Generating Quasi-symbolic Representation of Three-Dimensional Flow

Toyoaki Nishida
Graduate School of Information Science
Advanced Institute of Science and Technology, Nara
8916-5, Takayama-cho, Ikoma-shi, Nara 630-01, Japan
nishida@is.aist-nara.ac.jp

Abstract

Understanding flow in the three-dimensional phase space is challenging both to human experts and to current computer science technology. To break through the barrier, we are building a program called PSX3 that can autonomously explore the flow in a three-dimensional phase space, by integrating AI and numerical techniques.

In this paper, I point out that quasi-symbolic representation called *flow mappings* is effective as a means of capturing qualitative aspects of three-dimensional flow and present a method of generating flow mappings for a system of ordinary differential equations with three unknown functions. The method is based on a finding that geometric cues for generating a set of flow patterns can be classified into five categories. I demonstrate how knowledge about interaction of geometric cues is utilized for intelligently controlling numerical computation.

Introduction

Behavior of ordinary differential equations (ODEs) can be qualitatively understood by analyzing the topological and geometric features of the *flow* or the *phase portrait*, which is the set of all solution curves or *orbits* in the phase space. Usually, essential part of qualitative analysis is carried out by applied mathematicians who have not only ample knowledge and experiences but also perceptual abilities of recognizing complex geometric objects. In the process of experts' understanding of ODEs, qualitative analysis precedes quantitative analysis. The role of qualitative analysis is to grasp the rough picture of the behavior of a given system of ODEs and to focus the scope and range of quantitative analysis. It is often the case that human experts' work is so substantial that it results in a full journal paper such as (Matsumoto *et al.*, 1985). It is quite challenging to analyze and model experts' ability and skills.

Recently, automating qualitative analysis of system of ODEs has been attempted by several authors (Sacks, 1991; Nishida and Doshita, 1991; Nishida *et al.*, 1991; Kalagnanam, 1991; Zhao, 1991). Unfortunately, most of these methods are only applicable to two-dimensional flows.

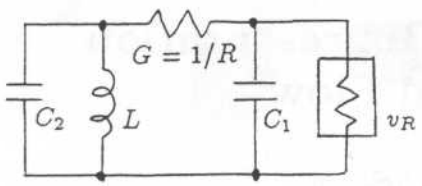
Qualitative analysis of three-dimensional flow is quite challenging both to human experts and to the current AI technology. To human experts, three-dimensional flow is hard to grasp intuitively even if it is visualized, for it is abstract and does not provide enough visual constraints for recovering three-dimensional geometry from visual presentation. In addition, flow itself may sometimes be fairly complex.

To the AI technology, the task is not easy. Firstly, some mathematical theories, such as existence and non-existence theorems, are hard to interpret procedurally. Secondly, AI does not have enough experience with representing and reasoning about geometric objects in continuous space. Finally, numerical errors are inevitable which may cause serious logical errors.

To break through the barrier, we are building a program called PSX3 that can autonomously explore the flow in a three-dimensional phase space by integrating AI and numerical techniques. A key issue is designing an effective representation that would allow us to integrate experts' high level decision making process and powerful numerical analysis methods. I point out that quasi-symbolic representation called *flow mappings* is an effective means of representing qualitative aspects of three-dimensional flow. We have proposed the notion of flow mappings as a means of representing flow in (Nishida and Doshita, 1991; Nishida *et al.*, 1991). However, we have only demonstrated their utility for two-dimensional flows. We have yet to validate the concept for three or higher dimensional flows.

In this paper, I present a method of generating a set of flow mappings for a system of ODEs with three unknown functions. The method is based on a finding that geometric cues for constructing a set of flow patterns can be classified into five categories. I demonstrate how knowledge about interaction of geometric cues is utilized for intelligently controlling numerical computation.

(a) the circuit



(b) characteristic of v_R

$$i_R = g(v_R) = m_0 v_R + \frac{1}{2}(m_1 - m_0)|v_R + B_p| + \frac{1}{2}(m_0 - m_1)|v_R - B_p|$$

(c) the circuit equation

$$\begin{cases} C_1 \frac{dv_{C_1}}{dt} = G(v_{C_2} - v_{C_1}) - g(v_{C_1}) \\ C_2 \frac{dv_{C_2}}{dt} = G(v_{C_1} - v_{C_2}) + i_L \\ L \frac{di_L}{dt} = -v_{C_2} \end{cases}$$

(d) constants and transformation

$$C_1 = 1/9, C_2 = 1, L = 1/7, G = 0.7,$$
$$m_0 = -0.5, m_1 = -0.8, B_p = 1,$$
$$v_{C_1} = x, v_{C_2} = y, i_L = z$$

(e) trace of an orbit near $(0, 0, 0)$

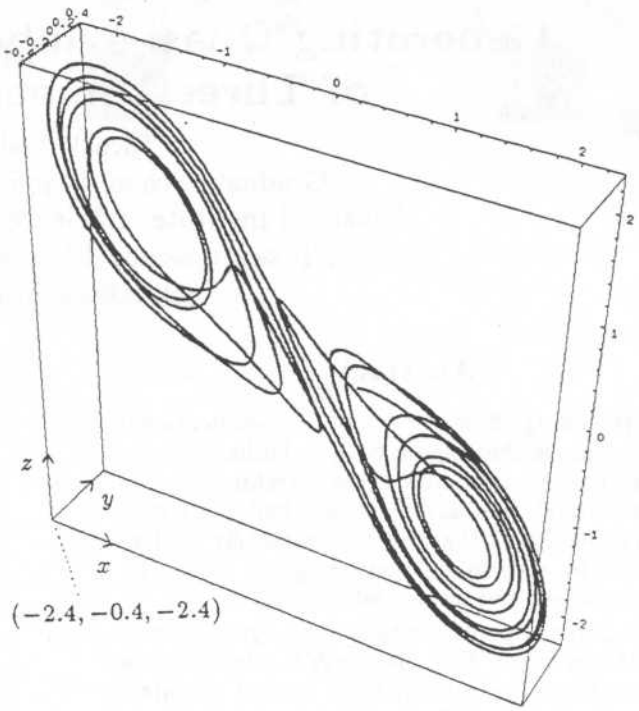


Figure 1: Matsumoto-Chua's circuit (Matsumoto *et al.*, 1985) and a trace of an orbit near a double scroll attractor

Flow in Three-Dimensional Phase Space

In this paper, we consider qualitative behavior of systems of ODEs of the form:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in \mathrm{R}^3$ and $f : \mathrm{R}^3 \to \mathrm{R}^3$. From geometric points of view, we can think of (1) as introducing a *vector field* in the three-dimensional *phase space* spanned by three unknown functions $\mathbf{x} = \{x(t), y(t), z(t)\}$ in the sense that it defines flow vector $\frac{d\mathbf{x}}{dt} = \{\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}\}$ at each point in the phase space. Given a vector field, we can think of *orbits* or trajectories resulting from traversing the phase space according to the vector field. The qualitative behavior of (1) can be grasped by studying the topological structure of the *phase portrait*, the set of all orbits in the phase space.

For a while, I focus on systems of piecewise linear ODEs in which f is represented as a collection of linear functions and constants.[1] Although they are but a subclass of ODEs, systems of piecewise linear ODEs equally exhibit complex behaviors under certain conditions.

For example, consider a system of piecewise linear ODEs:

$$\begin{cases} \frac{dx}{dt} = -6.3x + 6.3y - 9g(x) \\ \frac{dy}{dt} = 0.7x - 0.7y + z \\ \frac{dz}{dt} = -7y \end{cases} \tag{2}$$

where,

$$g(x) = \begin{cases} -0.5x + 0.3 & (x < -1) \\ -0.8x & (-1 \le x \le 1) \\ -0.5x - 0.3 & (1 < x). \end{cases}$$

System of ODEs (2) is obtained by simplifying the circuit equations of Matsumoto-Chua's circuit (third order, reciprocal, with only one nonlinear, 3-segments piecewise linear resistor v_R; see Figure 1a).

In spite of its simplicity in form, (2) exhibits a fairly complex behavior, for the phase portrait contains a chaotic attractor[2] with a "double scroll" structure, that is, two sheet-like thin rings curled up together into spiral forms as shown in Figure 1e (Matsumoto *et al.*, 1985). Orbits approach the attractor as time goes and manifest chaotic behaviors as they irregularly transit between the two "rings."

[1] Later, I will discuss how the method presented can be extended to cases in which only general restrictions (continuity) are posed on f.

[2] Roughly, an attractor is a dense collection of orbits that nearby orbits approach as $t \to \infty$. The reader is referred to (Guckenheimer and Holmes, 1983) for complete definition and discussion.

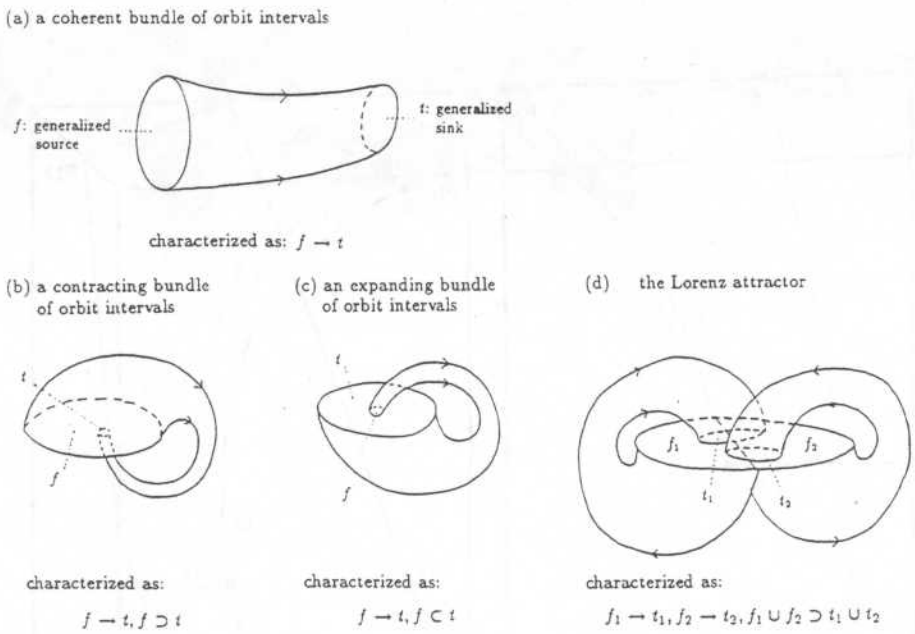


Figure 2: A flow mapping and an aggregated bundle of coherent intervals of orbits

Chaotic attractors may exist only in three or higher dimensional phase space. This fact makes analysis of high dimensional flows significantly harder than two-dimensional flows. Analysis of the double-scroll attractor was reported in a full journal paper (Matsumoto *et al.*, 1985) in applied mathematics.

Flow Mappings as Representation of Flow

We would like to represent flow using finite-length, quasi-symbolic notations, for many powerful AI techniques assume the input to be symbolically represented. The key idea I present in this paper is to partition orbits into intervals (*orbit intervals*) and aggregate them into "coherent" bundles (hereafter, *bundles of orbit intervals*) so that the flow can be represented as a sum of finitely many bundles of orbit intervals. I define the *coherency* of orbit intervals with respect to a finite set of *sensing planes* arbitrarily inserted into the phase space: a bundle of orbit intervals Φ is *coherent*, if all orbit intervals in Φ come from the same *generalized source* (or *g-source*) and go to the same *generalized sink* (or *g-sink*) without being cut by any sensing plane, where g-source and g-sink are either (a) a fixed point or a repellor/attractor with more complex structure, or (b) a singly connected region of a sensing plane.

A flow mapping represents a bundle of orbit intervals as a mapping from the g-source to the g-sink. Thus, it is mostly symbolic. However, it is not completely symbolic as we represent the shape of g-sources and g-sinks approximately. Figure 2a schematically illustrates a bundle of orbit intervals and its representation by a flow mapping, where the g-sinks and g-sources are connected regions of a sensing plane. We are interested in minimal partitioning of flow into coherent bundles of orbit intervals that would lead to minimal description length.

Figure 3 shows minimal partitioning of the flow of Matsumoto-Chua equation (2) in a three-dimensional region $R : 1 \leq x \leq 3, -2 \leq y \leq 2, -3 \leq z \leq 3, 0.566(x-1.5)-0.775y+0.281(z+1.05) \geq 0$ into bundles of orbit intervals. Plane $0.566(x-1.5)-0.775y+0.281(z+1.05) = 0$ is a two-dimensional eigenspace and line $p_{26}p_{29}$ is a one-dimensional eigenspace of a fixed point p_{26}. Orbits in region $0.566(x-1.5)-0.775y+0.281(z+1.05) > 0$ approach the two-dimensional eigenspace with turning around the eigenspace $p_{26}p_{29}$. As they approach the two-dimensional eigenspace, the spiral becomes bigger and diverges. The flow in R can be partitioned into fifteen bundles of orbit intervals. For example, orbit intervals entering R through region $v_1v_5p_{41}p_{40}p_{10}$ can be partitioned into five bundles of orbit intervals:

$$\begin{aligned}
& \Phi_1 : v_1p_2p_3p_4p_5 \rightarrow v_1p_1p_8p_6p_5 \\
\oplus\ & \Phi_2 : p_2v_5p_{43}p_{39}p_{38}p_{30}p_{28}p_4p_3 \\
& \qquad \rightarrow p_1p_{51}p_{35}p_{36}p_{34}p_{23}p_{22}p_9p_8 \\
\oplus\ & \Phi_3 : p_5p_4p_{28}p_{32}p_{31}p_{10} \rightarrow p_{11}p_{15}p_{38}p_{30}p_{31}p_{10} \\
\oplus\ & \Phi_4 : p_{29}p_{30}p_{31}p_{32}p_{28}p_{30} \rightarrow p_{25}p_{23}p_{24}p_{23}p_{34}p_{33} \\
\oplus\ & \Phi_5 : p_{39}p_{43}p_{41}p_{40} \rightarrow p_{44}p_{43}p_{41}p_{42}.
\end{aligned}$$

The advantage of flow mappings is threefold. First, they can be used to locate important orbits such as repellors[3] and attractors. Thus, flow mappings representing bundle as shown in Figure 2b and Figure 2c suggest possible existence of attractors and repellors,

[3] Repellors are orbits that repel nearby orbits (or attract nearby orbits as $t \rightarrow -\infty$).

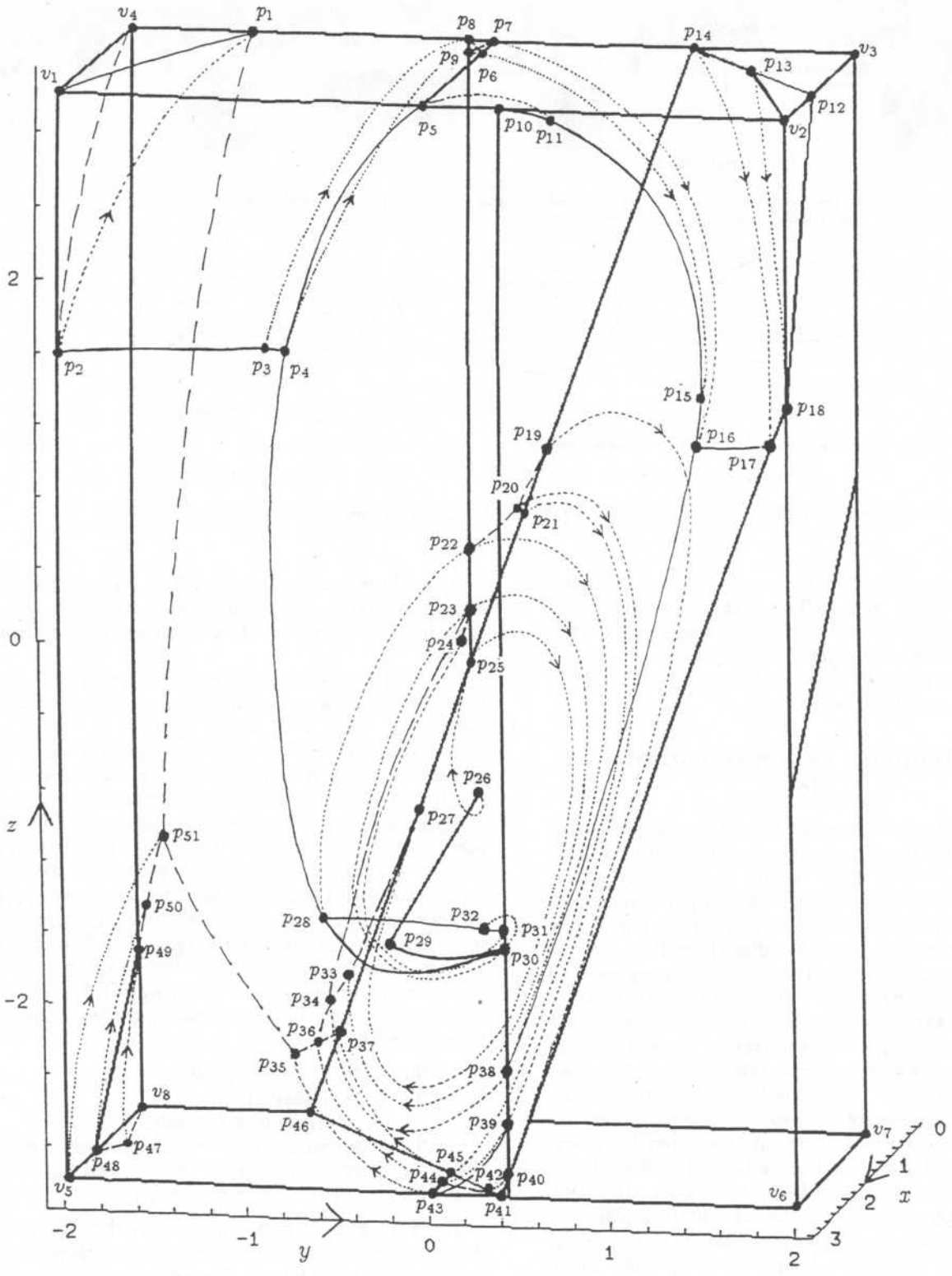


Figure 3: Anatomy of flow of Matsumoto-Chua equation (2) in region $R : 1 \leq x \leq 3, -2 \leq y \leq 2, -3 \leq z \leq 3, 0.566(x - 1.5) - 0.775y + 0.281(z + 1.05) \geq 0$

respectively. Figure 2d characterizes the structure of the Lorenz attractor, known as an chaotic attractor (Guckenheimer and Holmes, 1983). Detection of these structure motivates focusing more detailed analysis.

Second, they provide an abstract and concise characterization of flow, allowing to capture the flow at a desired grain size. For example, when distances between sensing planes is sufficiently large, an attractor with a complex internal structure may be abstracted simply as a source of orbits, while the smaller the distance becomes, the finer resolution one may gain to see the internal structure of an attractor.

Third, flow mappings economically represent bundles of orbit intervals, three-dimensional geometric objects, only by referring to a pair of two or less dimensional geometric objects. This means that flow mappings convey less but only essential information for reasoning about qualitative behavior than more straightforward representation, such as polyhedral approximation (Zhao, 1991).

Generating Flow Mappings for Three-Dimensional Flow

In order to design an algorithm of generating flow mappings for a given flow, I have studied the relationships between geometric patterns that flow makes on the surface of sensing planes and the topological structure of underlying orbit intervals, and found that they can be classified into five categories called *geometric cue interaction patterns*. My algorithm makes use of geometric cue interaction patterns as local constraints both to focus numerical analysis and interpret the result.

Geometric Cues

Let us consider characterizing flow in a convex region called a *cell* which is bounded by sensing planes by a set of flow mappings. In order to do that we study geometry that orbits make on surfaces of a sensing plane.

I classify the surface in terms of the orientation of orbit there. A contingent section S of the surface is called an *entrance section* if S is on a single sensing plane and orbits enter the cell at all points of S except some places where the orbits are tangent to the surface. An *exit section* is defined similarly. An entrance or exit section (*e.g.*, exit section $v_1p_5p_7v_4$) may be further partitioned into smaller sections (*e.g.*, sections $v_1p_1v_4$, $v_1p_5p_6p_8p_1$, and $p_6p_7p_8$) by one or more *section boundary* (*e.g.*, v_1p_1 and p_6p_8), which may be either (a) an intersection of sensing planes, (b) an image or an inverse image of a section boundary, or (c) an intersection of a two-dimensional eigenspace and the cell surface. Section boundaries play an important role as primary geometric cues on the surface.

Tangent sections separate entrance and exit sections. Tangent sections are further classified into two categories: a *concave section* (*e.g.*, p_5p_7) at which orbits come from the inside the cell, touch the surface, and go back to the cell, and a *convex section* (*e.g.*, v_1p_5 and $p_{10}p_{31}$) at which orbits come from the outside the cell, touch the surface, and leave the cell.

An intersection of an eigenspace and the surface[4] is called a *pole* or a *ground* depending on whether the eigenspace is one-dimensional or two-dimensional, respectively. In Figure 3, point p_{29} is a pole and line segments $p_{41}p_{40}$, $p_{40}p_{18}$, $p_{18}p_{12}$, *etc* are grounds.

A *thorn* is a one-dimensional geometric object which thrusts outward from section boundary into an entrance/exit section. In Figure 3, there are two thorns: $p_{23}p_{24}$ and $p_{30}p_{29}$. Thorns result from peculiarity of eigenspace.

Interaction of geometric cues may result in a *junction* of various types. For example, section boundaries p_2p_4 and $p_5p_4p_{28}$ in Figure 3 meet at p_4, making a *T-junction*, while $v_1p_1p_{51}$ and $v_4p_1p_8$ make an *X-junction* at p_1.

Some geometric cues such as fixed point p_{26} or convex section $p_{10}p_{31}$ are *trivial* in the sense that they can be easily recognized by local computation without tracking orbits, while others such as a T-junction at p_4 are *nontrivial* because they cannot be found without predicting their existence and validation by focused numerical analysis.

Geometric Cue Interaction Patterns

I have classified interactions of geometric cues into five categories, as shown in Figure 4. Each pattern is characterized by a *landmark orbit* such as X_1X_2 in an X-X interaction or T_1X in a T-X interaction that connect geometric cues.

A X-X ("double X") interaction is an interaction between boundary sections. In Figure 3, example of a X-X interaction is with the landmark orbit p_2p_1.

A T-X and a T-T ("double T") interaction co-occurs with a concave section, which "pushes in" or "pops out" bundle of orbit intervals. In Figure 3, example of a T-X interactions is with landmark orbit $p_{28}p_{22}p_{38}$.

A *Pole-T* interaction results from peculiarity of orbits in an eigenspace of a saddle node. The closer the start (or end) point of an orbit approaches the ground, the closer the end (or start) point of an orbit approaches the pole. Special care is needed for searching for a *Pole-T* interaction when the derivative of the flow at the fixed point has complex eigenvalues, for a boundary edge may turn around the pole infinitely many times.

A *Thorn-T* interaction accompanies peculiarity, too. A T-junction consisting of a section boundary, a convex section, and a concave section is mapped to/from the top of a thorn. Points on the section boundary of the T_i-junction are mapped to/from the concave section, points on which are in turn mapped to/from the body of the thorn.

[4] For simplicity, I assume that no surface of a cell is an *eigenspace*, which is a special subspace consisting of orbits tending to/from a saddle node.

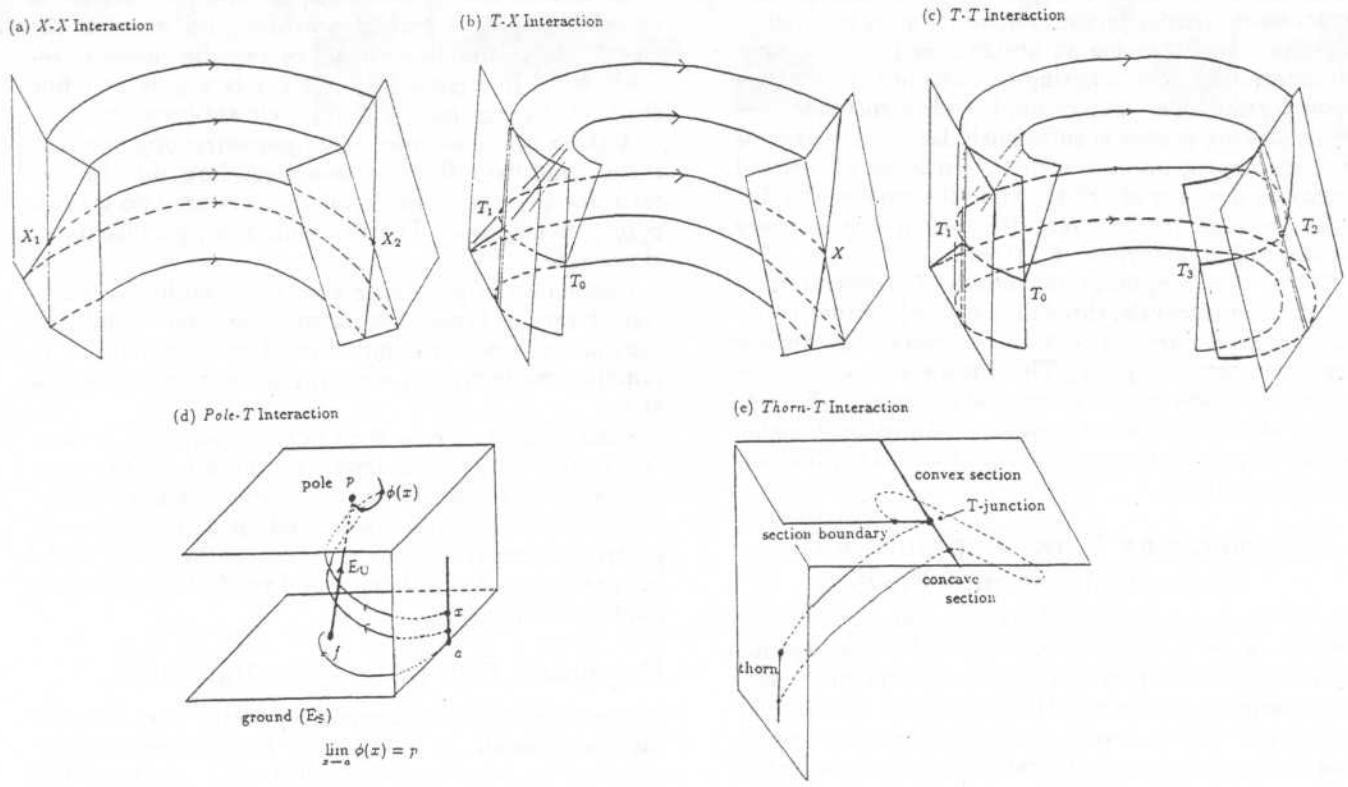


Figure 4: Geometric Cue Interaction Patterns

(stage 1) trivial geometric cue recognition: identify trivial geometric cues on the surface and classify each surface into entrance, exit, concave, and convex sections;

(stage 2) nontrivial geometric cue recognition: seek nontrivial geometric cues by tracking orbits from trivial geometric cues and matching the result with geometric cue interaction patterns;

(stage 3) cell surface partitioning: partition entrance and exit sections based on trivial and nontrivial geometric cues;

(stage 4) flow mapping generation: generate flow mappings by analyzing how partitioned cell surfaces (and a fixed point if any in the given cell) are correlated by the flow.

Figure 5: A procedure for generating flow mappings for a given cell

Analysis Procedure

Roughly, a procedure for generating flow mappings for a given cell is divided into four stages: trivial geometric cue recognition, nontrivial geometric cue recognition, cell surface partitioning, and flow mapping generation, as shown in Figure 5.

Trivial Geometric Cue Recognition At this stage, the surface is classified with respect to the orientation of flow and trivial geometric cues are recognized. For piecewise linear flow, computation is relatively simple and complete information can be obtained almost always. Table 1 shows class of computation required to solve major subproblems arising at this stage.

Nontrivial Geometric Cues Recognition This is the most essential part of the procedure. Reasoning is required to find nontrivial geometric cues and correlate them with trivial geometric cues. The procedure for this stage is based on knowledge about geometric cue interaction patterns. Currently, the knowledge is encoded procedurally as a process of planning, monitoring, and interpreting the result of numerical computation. First, orbits are tracked forward and/or backward from concave sections, section boundaries, and the ground. Note that we can track only finitely many orbits, so that care must be taken not to miss important geometric cues. Then, the images and/or inverse images of trivial geometric cues are examined to see whether there is a possibility of the existence of a nontrivial geometric cue. Explanation is sought that may correlate the geometric cues, by consulting a library of geometric cue interaction patterns.

Cell Surface Partitioning The goal of this stage is to find a minimal partitioning of the cell surface so that each region may correspond to a g-source or g-sink of a maximal bundle of orbit intervals. Minimal cycles consisting of pieces of one-dimensional objects (either tangent sections, section boundaries or ground) as well as zero-dimensional objects (junctions) are sought from the set of trivial and nontrivial geometric cues identified at previous stages, and they are identified as a boundary of a two-dimensional g-sources/g-sinks.

Flow Mapping Generation Flow mappings are generated by examining records of tracking orbits to see how g-sources/g-sinks[5] are correlated with each other by orbit intervals involved in the given cell.

Implementation

The analysis procedure described above is implemented as PSX3 (Nishida, 1993), except for procedures for *Pole-T* and *Thorn-T* interactions.

In order to cope with increasingly complex mathematical problems, we have connected PSX3 with Mathematica (Wolfram, 1988). Mathematica also provides us with a fancy graphics to help understanding three-dimensional geometric objects. PSX3 automatically produces a command sequence which allows Mathematica to produce a three-dimensional graphics for the user.

We have tested the current version of PSX3 against a few systems of piecewise linear ODEs whose flow does not contain *Pole-T* or *Thorn-T* interactions.

Example (1)

Let us see how early stages of the analysis procedure work for the top-left portion of the cell shown in Figure 3. Figure 6a shows the result of trivial geometric cue recognition. Figure 6b and 6c show the way orbits are tracked and nontrivial geometric cues are recognized. In the former, as p_is move downward from vertex v_1, their images $\phi(p_i)$s jump from the top plane to the rear plane. For this case, PSX3 assumes the existence of an X junction. In the latter, as q_js go to the right, their inverse images $\phi^{-1}(q_j)$ jump from the left plane to the front plane, for which PSX3 assumes another X junction. Figure 6d shows that a X-X interaction is chosen an underlying geometric cue interaction pattern that correlates the two X-junctions.

Example (2)

Now Consider a linear flow

$$\begin{cases} \frac{dx}{dt} = -x - 2y + 2 \\ \frac{dy}{dt} = 3x - y - \frac{1}{2}z - 2 \\ \frac{dz}{dt} = -\frac{1}{2}z \end{cases} \tag{3}$$

in cell:

$$0.2 \leq x \leq 2, 0.2 \leq y \leq 2, -0.2 \leq z \leq 2, 1 - \frac{3}{2}x \leq y.$$

Figure 7a shows some of trivial geometric cues together with some orbit intervals and the frame of the cell. Figure 7b shows the result of cell surface partitioning that PSX3 has produced together with internal representation of flow mappings:

```
((CYCLE-40 FIXED-POINT-1) (CYCLE-45 CYCLE-33)
(CYCLE-36 CYCLE-47) (CYCLE-50 CYCLE-41)
(CYCLE-51 FIXED-POINT-1) (CYCLE-44 CYCLE-52)
(CYCLE-54 CYCLE-53) (CYCLE-55 CYCLE-46)
(CYCLE-56 CYCLE-37) (CYCLE-57 CYCLE-35)
(CYCLE-58 FIXED-POINT-1) (CYCLE-59 CYCLE-49)
(CYCLE-60 CYCLE-42) (CYCLE-61 CYCLE-39)
(CYCLE-62 CYCLE-34) (CYCLE-63 FIXED-POINT-1)
(CYCLE-64 CYCLE-48) (CYCLE-65 CYCLE-43)
(CYCLE-66 CYCLE-38) (CYCLE-67 CYCLE-32)
(CYCLE-68 FIXED-POINT-1)),
```

which is a list of pairs of the form (α β) which represents a flow mapping from α to β by orbit intervals contained in the cell. For example, the first item says that all points in CYCLE-40 are mapped to fixed point FIXED-POINT-1 (see also annotations given to Figure 7b).

[5] Two-dimensional ones should be found in cell surface partitioning, and the rest (zero-dimensional ones; *i.e.*, fixed points) are to be found in trivial geometric cue recognition, if any in the give cell.

Table 1: Class of Computation Required to Solve Major Subproblems at (stage 1)

required information	class of computation
location of fixed point	solving linear simultaneous equation
type of fixed point	computing eigenvector of a 3×3 matrix (which requires solving a cubic equation with one unknown)
eigenspace of saddle node	computing eigenvector of a 3×3 matrix
location of tangent section	solving linear simultaneous equation
subcategorization of tangent sections	computing the sign of a polynomial
orbit	integration (If one would like to take full advantage of linear system, s/he could resort to an analytic solution method whose hardest part is computing eigenvalues of a 3×3 matrix.)
intersection of an orbit and the boundary	integration and computing the sign of a polynomial (If one would like to full advantage of linear system, s/he could obtain the result by computing eigenvalues of a 3×3 matrix and solving nonlinear simultaneous equation with three unknowns.)

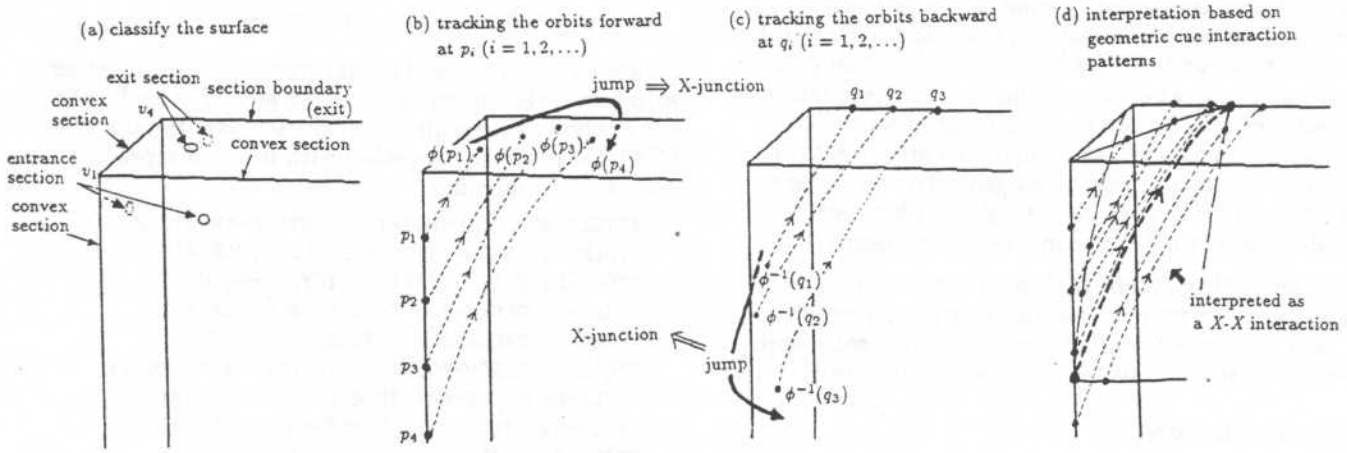
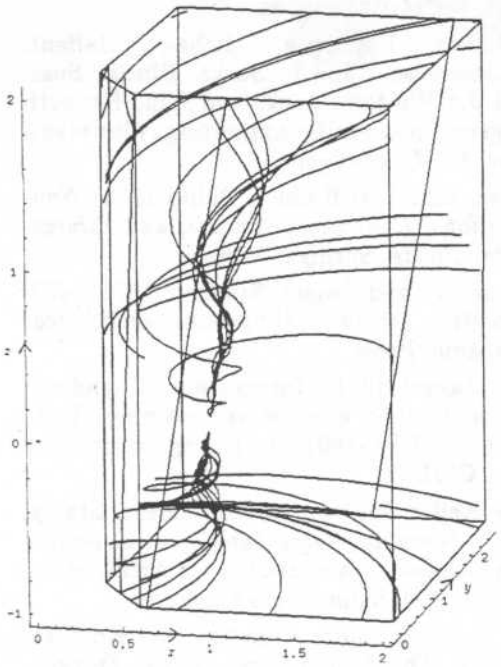


Figure 6: A process of generating flow mappings for (2)

(a) trivial geometric cues, orbit intervals, and the frame of the cell



(b) result of cell surface partitioning

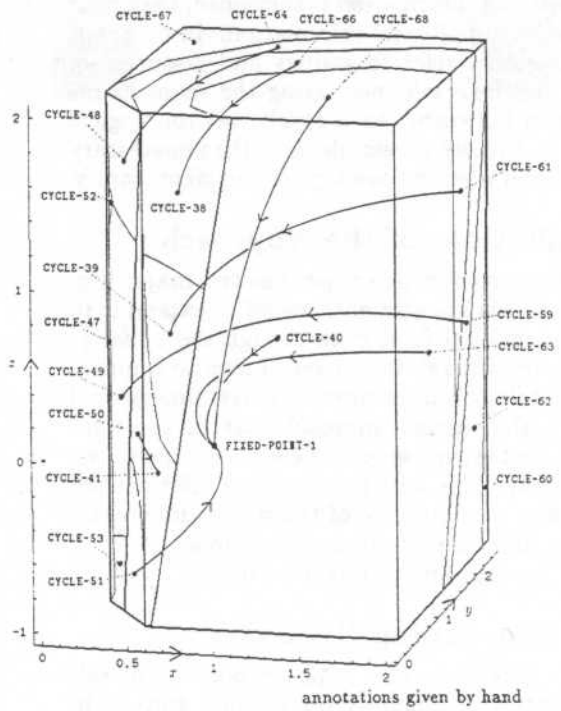


Figure 7: Trivial geometric cue recognition and cell surface partitioning for (3) by PSX3

Generalization to Nonlinear ODEs

So far, I have carefully limited our attention to systems of piecewise linear ODEs, for which the flow in each cell is linear. However, it is not hard to extend the method to nonlinear ODEs, if we are to handle only non-degenerate (*i.e.*, *hyperbolic*) flows.[6] What to be added is twofold: (a) a routine which will divide the phase space into cells that contain at most one fixed point, and (b) a general nonlinear (non-differential) simultaneous equation solver. Neither of these are very different from those that have been implemented for analyzing two-dimensional flow (Nishida and Doshita, 1991).[7]

Another thing we might have to take into account is the fact that certain assumptions such as planarity of an eigenspace do not hold any more. Fortunately, local characteristic of a nonlinear flow is equivalent to a linear flow, as linear approximation by Jacobian preserves local characteristics of nonlinear flow as far as the flow is hyperbolic. Thus, the local techniques work. Globally, we have not made any assumption that takes advantage of the linearity of local flow, so it also works. Implementation of these codes is, however, left for future.

[6] Degenerate flows are rare, even though generative property (Hirsch and Smale, 1974) (a proposition that the probability of observing a degenerate flow is *zero*) does not hold for three-dimensional flow.

[7] It should be noted that a nonlinear simultaneous equation solver may not always produce a complete answer. Dealing with incompleteness of numerical computation is open for future research. Some early results are reported in (Nishida *et al.*, 1991).

Related Work

This work can be thought of as development of a basic technology for intelligent scientific computation (Abelson *et al.*, 1989; Kant *et al.*, 1992), whose purpose is to automate scientific and engineering problem solving. In this paper, I have concentrated on deriving quasi-symbolic, qualitative representation of ODEs by intelligently controlling numerical analysis. Previous work in this direction involves: POINCARE (Sacks, 1991), PSX2NL (Nishida and Doshita, 1991), Kalagnanam's system (Kalagnanam, 1991), and MAPS (Zhao, 1991). KAM (Yip, 1991) is one of the frontier work, though it is for discrete systems (difference equations). Unfortunately, these systems are severely limited to two-dimensional flows, except MAPS.

Zhao claims MAPS (Zhao, 1991) can analyze n-dimensional flows too. MAPS uses polyhedral approximation of collection of orbits as intermediate internal representation. As polyhedral approximation represents rather the shape of flow than the topology,

it is not suitable for reasoning about qualitative aspects in which the topology of the flow is a main issue. The more precise polyhedral approximation becomes, the more irrelevant information is contained, making it harder to derive topological information. In contrast, flow mappings only refer to g-sinks and g-sources of bundles of orbit intervals, neglecting the shape of orbit intervals in between. As a result, (a) topological information is directly accessible, and (b) unnecessary computation and memory are suppressed significantly.

Limitations of the Approach

The method reported in this paper has two major limitations. First, it is not straightforward to extend it to general n-dimensional flow, even though the underlying concepts are general, for I have chosen to improve efficiency by taking advantages of three-dimensional flow. Second, the current approach may be too rigid with respect to the topology of flow. Sometimes, we may have to pay a big cost for complete information, especially when the topology of the flow is inherently complex (*e.g.*, fractal basin boundary (Moon, 1987)).[8] Making the boundary fuzzy might be useful.

Concluding Remarks

The problem tackled in this paper is both hard, relevant and central to qualitative reasoning and AI in general.

It is hard for a solution to the problem itself deserves a full journal paper in applied mathematics in a decade ago. Thus, it is a real world problem.

It is relevant and central to qualitative reasoning, for implementing an ability of reasoning about topology of geometric patterns in a continuous domain is essential to qualitative reasoning and will advance the current technology as it significantly generalize envisioning techniques. Symbolizing continuous world is one of the main issue of qualitative reasoning.

Automating qualitative analysis by intelligently controlled numerical analysis involves reasoning about complex geometry and topology under incomplete information.

In this paper, I have pointed out that complex geometric patterns of solution curves of systems of ODEs can be decomposed into a combination of simple geometric patterns called geometric cue interaction patterns, and shown how they can be utilized in qualitative analysis of three-dimensional flow.

As a further work, important future work involves generalization into n-dimensional flow and abstraction and focusing. From my own experience, it seems steady to proceed step by step by gradually increasing the number of dimensionality, rather than directly dive into n-dimensional flows, for applied mathematicians are more carefully investigating ODEs one by one, and deliberate in generalization.

[8] Note that this does not mean the current approach is not suitable for analyzing chaos. Remember that the example I have used in this paper is Matsumoto-Chua's double-scroll attractor which is known as a chaotic attractor.

References

Abelson, Harold; Eisenberg, Michael; Halfant, Matthew; Katzenelson, Jacob; Sacks, Elisha; Sussman, Gerald J.; Wisdom, Jack; and Yip, Kenneth 1989. Intelligence in scientific computing. *Communications of the ACM* 32:546–562.

Guckenheimer, John and Holmes, Philip 1983. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* Springer-Verlag.

Hirsch, Morris W. and Smale, Stephen 1974. *Differential Equations, Dynamical Systems, and Linear Algebra.* Academic Press.

Kalagnanam, Jayant 1991. Integration of symbolic and numeric methods for qualitative reasoning. Technical Report CMU-EPP-1991-01-01, Engineering and Public Policy, CMU.

Kant, Elaine; Keller, Richard; and Steinberg, Stanly, editors 1992. *Working Notes Intelligent Scientific Computation.* American Association for Artificial Intelligence. AAAI Fall Symposium Series.

Matsumoto, Takashi; Chua, Leon O.; and Komuro, Motomasa 1985. The double scroll. *IEEE Transactions on Ciruits and Systems* CAS-32(8):798–818.

Moon, Francis C. 1987. *Chaotic Vibrations — An Introduction for Applied Scientists and Engineers.* John Wiley & Sons.

Nishida, Toyoaki and Doshita, Shuji 1991. A geometric approach to total envisioning. In *Proceedings IJCAI-91.* 1150–1155.

Nishida, Toyoaki; Mizutani, Kenji; Kubota, Atsushi; and Doshita, Shuji 1991. Automated phase portrait analysis by integrating qualitative and quantitative analysis. In *Proceedings AAAI-91.* 811–816.

Nishida, Toyoaki 1993. Automating qualitative analysis of three-dimensional flow. (unpublished research note, in preparation).

Sacks, Elisha P. 1991. Automatic analysis of one-parameter planar ordinary differential equations by intelligent numeric simulation. *Artificial Intelligence* 48:27–56.

Wolfram, Stephen 1988. *Mathematica - A System for Doing Mathematics by Computer.* Addison-Wesley Pub. Co. (the second edition was published in 1991).

Yip, Kenneth Man-kam 1991. Understanding complex dynamics by visual and symbolic reasoning. *Artificial Intelligence* 51(1-3):179–222.

Zhao, Feng 1991. Extracting and representing qualitative behaviors of complex systems in phase spaces. In *Proceedings IJCAI-91.* 1144–1149.