyclic behavior in the first and second dimension, whereas the third dimension is attracted and the fourth is repelled by the origin. The cyclic behavior in the ground plane is additionally visualized with OLIC.

Three-dimensional parallel coordinates [17] are again based on the parallel coordinate method. The basic idea of parallel coordinates is to depict each coordinate component on a one-dimensional space. All these one-dimensional spaces are put together within a two-dimensional space and linked with one-dimensional polylines. All information is packed in the two-dimensional space. Since the visualization of three-dimensional structures poses no problem we can increase each dimension of the parallel coordinate method. The basic information now resides in separate two-dimensional spaces (planes) where two dimensional projections of trajectories are shown. These planes are combined within three-space and linked by surfaces which connect the separate projections of trajectories (see left part of Figure 8).

The positioning of the planes is more flexible in comparison to the parallel lines of the parallel coordinate method. The planes can be moved and rotated within three-space to avoid occlusion in different regions of the structure. The right side of Figure 8 shows two coinciding planes, where the connecting surface extends into the third dimension to give a better overview of the linking.

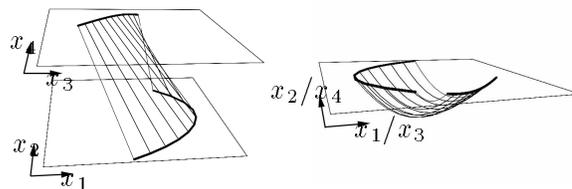


Fig. 8. Three-dimensional parallel coordinates

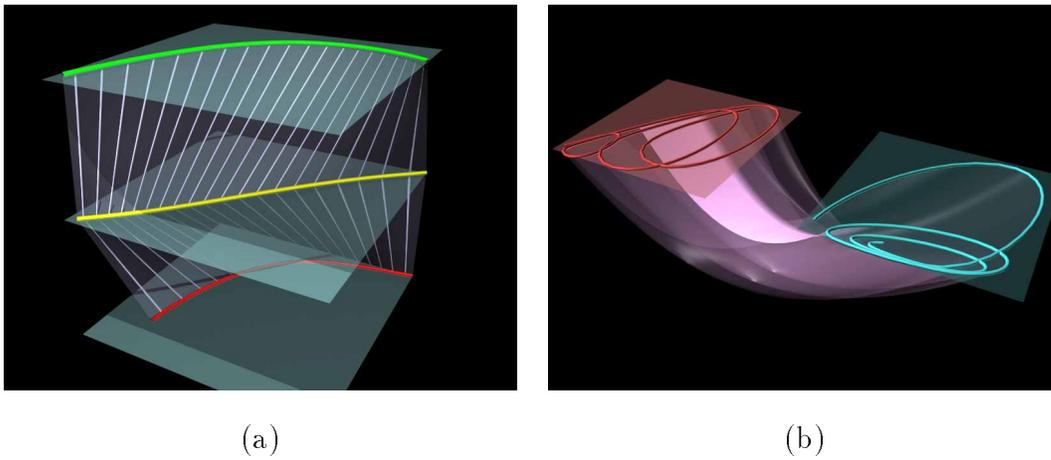


Fig. 9. A six-dimensional stacked predator-prey model with simple linkage (a) – Complex linkage with highlighted time interval for the trajectory of a chaotic attractor (b)

In Figure 9(a) a six-dimensional predator-prey system is stacked with the extended parallel coordinate method. Due to the simple shape of the separated trajectories the linking surfaces can be seen easily. Figure 9(b) on the other hand shows a trajectory of a complex dynamical system. Here the structure of the linkage can not be perceived easily. Therefore a small temporal interval has been highlighted on the linking surface. This interval can be animated or interactively moved forward and backward to reveal more clearly the structure of the linkage.

4 Advanced Streamsurface Representations

Streamsurfaces are generated by following a line of initial positions through the phase space of a dynamical system. They help to understand topological structures. With highly occluding streamsurfaces an opaque representation of the surfaces is not appropriate. It is also not a good idea to represent a streamsurface being homogeneously transparent. Portions of the image may be covered by many layers of semi-transparent parts of the streamsurface. This makes it quite difficult to recognize the spatial arrangement of the objects. There are methods which modulate the transparency locally within a surface [8]. Stream arrows [10] also belong in this area of research. Stream arrows segment a streamsurface with arrow shapes. Semi-transparent parts of the streamsurface allow the viewer to see through and perceive parts of the model that otherwise would have been occluded. The opaque part of the streamsurface on the other hand still conveys a clear impression of the shape of the streamsurface. See Figure 10 for typical examples. The arrows used in the segmentation are distorted according to the local flow and also indicate the flow direction. Long stream arrows, for example, visualize a streamsurface

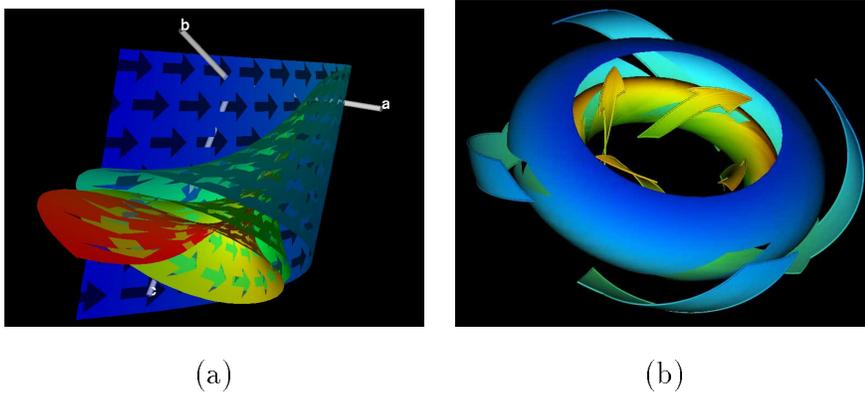


Fig. 10. Two examples of streamsurfaces with stream arrows

region of high velocity. Comparing the width of an arrow's head to the width of its tail indicates regions of convergent or divergent behavior. Stream arrows can be animated by moving them along streamlines. Only a short animation sequence has to be calculated which is then cycled.

Stream arrows are generated by one of two approaches. On the one hand they can be realized as an alpha-texture that is composed of a set of regularly arranged arrows. Either the stream arrows or the remaining surface portions can be rendered transparently. With this approach the geometric database of the streamsurface which is a set of triangles is not modified.

On the other hand stream arrows can be represented directly by geometrically separating the original streamsurface into a set of arrow-shaped patches and the remaining surface which contain holes. This approach allows to operate on both parts independently from each other. In the segmentation process the streamsurface is tiled with a base stream-arrows tile (texture) which contains a single arrow. The shape of the arrow, its size, and the tessellating scheme are given as parameters of the base tile.

Figure 10(a) illustrates the stream-arrows texture applied to a highly curled streamsurface. The streamsurface is generated by following a line of initial conditions (colinear to the c -axis) through phase space. As the streamsurface evolves over time it forms a shell-like roll. Color coding was used to distinguish states later in time from earlier ones, i.e., as time goes on the corresponding streamsurface regions are colored blue, green, yellow, and red, respectively. Due to local varying velocities and large divergence, arrows located in the red part of the streamsurface are bigger than those in the green region. Figure 10(b) displays a torus shaped streamsurface with stream arrows.

Spot noise is a stochastic texture synthesis technique to visualize scalar fields and vector fields [14]. A spot noise texture is constructed by accumulating randomly weighted and positioned spots. An anisotropic spot, e.g., cross or hash, accentuates horizontal and vertical directions in the resulting spot noise

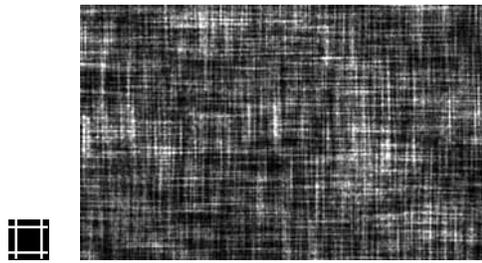


Fig. 11. spot (enlarged) and the resulting spot noise texture

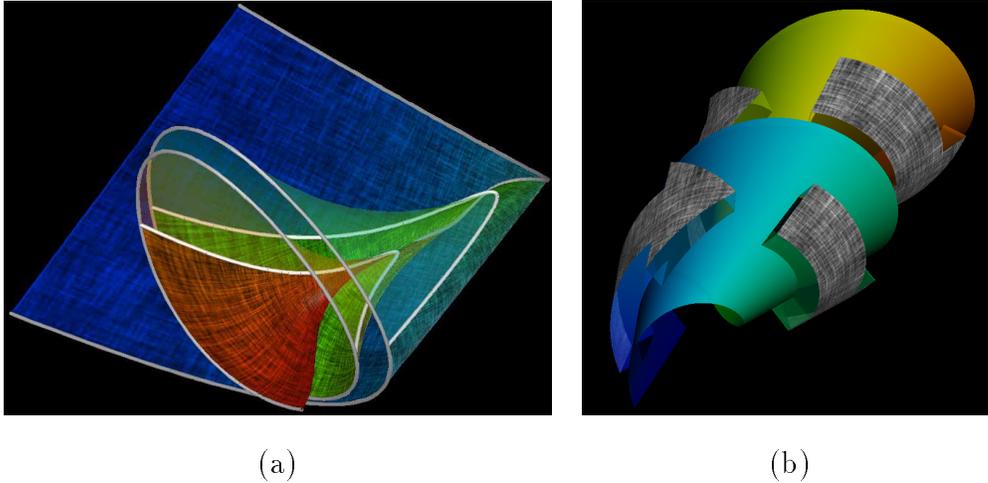


Fig. 12. Streamsurface with anisotropic spot noise texture (a) –Stream arrows shifted out of the stream surface and anisotropic spot noise (b)

texture(see Figure 11). Anisotropic spot-noise textures can be efficiently used to emphasize streamlines and timelines simultaneously within a streamsurface. The anisotropic spot noise of Figure 11 is mapped onto the streamsurface so that the horizontal direction of the texture is aligned with the streamlines and the vertical direction of the texture is aligned with the timelines. Figure 12(a) shows a textured streamsurface. The upper part of the streamsurface is rendered semi-transparently. In Figure 12(b) the separated stream arrows are slightly shifted in the direction perpendicular to the remaining streamsurface portions.

5 Local Analysis - Visualizing Poincaré Sections

Poincaré sections are an important tool for the investigation of dynamical systems that exhibit periodic or quasi-periodic behavior [9]. A 2D Poincaré section through a periodic 3D flow is a planar cross-section transverse to the flow. The periodic trajectory (also called base cycle) intersects the Poincaré section at its center. The corresponding Poincaré map P is a map which correlates consecutive intersections of a trajectory with the Poincaré section (see

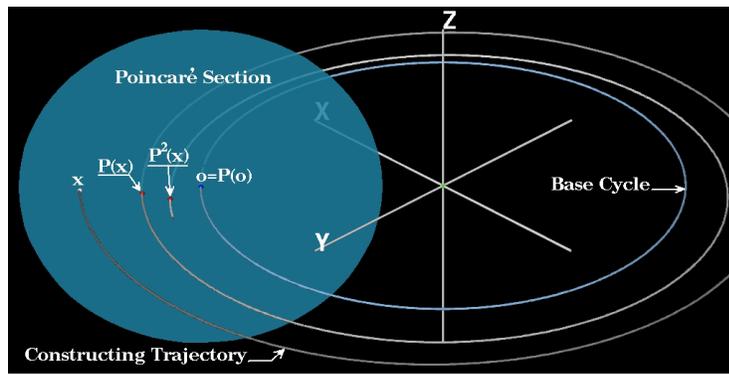


Fig. 13. An illustration of the Poincaré map definition

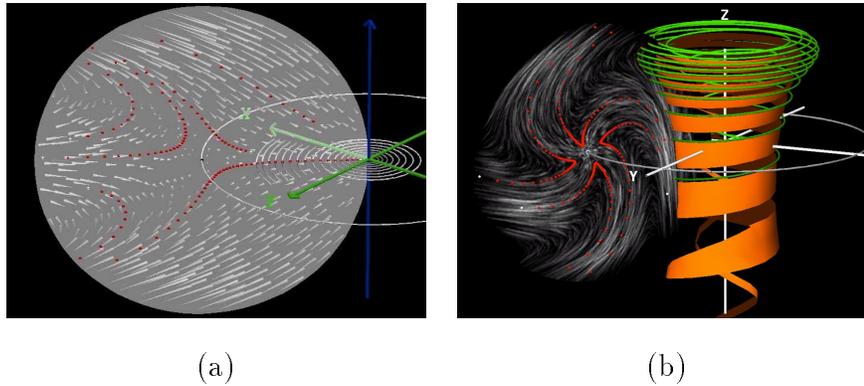


Fig. 14. Poincaré map visualized with directed strokes (a) – and with spot noise, streamlines and streamsurface added (b)

Figure 13. The Poincaré map is a discrete dynamical system of one dimension less than the underlying continuous flow. Many flow properties of the dynamical system carry over to the Poincaré map which is easier to analyze. These properties are, e.g., periodicity, quasi-periodicity, and stability behavior.

Figure 14(a) illustrates a non-linear saddle cycle. Corresponding points x and $P(x)$ are connected by directed strokes. The Poincaré section itself is rendered as semi-transparent disk. The small spheres represent sequences of successive applications of P , i.e., $\{P^j(x_i) | j \geq 0\}$ for several initial positions x_i . In Figure 14(b) spot noise shows the dynamics within the Poincaré section. The spot noise texture consists of elliptic spots. The focal points of the ellipses are positioned at x and $P(x)$ respectively. Additionally streamlines and a streamsurface are displayed. The simultaneous visualization of a Poincaré map and the underlying flow helps to better understand the flow characteristics.

The visualization of repeated applications of P , i.e., P^n , $n > 1$ is done by placing a texture onto the Poincaré section (see Figure 15(a)). Each application of P transforms the texture and after several steps an image like in Figure 15(b) emerges. Directed strokes additionally illustrate the dynamics within the Poincaré section. An efficient texture transformation is achieved

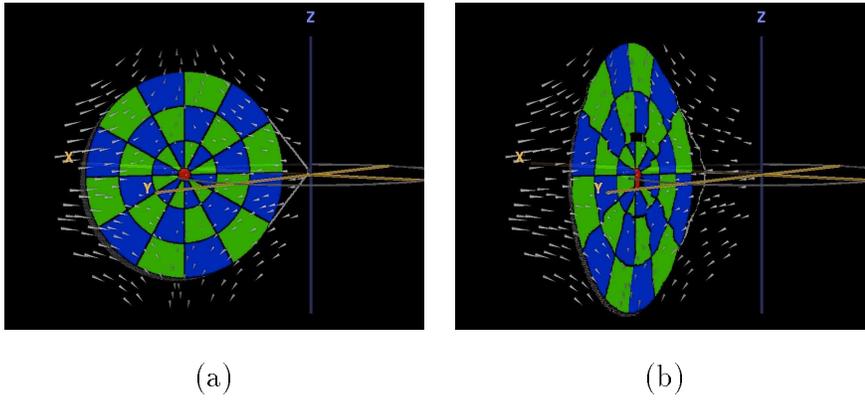


Fig. 15. A texture on the Poincaré section (a) is distorted after repeated applications of Poincaré map P (b)

with image warping techniques. Map P is evaluated only for a small set of lines in the Poincaré section. The remaining and larger part of the Poincaré section is transformed with respect to the previously modified lines. A costly evaluation of map P is not done in this case.

6 Visualizing Econometric Models

Econometricians often model economic processes as sets of differential equations. Visualization proves to be a valuable tool to analyze the often intricate behavior of such models. Two examples are discussed in the following: the Dynastic Cycle and the Wonderland model.

The Dynastic Cycle [5] is a three-dimensional dynamical system, that models the rise and fall of dynasties in ancient China. The behavior is characterized by alternating periods of anarchy and despotism. The three system variables X , Y , and Z express the number of farmers, bandits, and soldiers, respectively. The model defines their interactions similarly to well-known food-chains (prey, predator, and super-predator). The evolution induced by the Dynastic Cycle is governed by so-called slow-fast dynamics. Two of the system variables (X , Y) are *fast* variables that change rapidly in comparison to the *slow* one (Z). The knowledge about this slow-fast characteristics simplifies the analysis and must be considered during visualization.

Figure 16(a) shows a typical phase-space representation of the Dynastic Cycle. Particular states of special interest were labeled to support the interpretation of the model. A small number of soldiers indicates anarchy (states 2 and 3), whereas a large number of soldiers is representative for despotism (states 4 and 1). The temporal evolution is visualised by sweeping a circular cross-section along a streamline. Color encodes the velocity of the system. Blue

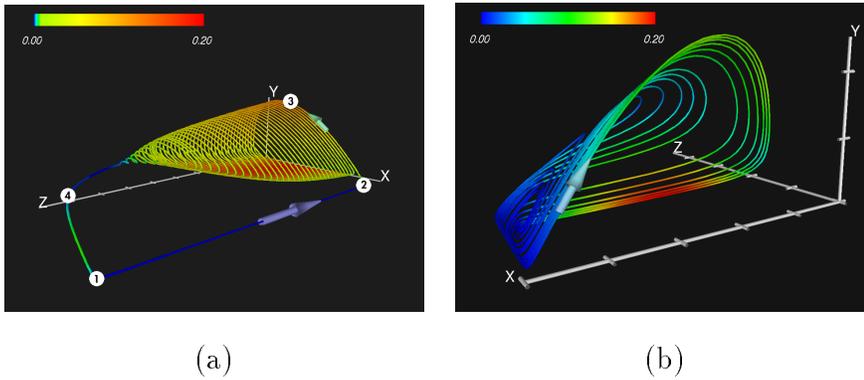


Fig. 16. A typical trajectory of the Dynastic Cycle (a) A chaotic trajectory of the Dynastic Cycle (b)

color indicates slow motion, whereas yellow or red colors represent fast transitions. Three-dimensional arrow-glyphs are positioned at certain locations of the streamline to visualize the direction of the flow. By varying system parameters the long-term behavior induced by the model changes. Figure 16(b) shows such an altered system, which exhibits chaotic behavior. Again blue color indicates regions of slow developments, whereas yellow and red denote areas of rapid changes.

The Wonderland model [16] describes the interactions between population growth, economic activities, environment, and pollution. It is a four-dimensional system with three slow system variables and one fast variable. Similarly to the Dynastic Cycle the slow-fast characteristics of the induced dynamics can be exploited to analyze the model. Mainly three scenarios, i.e. the “dream scenario” (clean environment forever), the “horror scenario” (total environmental crash), and the “escape scenario” (cycles of environmental degeneration and regeneration), can be distinguished. Certain surfaces in phase space (called critical manifolds) determine the long-term behavior and are therefore of special interest in the visualization process.

Figures 17(a) shows a typical phase-space representation of the Wonderland model. Only the three most interesting system variables are depicted. A trajectory is given as a tube with a spiraling texture mapped onto it. The distance between adjacent coils of this texture encodes local velocity. The critical manifolds (Z_0 , Z_1 , Z_c) are displayed as semi-transparent surfaces. Transparency is modulated on Z_c , the most interesting manifold, to give a better impression of the spatial situation. Color encodes attracting or repelling behavior of points on the manifolds. Manifold Z_c changes over time and determines which of the scenarios (dream, horror, escape) occurs. In Figures 17(b) particles are placed on the manifold Z_c to illustrate the flow close to the manifold. Particles are initially distributed evenly over the manifold and follow the flow dynamics close to manifold Z_c .

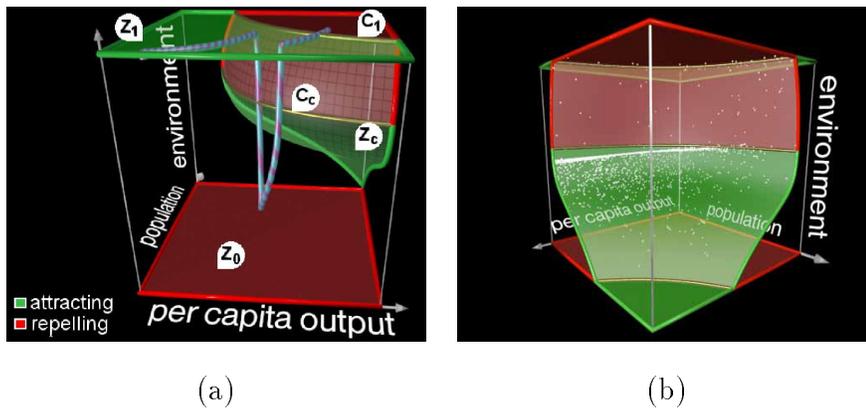


Fig. 17. Wonderland model with critical manifolds (“escape scenario”) (a) – particle system visualizes flow close to a critical manifold (b)

7 Conclusion

Graphical analysis is a valuable tool in the investigation of analytically defined dynamical systems. An overview of various visualization techniques shows that different investigation goals lead to greatly varying graphical representations. Several current research directions, i.e., texture based techniques, visualizing high-dimensional dynamical systems, advanced streamsurface representations, visualizing Poincaré sections and visualizing econometric models, emphasize the diversity of graphical tools available for the analysis of analytically defined dynamical systems. Further information on the described techniques is available at <http://www.cg.tuwien.ac.at/research/vis/dynsys/>. Color versions of the images are available at <http://www.elsevier.com/locate/future>.

8 Acknowledgements

The authors would like to thank Helmut Doleisch, Gustav Feichtinger, Georg Fischel, Andreas König, Thomas Kucera, Lukas Mroz, Alexia Prskawetz, Werner Purgathofer.

References

- [1] R. H. Abraham and C. D. Shaw. *Dynamics - the Geometry of Behavior*. Addison-Wesley, 1992.
- [2] D. K. Arrowsmith and C. A. Place. *An Introduction to Dynamical Systems*. Cambridge University Press, 1990.

- [3] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings SIGGRAPH '93*, pages 263–272, 1993.
- [4] W. C. de Leeuw and J. J. van Wijk. A probe for local flow visualization. In *IEEE Visualization '93 Proceedings*, pages 39–45. IEEE Computer Society, October 1993.
- [5] G. Feichtinger, A. Prskawetz, E. Gröller, and G. Fischel. Despotism and anarchy in ancient china: Visualizing the dynastic cycle. *Jahrbuch der Wirtschaftswissenschaften*, Band 47(Heft 1):1–13, 1996.
- [6] J. Helman and L. Hesselink. Visualization of vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.
- [7] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry. In *IEEE Visualization '90 Proceedings*, pages 361–378. IEEE Computer Society, October 1990.
- [8] V. Interrante, H. Fuchs, and St. Pizer. Illustrating transparent surfaces with curvature-directed strokes. In *IEEE Visualization '96 Proceedings*, pages 211–218. IEEE Computer Society, October 1996.
- [9] H. Löffelmann, Th. Kucera, and E. Gröller. Visualizing Poincaré maps together with the underlying flow. In *International Workshop on Visualization and Mathematics'97 Proceedings*. Springer, September 1997.
- [10] H. Löffelmann, L. Mroz, E. Gröller, and W. Purgathofer. Stream arrows: Enhancing the use of streamsurfaces for the visualization of dynamical systems. *The Visual Computer*, 13:359–369, 1997.
- [11] 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.
- [12] F. J. Post, Th. van Walsum, F. H. Post, and D. Silver. Iconic techniques for feature visualization. In *IEEE Visualization '95 Proceedings*, pages 288–295. IEEE Computer Society, October 1995.
- [13] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [14] J. J. van Wijk. Spot noise: Texture synthesis for data visualization. In *Proceedings SIGGRAPH '91*, pages 309–318, July 1991.
- [15] R. Wegenkittl and E. Gröller. Fast oriented line integral convolution for vector field visualization via the internet. In *IEEE Visualization '97 Proceedings*, pages 309–316. IEEE Computer Society, October 1997.
- [16] R. Wegenkittl, E. Gröller, and W. Purgathofer. Visualizing the dynamical behavior of wonderland. *IEEE Computer Graphics & Applications*, 17(6):71–79, December 1997.

- [17] R. Wegenkittl, H. Löffelmann, and E. Gröller. Visualizing the behavior of higher dimensional dynamical systems. In *IEEE Visualization '97 Proceedings*, pages 119–125. IEEE Computer Society, October 1997.