ELSEVIER

Artificial Intelligence 75 (1995) 3-29

Artificial Intelligence

Qualitative analysis of behavior of systems of piecewise linear differential equations with two state variables

Toyoaki Nishida[a,*], Shuji Doshita[b]

[a] *Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-01 Japan*

[b] *Department of Information Science, Kyoto University, Sakyo-ku, Kyoto, 606-01 Japan*

Abstract

The set of all solution curves in the phase space for a system of ordinary differential equations (ODEs) is called the *phase portrait.* A phase portrait provides global qualitative information about how a given system of ODEs behaves under different initial conditions. We are developing a method called *Topological Flow Analysis* (TFA) for automating analysis of the topological structure of the phase portrait of systems of ODEs.

In this paper, we describe the first version of TFA for systems of piecewise linear ODEs with two state variables. TFA has several novel features that have not been achieved before. Firstly, TFA enables to grasp characteristics of all behaviors of a given system of ODEs. Secondly, TFA allows to symbolically represent behaviors in terms of critical geometric features in the phase space. Finally, TFA integrates qualitative and quantitative analysis. The current version of TFA has been implemented as a program called PSX2PWL using Common Lisp.

1. Introduction

Analysis of dynamical systems is essential to model-based reasoning as a means of deriving behavior from structure. One of the major concerns of qualitative reasoning has been deriving qualitative behavior of continuous dynamical systems. Unfortunately, most conventional qualitative reasoning techniques proposed so far lack an adequate description language of complex behavior and completeness of prediction due to explosion of ambiguities.

[*] Corresponding author. When this article was originally written, this author was with the Department of Information Science, Kyoto University.

0004-3702/95/$09.50 © 1995 Elsevier Science B.V. All rights reserved
SSDI 0004-3702(94)00062-X

To address these problems, we are developing a method called *Topological Flow Analysis* (TFA) for analyzing the topological structure of the phase portrait of systems of ordinary differential equations (ODEs). TFA is based on dynamical systems theory (DST), a sophisticated mathematical theory about qualitative analysis of dynamical systems.

In this paper, we present the first version of TFA for systems of piecewise linear ODEs (PWL-ODEs) with two state variables. TFA has several novel features. Firstly, TFA enables to grasp characteristics of behaviors of a given system of PWL-ODEs under all possible initial conditions. TFA does this not by directly computing or approximating all possible solutions, but by reasoning at the abstract level. Secondly, TFA allows to symbolically represent behaviors in terms of critical geometric features in a phase space. The use of symbolic representation does not only make sense as an explicit description of the behavior, but it also helps to navigate the analysis process. Finally, TFA integrates qualitative and quantitative analysis. Qualitative analysis guides the overall analysis process. Quantitative analysis helps qualitative analysis resolve ambiguity.

The current version of TFA, implemented using Common Lisp, is called PSX2PWL. [1]

2. Dynamical systems theory

TFA is based on *Dynamical Systems Theory* (DST for short) [2,3]. DST, called a qualitative theory of dynamical systems, provides a powerful method of reasoning about the geometric and topological features of solution curves of systems of ODEs in a phase space. This enables, for example, to derive qualitative information of the behavior of a given system of ODEs, even when a precise form of the solution is not known. In this section, we introduce basic notions of DST.

Consider the following system of ODEs:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = f(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ is a vector of *state variables* $(x_1, \ldots, x_n)$ each of which gives some value in $\mathbb{R}$ as a function of $t \in \mathbb{R}$. An n-dimensional space spanned by the set of all state variables $\{x_i\}$ is called the *phase space*. For each point $\boldsymbol{c} = (c_1, \ldots, c_n)$ in the phase space, Eq. (1) specifies the rate and the orientation of state change:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}\Big|_{x=c} = \left(\frac{\mathrm{d}x_1}{\mathrm{d}t}, \ldots, \frac{\mathrm{d}x_n}{\mathrm{d}t}\right)\Big|_{x=c}.$$

In other words, Eq. (1) defines a *vector field* in the phase space. A specific solution corresponding to initial state $\boldsymbol{a} = (a_1, \ldots, a_n)$ is a trajectory such that it passes through the point corresponding to $\boldsymbol{a}$ in the phase space and it is tangent to the vectors specified by the vector field at each point (Fig. 1). Such a trajectory is called a *solution curve*, or an *orbit*. The collection of all orbits in a phase space is called the *phase portrait*.

[1] Originally, it was called PSX (Phase Space eXplorer). Now, we have several other versions of PSXs for more complex systems of ODEs, as described in the postscript of this paper.

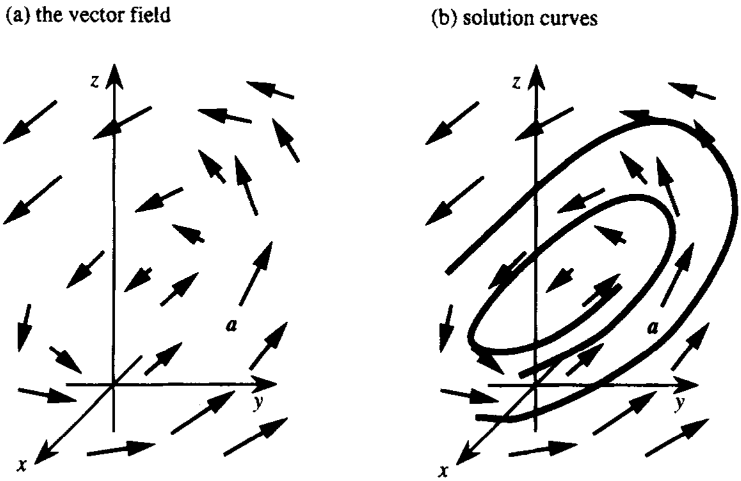

Fig. 1. Geometric meaning of system of ODEs.

Theoretically, we can think of an orbit passing on $x \in U \subset \mathbb{R}^n$ as a mapping $\phi_x(t) : \mathbb{R} \rightarrow V$ which maps x to $y \in V \subset \mathbb{R}^n$ as a function of $t \in \mathbb{R}$. We can also think of (1) as specifying a "flow" $\Phi(x,t) = \phi_x(t) : U \times \mathbb{R} \rightarrow V$. If uniqueness of solution holds for a given system of ODEs, orbits never intersect with others nor with themselves (*the non-intersection constraint*).

There is an obvious correspondence between the geometric properties of orbits and the aspects of dynamical behavior. For example, points in a phase space at which the right-hand side of Eq. (1) is zero are called *fixed points*, and they correspond to equilibrium states which will not evolve for ever. Fixed points are further classified into *sinks*, *sources* and *saddle nodes*. Orbits near a sink and a source arbitrarily approach the sink as $t \rightarrow \infty$ and $-\infty$, respectively, while both kinds of orbits exist around a saddle node. Closed orbits correspond to periodic behaviors.

DST is particularly concerned with analyzing topological structure of asymptotic behaviors of dynamical systems as $t \rightarrow \pm\infty$. In DST, it is proved that orbits in a two-dimensional planar phase space either (1) diverge for place at infinity, (2) approach a fixed point, or (3) approach a closed orbit (called a *limit cycle*), as $t \rightarrow \pm\infty$. Roughly, orbits to which nearby orbits approach as $t \rightarrow \infty$ and $-\infty$ are called *attractors* and *repellors*, respectively. Attractors and repellors play an important role in qualitative analysis of phase portrait. A set of points which an orbit passing a point x in a phase space approaches as $t \rightarrow \infty$

$$\{y \mid \exists t_n(\rightarrow \infty)[\lim_{n\rightarrow\infty} \phi_{t_n}(x) = y]\}$$

is called the *ω-limit set* of x. The α-limit set of x is defined similarly, by replacing ∞ by $-\infty$. Although an ω-limit set becomes empty if the orbit diverges for a place at infinity as $t \rightarrow \infty$, we regard a place at infinity as a special kind of places and allow an ω-limit set to contain a place at infinity. An ω-limit set extended in this way is called

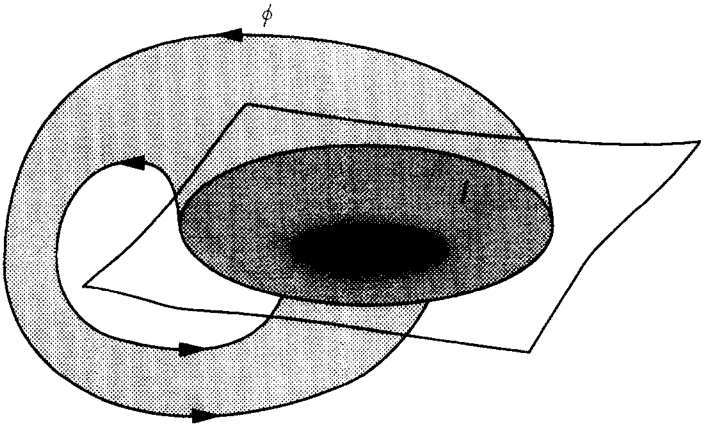

Fig. 2. Contracting recursive bundle of orbit intervals.

a *generalized sink*, or a *g-sink* for short. A similar extension to an α-limit set is called a *generalized source* (or a *g-source* for short).

Unfortunately, it is not straightforward to implement a program that can explore the phase space of systems of ODEs based on DST. For example, DST does not provide us with a procedure for finding an attractor in a phase space, though it gives a definition. Hence, we need to develop a computational theory of representing and reasoning about continuous flow.

3. Reasoning about phase portraits

TFA is a computational theory of DST. TFA is based on several novel ideas.[2]

3.1. Reasoning about bundle of orbits

We focus on bundles of orbits rather than single orbits, to derive useful conclusions which cannot be made about single orbits. Suppose, for example, we have found in a given phase portrait a pattern as shown in Fig. 2, in which a bundle of orbits ϕ is transverse to regions I and J on the same hypersurface such that $J \subset I$. This pattern is called a *contracting recursive bundle of orbit intervals.* All orbits transverse to I are also transverse to J, and never leave the region occupied by ϕ. If the phase space is two-dimensional, region J is finitely bounded, and region I and J do not share a boundary, then ϕ contains one or more attracting limit cycle $\{a_1, \ldots, a_k\}$ as shown in Fig. 3. All orbits transverse to the region $I - J$ approach one of those limit cycles as $t \to \infty$.

[2] In the description below, we attempt to generalize these ideas to n-dimensional phase spaces as much as possible, because most of them are independent of the dimensionality of the phase space.

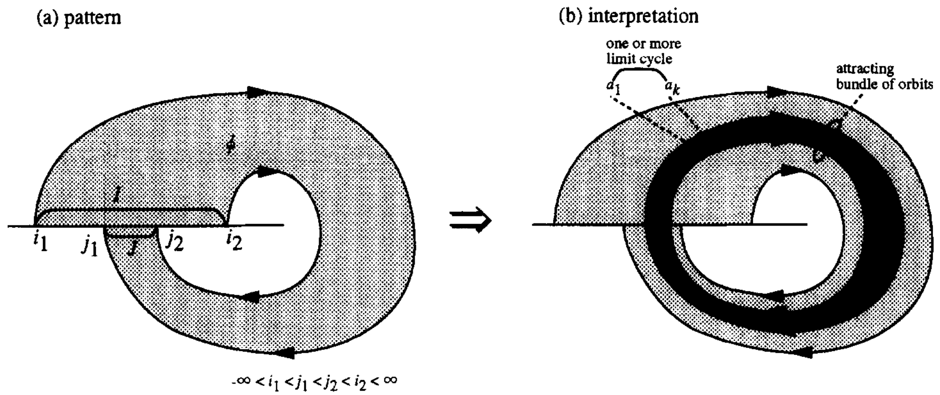

Fig. 3. Circumstance in which existence of an attracting limit cycle is predicted.

Formally, a bundle of orbit intervals is defined as a set of intervals of orbits ϕ such that

(1) no fixed point is involved in the region occupied by ϕ;[3]

(2) for any point p in the internal region occupied by ϕ, there exists a hypersurface s which is transverse to the region occupied by ϕ at region a that contains p and

 (a) all intervals of orbits that belong to ϕ are transverse to s once and only once.

 (b) for any point q in a, there exists an interval of an orbit involved in ϕ which is transverse to s at q.

The hypersurface s in the above definition is called a *cross section* of ϕ. Intuitively, conditions (1) and (2a) require the uniformity of orbit intervals involved in ϕ, and (2b) requests the density and continuity of orbit intervals in ϕ, as shown in Fig. 4.

3.2. *Representing a bundle of orbits as flow mappings between hyperplanes*

We use several $(n-1)$-dimensional hyperplanes, called *sampling hyperplanes*, in an n-dimensional phase space to "sample" data about bundles of orbit intervals. We abstract each bundle of orbits as a sequence of mappings between locally defined $(n-1)$-dimensional hyperplanes.

When $n = 2$, sampling hyperplanes are straight lines, which we call *sampling lines*. Consider a bundle of orbit intervals ϕ that intersects with sampling hyperplanes p_1 and p_2 at r_1 and r_2, respectively, as shown in Fig. 5. An interval ϕ_0 of ϕ delimited by r_1 and r_2 continuously maps points on r_1 to r_2. We represent this as

$$\phi_0 : r_1 \rightarrow r_2$$

and call it a *flow mapping*.

[3] Fixed points may be on the boundary of the region.

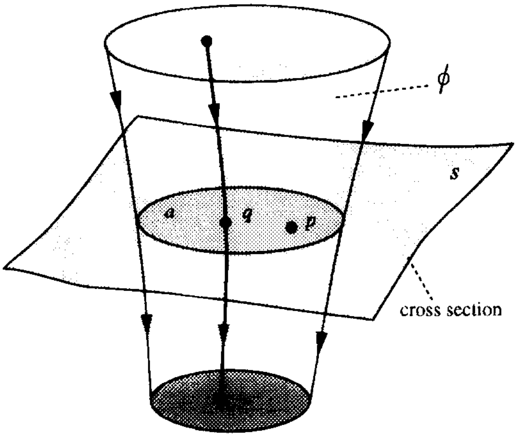

Fig. 4. Bundle of orbit intervals and its cross section.

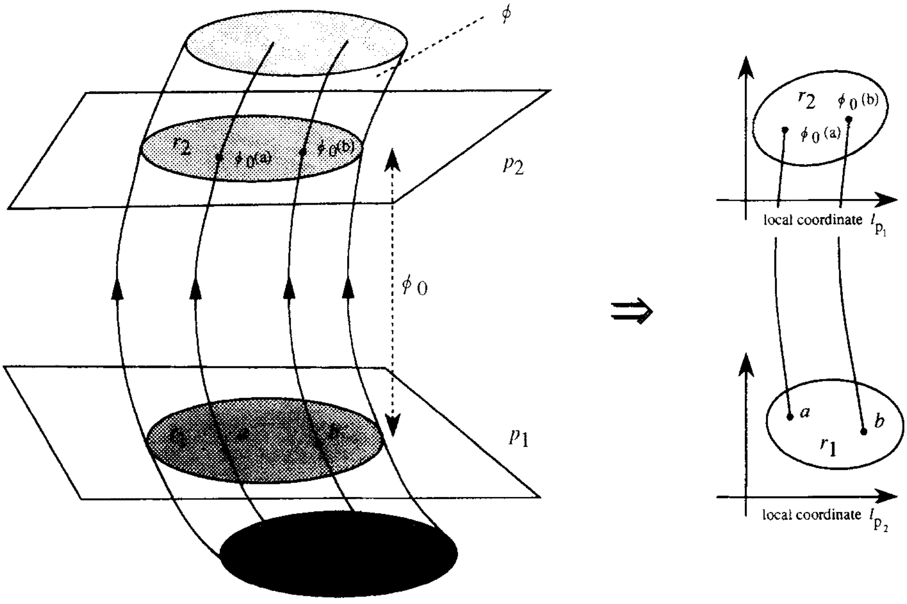

Fig. 5. Representing bundle of orbits as mapping between hyperplanes.

Although it is hard to find an effective representation for r_1 and r_2 in a general n-dimensional phase space, we currently step aside from this problem by limiting our concern to flows in two-dimensional phase spaces, where r_1 and r_2 are line segments which can be completely specified by the location of the two endpoints.

3.3. Deriving global behavior by analyzing the structure of flow mappings

We can derive global behavior by analyzing the structure of flow mappings. Given a couple of bundles of flow mappings $\phi_1 : I \to J$ and $\phi_2 : J \to K$, let us define *composition* $\phi_2 \circ \phi_1$ of ϕ_1 and ϕ_2 as follows:

$$\phi_2 \circ \phi_1(x) = \phi_2(\phi_1(x)) = y \quad \text{iff} \quad \exists z\ [\phi_1(x) = z,\ \phi_2(z) = y].$$

We call a composite flow mapping $\phi_m \circ \cdots \circ \phi_1$ a *contracting recursive mapping*, if the range is a subset of the domain, namely,

$$\phi_m \circ \cdots \circ \phi_1(I) \subset I.$$

Similarly, we call $\phi_m \circ \cdots \circ \phi_1$ an *extending recursive mapping*, if

$$\phi_m \circ \cdots \circ \phi_1(I) \supset I.$$

Existence of a contracting recursive mapping entails existence of an attractor. Similarly, existence of an extending recursive mapping entails existence of a repellor.

3.4. Taking advantage of strong constraints of a two-dimensional piecewise linear flow

In order to make the above ideas computational, we need to develop

- a procedure for finding sampling hyperplanes that produces a useful set of flow mappings for grasping global behaviors, and
- a procedure for abstracting flow as mapping between sampling hyperplanes.

These requirements are not trivial to achieve in general, even for two-dimensional flows. We take advantage of strong constraints on two-dimensional piecewise-linear flows. We have studied the properties of linear flow in convex regions (cells) in two-dimensional phase spaces and have found that the local flow in a cell can be computed by analyzing a few orbits that pass on particular points on the boundary of the cell. Our method of generating flow mappings is to

- partition the phase space into convex regions called *cells* by invariant manifolds and boundaries between linear regions, and
- examine the property of the flow on the boundary of each cell.

This algorithm produces a sufficient set of sampling hyperplanes in the sense that the resulting flow mappings makes it possible to capture all attractors and repellors in the phase space.

4. Properties of systems of PWL-ODEs with two state variables

In this section, we exploit properties of systems of PWL-ODEs with two state variables to more detail.

Consider a system of PWL-ODEs of the form:

$$\left\{ R_i : \left\langle p_i(\boldsymbol{x}) \middle| \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = f_i(\boldsymbol{x}) \right\rangle \right\} \quad (1 \leqslant i \leqslant m, \boldsymbol{x} \in \mathbb{R}^2). \tag{2}$$

Each constituent delimited by a pair of angular brackets is called a *linear region*. Each linear region may be given a unique label R_i and is specified as a logical combination of linear inequalities $p_i(\boldsymbol{x})$ that gives a condition for the linear approximation R_i to be effective. The local flow in a linear region R_i is defined as a linear formula $f_i(\boldsymbol{x})$ of $\boldsymbol{x} \in \mathbb{R}^2$.

If the collection of linear regions covers $\mathbb{R}^2$ without overlap,[4] and the vector flows of adjacent linear regions are identical at both sides of the boundary, the solution is uniquely determined, the phase space is Euclidean (i.e., $\mathbb{R}^2$), and the flow is C^1 (i.e., the curve itself and its derivative are continuous). In other words, if the two conditions hold, whenever an orbit leaves (enters) an linear region, the next (previous) linear region is uniquely determined and transition at the boundary of linear regions is continuous.

Otherwise, the behavior may not be well-defined. The state transition may run into an undefined status, cause several variables to change discontinuously, or fail to uniquely specify the next state. Our algorithm presented in this paper can handle both cases.

A system of linear ODEs with two state variables dominating each linear region is written as:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = A\boldsymbol{x} + \boldsymbol{b}$$

where

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

As is obvious from the above, linear systems have only one fixed point at $-A^{-1}\boldsymbol{b} \in \mathbb{R}^2$ when $\det(A) \neq 0$[5]. By translating the coordinate to the fixed point by $\boldsymbol{v} \leftarrow \boldsymbol{x} + A^{-1}\boldsymbol{b}$, the above formula can be rewritten as

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = A\boldsymbol{v}.$$

The properties of linear flows in a two-dimensional phase space are simple. They can be classified into several categories by computing the eigenvalues and eigenvectors of the coefficient matrix A. In particular, we can grasp the asymptotic behavior of orbits from the eigenvalues and eigenvectors. When eigenvalues are real, the flow is usually divided into four independent subflows[6] by two *invariant manifolds*[7] spanned by the eigenvectors. When eigenvalues are complex (i.e., $a \pm \mathrm{i} \cdot b$, where $b \neq 0$), the flow can be categorized into subclasses depending on the sign of a. The flow is called a *spiral sink* if a is negative, a *center* if a is zero, and a *spiral source* if a is positive.

[4] Boundaries of linear regions are allowed to overlap.

[5] Since this is almost always the case, we assume this throughout this paper.

[6] As long as the two eigenvalues are different from each other.

[7] One-dimensional region (i.e., straight line) spanned by eigenvectors originated at the fixed point. Orbits on an invariant manifold are isolated from orbits elsewhere; orbits on an invariant manifold never leave the manifold and orbits elsewhere never come in the manifold.

Orbits of a linear flow do not have any point of inflection. Hence, each orbit other than a fixed point is either right-turning, left-turning, or straight. Furthermore, the way a given orbit turns can be determined by examining the sign of

$$\left[\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} \times \frac{\mathrm{d}^2\boldsymbol{x}}{\mathrm{d}t^2}\right]_z = \frac{\mathrm{d}x_1}{\mathrm{d}t} \cdot \frac{\mathrm{d}^2x_2}{\mathrm{d}t^2} - \frac{\mathrm{d}^2x_1}{\mathrm{d}t^2} \cdot \frac{\mathrm{d}x_2}{\mathrm{d}t}$$

at some point on it. Thus, the orbit is right-turning if the sign is negative; straight if the sign is zero; and left-turning if the sign is positive.

5. Properties of local flow

In this section, we study properties of local flow that two-dimensional linear flows make in convex regions of the phase space. Let us first introduce a few concepts to characterize local flow.

Definition 1. A convex region of a phase space is called a *cell*, if it is bounded by a polygon or a polyline, it has no fixed point or invariant manifold inside the cell, and it is dominated by a single linear flow.

Geometrically, a cell is divided into the internal region and the boundary. Each line segment comprising the boundary is called a *boundary edge*. Each couple of adjacent boundary edges is delimited by a vertex. A cell may be a finitely bounded region or an unbounded open region. Flow mappings are represented as inter-boundary edge mappings.

In order to represent the location of a point on the cell boundary, we introduce sampling lines defined as follows:

Definition 2. Sampling line l_k is represented as a pair $\langle \boldsymbol{b}, \boldsymbol{o}\rangle$, where $\boldsymbol{b} = (b_1, b_2) \in \mathbb{R}^2$, called the *base*, is a point on l_k, and $\boldsymbol{o} = (o_1, o_2) \in \mathbb{R}^2, |\mathbf{O}| = 1$ is the orientation of l_k.

By this convention, the location $\boldsymbol{x}$ of a point p on l_k can be represented as "$\boldsymbol{x} = \boldsymbol{b} + u \cdot \boldsymbol{O}$". We call this system of locating points *local coordinate* c_k associated with sampling line l_k, and we use a denotation "$loc(p) = u$ with respect to c_k" which reads "the location of p is u with respect to c_k". Likewise we mean by "(u, v) with respect to c_k" an open interval $\overline{pq}$ bounded by two points p and q, such that $loc(p) = u$ with respect to c_k and $loc(q) = v$ with respect to c_k. If $loc(p) = u$ with respect to c_k, $loc(q) = v$ with respect to c_k and $u < v$ for two points p and q on l_k, we say that p is *smaller than* q with respect to c_k, and denote it as $p \prec_{c_k} q$.

A sampling line $l_k = \langle \boldsymbol{b}, \boldsymbol{o}\rangle$ partitions the phase space into two disjoint regions: one to the right of it, $\{x \mid D_{l_k}(\boldsymbol{x}) < 0\}$ and the other to the left of it, $\{x \mid D_{l_k}(\boldsymbol{x}) > 0\}$, where

$$\begin{aligned} D_{l_k}(\boldsymbol{x}) &= [\boldsymbol{o} \times (\boldsymbol{x} - \boldsymbol{b})]_z \\ &= o_1 \cdot (x_2 - b_2) - o_2 \cdot (x_1 - b_1). \end{aligned}$$

In general, points on a sampling line that delimits interval are called *landmarks*. It should be noted that the exact address of landmarks is not necessary for understanding the structure of the flow. Instead, only a total ordering of landmarks on a sampling line is needed. In two-dimensional phase spaces, mapping ϕ_0 is order preserving. In other words, the order of landmarks is preserved in the mapping from r_1 to r_2.

5.1. Flow on the boundary of cells

Let c be a cell with local flow specified by a linear differential equation $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = A\boldsymbol{x} + \boldsymbol{c}$, p be a point on the boundary of c that is not a vertex, $l_k : \langle \boldsymbol{b}, \boldsymbol{o} \rangle = \langle (b_1, b_2), (o_1, o_2) \rangle$ be a sampling line passing p, and c_k be a local coordinate associated with l_k. We can see how the flow in c intersects with l_k, by evaluating the z-component of the cross product $\boldsymbol{o} \times \mathrm{d}\boldsymbol{x}/\mathrm{d}t$:

$$\left[o \times \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} \right]_z = pu + q \tag{3}$$

where

$$p = o_1^2 \cdot a_{21} + (a_{22} - a_{11}) o_1 o_2 - a_{12} o_2^2,$$

$$q = (a_{21} b_1 + a_{22} b_2 + c_2) o_1 - (a_{11} b_1 + a_{12} b_2 + c_1) o_2.$$

We say that the flow is *transverse-right* (*transverse-left*) to the sampling line l if the sign of $[\boldsymbol{o} \times \mathrm{d}\boldsymbol{x}/\mathrm{d}t]_z$ is negative (positive). The flow is *inward* if the internal region of the cell is to the right of the sampling line and the flow is transverse-right to the sampling line or if the internal region of the cell is in the left side of the sampling line and the flow is transverse-left to the sampling line. Such a point on the boundary is called an *entrance* to the cell. A source on the boundary is also regarded as an entrance. Maximally continuous segments of a boundary at which orbits transversely enter the cell are called *entrance segments*. An entrance segment is a point on the boundary if it is a source; otherwise, it is an open polyline segment consisting of one or more boundary edge. Likewise we define an *exit* and an *exit segment* for outgoing flow. Exit segments include sinks located on the boundary as a special case. Inter-boundary segment mappings play a useful role as intermediate representation for generating flow mappings at the inter-boundary edge level.

A point on the boundary of a cell is called a *singular node* if the flow is tangent to the boundary at that point. A maximal aggregation of contiguous singular nodes is called a *singular segment*. Singular nodes and segments play a crucial role in understanding the structure of the flow in a cell. From the definition (Eq. (3)) there is only one singular point at $-q/p$ with respect to l_k if $p \neq 0$. Singular segments are either an isolated point, a closed interval, or two closed intervals sharing a saddle node, as shown in Fig. 6. In general, a singular segment is a polyline segment if the boundary edge is on an invariant manifold; otherwise it consists of a single isolated point. When a singular segment consists of a single point, we call it a *singular node*.

The orbit passing through a singular node lies at the same side of the boundary immediately before and after passing the singular segment, for orbits of linear flow

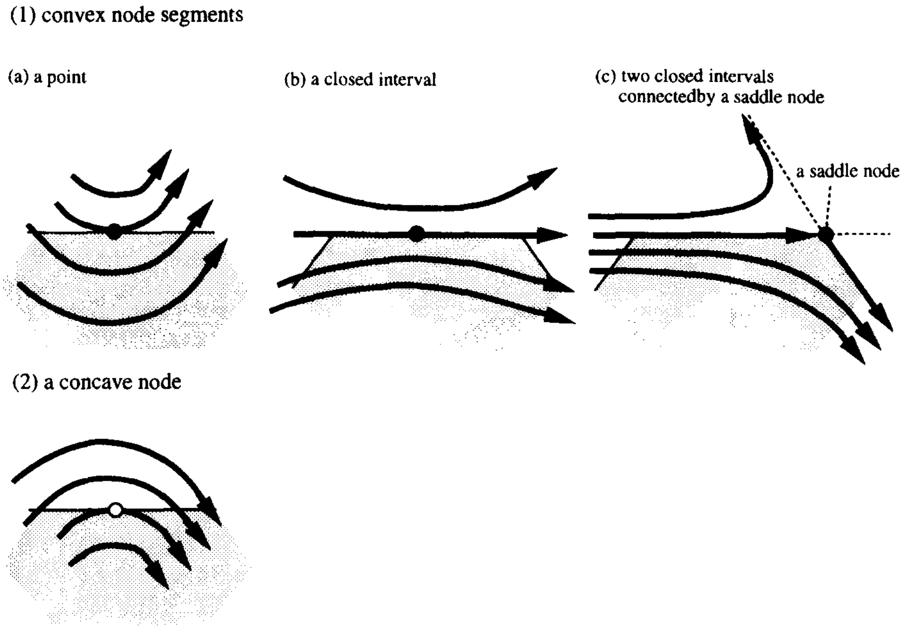

Fig. 6. Geometric configuration of singular segments.

Table 1
Rules for determining the type of a singular node—when the flow at the singular node is oriented in the same direction as that of the sampling line associated with the edge

The curvature of the orbit	The relative position of the cell with respect to the boundary	
	to the right	to the left
right-turning	concave node	convex node
left-turning	convex node	concave node

do not have any point of inflection. If the orbit lies outside the boundary immediately before and after passing the singular segment, we call the singular segment a *convex node segment*, or a convex node if it consists of a single point. Due to the properties of linear flows, a singular segment other than a convex node segment consist of a single point, which we call a *concave node*. By definition, an orbit passing on a concave node lies inside of the cell immediately before and after passing it.

The type of singular segment is determined by the curvature of orbit at the singular segment, the side in which the orbit lies immediately before and after the visit, and the relationship between the orientation of the flow at the tangent point and that of the sampling line associated with the edge. Table 1 shows a set of rules for determining the type of a singular node when the orientation of the flow conforms to that of the edge at the tangent point. A similar set of rules are defined for the other case, where the flow at a singular node is the opposite to the orientation of the edge. In what follows, entrance segments, exit segments and singular segments are generally referred to as *boundary segments*.

Since linear flows have strong constraints on the geometry of orbits, we can obtain useful clues about the flow of a cell, by a relatively simple computation. In particular, if a given cell has at most one concave node on the boundary, we can uniquely identify the flow pattern inside the cell. Even if there is more than one concave node on the boundary and unique interpretation is not possible, qualitative analysis enables us to generate a numerical computation plan for resolving ambiguities.

In the rest of this section, we first describe the properties of local flow in closed cells, and then extend the analysis to open cells, for flow patterns in closed cells are simpler than those in open cells.

5.2. *Properties of local flow in closed cells*

Orbits in closed cells come from entrance segments and tend towards exit segments. Let us consider a smooth closed curve C in a given cell such that C runs near the boundary of the cell but C does not pass on any fixed point. Let us also consider a point p walking around on C counterclockwise from some designated point p_0 on C. Then, the orientation of the flow θ changes continuously from some value θ_0 to $\theta_0 + 2\pi k$ ($k \in \mathbb{Z}$), where the integer k is called the *index* of C (with respect to the given flow).

The following fact is known about the index.

Proposition 3. ([2, p. 50]). *The index of C is zero if C does not contain any fixed point.*

Proposition 4. *Let the number of convex node segments and concave nodes be n_v and n_c, respectively. Then,*

$$n_v = n_c + 2. \tag{4}$$

Proof. Let us represent the location of point p on C by the length x ($0 \leqslant x < |C|$) of the counterclockwise pass from p_0, and think about $\phi(x)$, where ϕ stands for the orientation of the flow at p with respect to the orientation of the tangent vector of C at p. Then, by putting together Proposition 3 and the fact that the tangent vector of C at p will eventually increase by 2π as p walks around on C, we can say that $\phi(x)$ is a continuous function varying from $(0, \phi_0)$ to $(|C|, \phi_0 - 2\pi)$ in the x-ϕ space, as shown in Fig. 7.

In the x-ϕ plane, convex and concave nodes appear as points where $\phi(x)$ cuts across each horizontal line $\phi = n\pi$ ($n \in \mathbb{Z}$) as shown in Fig. 8, and normally there is exactly one integer k such that horizontal lines $\phi = k\pi$ and $\phi = (k+1)\pi$ intersect with $\phi(x)$ in $0 \leqslant x < |C|$. □

Proposition 5. *The maximal difference in the orientation of flow in a cell is at most π.*

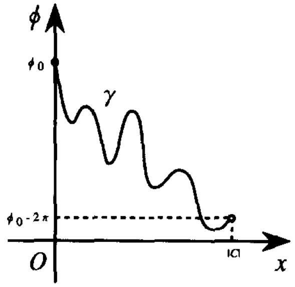

Fig. 7. An x-ϕ curve.

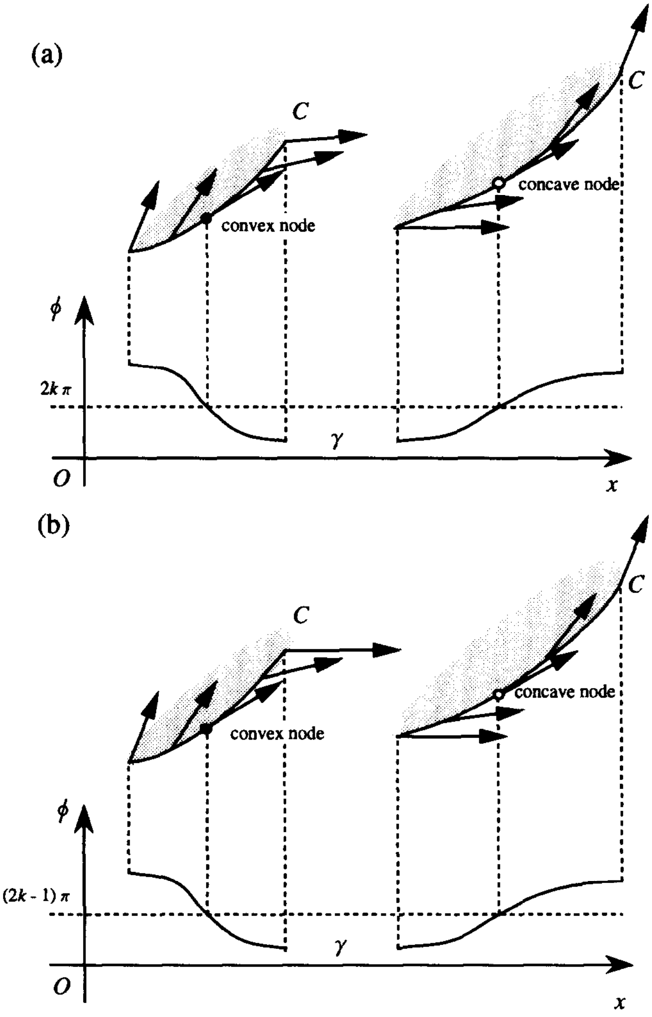

Fig. 8. Appearance of concave and convex nodes in the x-ϕ plane.

Proof. The proposition follows from our phase space partitioning policy that each cell is convex and all eigenspaces are used for partitioning the phase space. □

Definition 6. A *boundary segment list* for a cell is a sequence of singular nodes on the boundary, collected counterclockwise from some point on the boundary.

Note that there could be more than one boundary segment list for a cell, depending on where to start.[8]

Proposition 7. *No boundary segment list of a cell contains two consecutive concave nodes.*

Proof. Assume that concave nodes n_a and n_b are consecutive in a boundary segment list. Consider a closed convex loop C which is included in a given cell and is close enough to the boundary. Then, it follows from the assumption that there exists an integer k and a pair x_a and x_b such that $x_a < x_b$, $\phi(x_a) = k\pi$, $\phi(x_b) = k+1$, and $\phi(x)$ is increasing at a neighborhood of x_a and x_b. Let x_1 and x_2 be two values outside $[x_a, x_b]$ such that $x_1 < x_a$, $x_b < x_2$, $\phi(x_1) < \phi(x_a)$, $\phi(x_b) < \phi(x_2)$, and let (θ_1, ψ_1) and (θ_2, ψ_2) be pairs of the orientations of the flow and the boundary edge at points at x_1 and x_2, respectively. As $\phi_1 = \theta_1 - \psi_1$ and $\phi_2 = \theta_2 - \psi_2$, we have

$$\phi_2 > \phi_b = \phi_a + \pi > \phi_1 + \pi.$$

We also have

$$\psi_1 < \psi_2,$$

since C is convex. From the above, we have

$$\theta_2 = \psi_2 + \phi_2 > \psi_1 + \phi_1 + \pi = \theta_1 + \pi.$$

This contradicts Proposition 5. □

Proposition 8. *No boundary segment list for a cell contains a sequence*

$$\mathrm{C} \cdot \mathrm{V} \cdot \mathrm{V} \cdot \mathrm{C},$$

where C *and* V *stand for a concave node and convex node segment, respectively.*

Proof. The ϕ values of two consecutive nodes of the same type with respect to convexity differ by π. Hence, if there existed a pair of concave nodes as mentioned in the statement above, the ϕ values for the two concave nodes should differ by π, which means that both right-turning and left-turning orbits are contained in a single cell, resulting in a contradiction with the way cells are built. □

[8] We could avoid this problem by defining it as a circular list. However, we will fix the problem by introducing a notion of "normal" boundary segment list, as given in Definition 10.

It is noted that two consecutive convex nodes do not cause a contradiction, for convex nodes may be located on corners of the boundary and we cannot tell the orientation of the flow there.

Theorem 9. *For any cell, there exists a boundary segment list representation of the form*

$$\mathrm{V}\cdot\mathrm{V}\cdot(\mathrm{V}\cdot\mathrm{C})^*,$$

where V *and* C *stand for a convex node segment and a concave node, respectively.*

Proof. From Propositions 4, 7, and 8, it is clear that the only way of allocating convex and concave nodes on the boundary segment list is exactly as described in the statement of this theorem. □

In particular, when a concave node is involved in a boundary segment list, there exists a sequence of three consecutive convex node segments that have no concave node in-between. The center of such convex node segments is called the *center segment.* When no concave nodes are involved, either one of the two convex node segments is taken as the center segment.

Definition 10. A sequence of all boundary segments ordered clockwise from the center segment is called a *counterclockwise boundary segment list.*

A boundary segment s of a cell is said to be *to the right (left) of boundary segment p* of the same cell if s appears after (before) p in the counterclockwise boundary segment list. An orbit is said to *pass through a cell from left to right* if the boundary segment through which the orbit leaves the cell is to the right of the boundary segment through which the orbit enters the cell. Orbits that pass through a cell from right to left are similarly defined. The internal flow of a cell is either:

- all orbits pass through the cell from left to right (left–right flow); or
- all orbits pass through the cell from right to left (right–left flow).

It is easy to see whether a local flow is left–right or *vice versa.* For example, one might see the orientation of the flow at the boundary segment immediately to the right of the center segment.

If a concave node c is involved in the boundary, then the orbit passing on c intersects with each of the two sequences of the boundary segments between c and the center segment.

By taking into account the above propositions and the non-intersection constraint of orbits, we have the following proposition:

Proposition 11. *All orbits inside a cell are nested around the center segment.*

Now let us give a couple of examples. Suppose that we are given a flow pattern on the boundary of a cell, as shown in Fig. 9. Since no concave node segments are involved in the boundary, the local flow inside this cell is uniquely characterized as inter-boundary

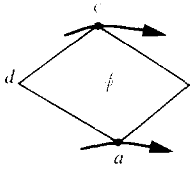

Fig. 9. Example of local flow in a closed cell (1).

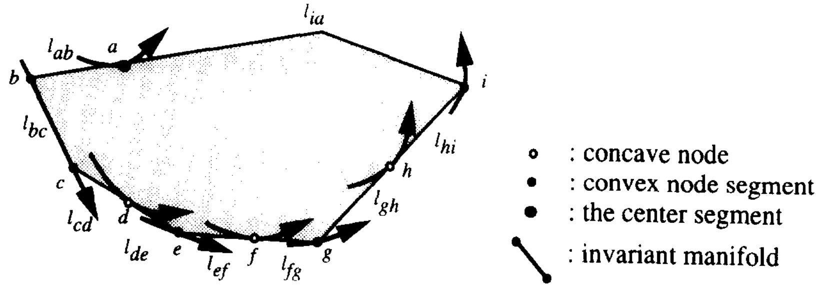

Fig. 10. Example of local flow in a closed cell (2).

segment mapping: $\overline{c\,d\,a} \to \overline{a\,b\,c}$. In order to characterize the flow as inter-boundary edge mappings, three possibilities arise:

(1) $\overline{d\,a} \to \overline{a\,\phi(d)}$, $\overline{\phi^{-1}(b)\,d} \to \overline{\phi(d)\,b}$, $\overline{c\,\phi^{-1}(b)} \to \overline{b\,c}$;

(2) $\overline{d\,a} \to \overline{a\,b}$, $\overline{c\,d} \to \overline{b\,c}$;

(3) $\overline{\phi^{-1}(b)\,a} \to \overline{a\,b}$, $\overline{d\,\phi^{-1}(b)} \to \overline{b\,\phi(d)}$, $\overline{c\,d} \to \overline{\phi(d)\,c}$.

where $\phi(d)$ is an image of d by ϕ, and $\phi^{-1}(b)$ is a point on the boundary that is mapped to b by ϕ. We use numerical computation to resolve this kind of ambiguities. For instance, we can track orbits passing on vertices and see which boundary element they intersect.

Suppose now that we are given a flow pattern on the boundary of a closed cell, as shown in Fig. 10. This case contains three concave nodes on the boundary, and we can think of eleven possible ways of qualitatively different patterns of local flow at the inter-boundary segment level, as shown in Fig. 11. Much more possibilities arise if we consider the case at the inter-boundary edge level.

Let us determine how many qualitatively different patterns of flow we have at the inter-boundary segment mapping level. Let $P_C(n)$ be the number of qualitatively different patterns of flow in a cell with n concave nodes. First, let us consider the case in which no single orbit passes more than one concave node and let the number of possibilities in that case be $P_C^N(n)$. $P_C^N(n)$ is recursively defined as follows:

$$P_C^N(0) = 1;$$

$$P_C^N(1) = 1;$$

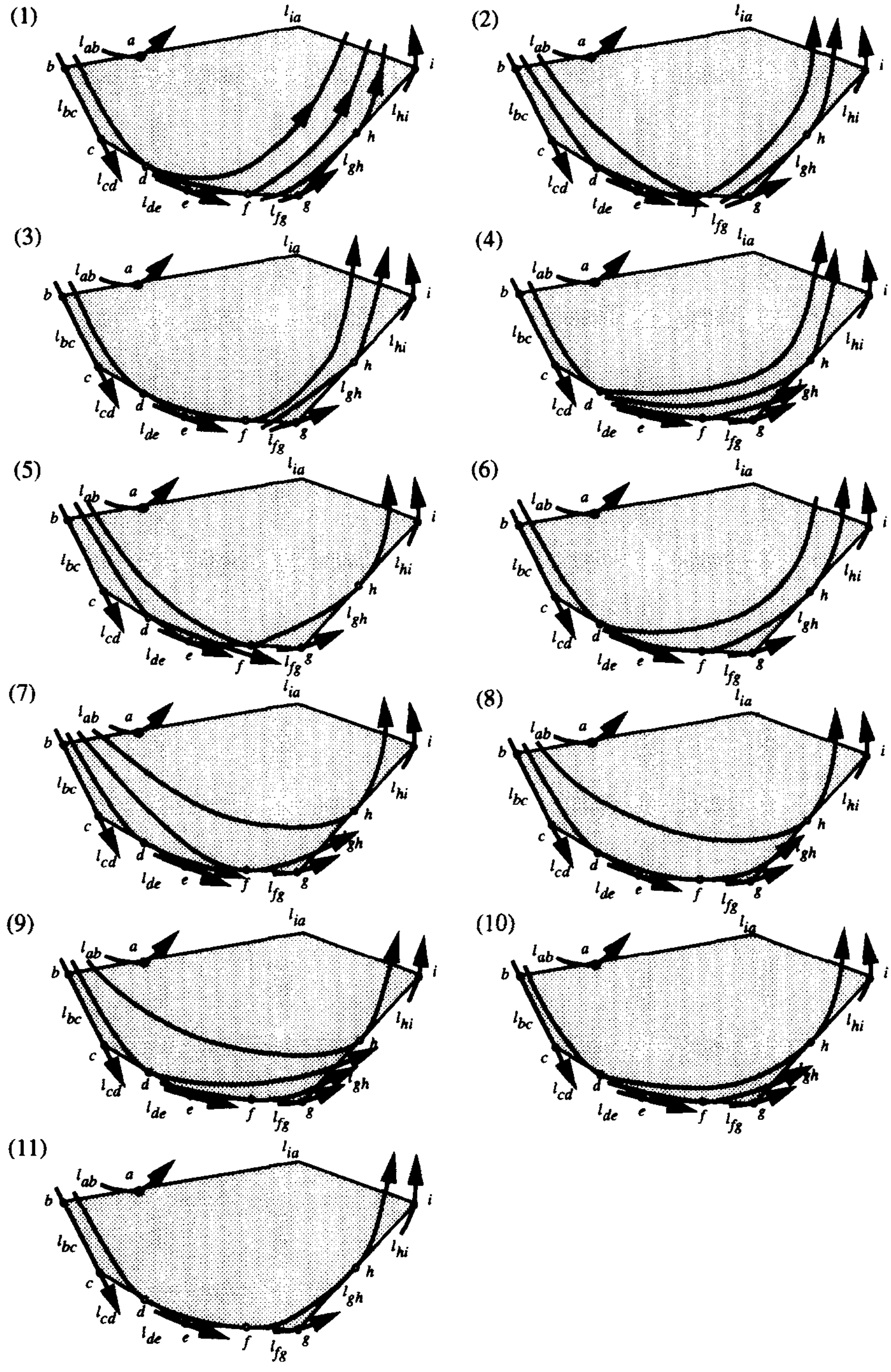

Fig. 11. Eleven possible ways of qualitatively different patterns of local flow for the cell shown in Fig. 10.

$$P_C^N(n) = \sum_{i=1}^{n} P_C^N(i-1) P_C^N(n-i) \quad (n \geqslant 2).$$

The values of $P_C^N(n)$ for small n are as follows:

$$P_C^N(2) = 2, \quad P_C^N(3) = 5, \quad P_C^N(4) = 14,$$
$$P_C^N(5) = 42, \quad P_C^N(6) = 132, \quad P_C^N(7) = 429, \ldots .$$

Using $P_C^N(n)$ as defined above, $P_C(n)$ is defined as follows:

$$P_C(0) = 1,$$
$$P_C(1) = 1,$$
$$\begin{aligned} P_C(n) = \sum_{p=1}^{n-1} \sum_{1 \leqslant i_1 < i_2 < \cdots < i_p \leqslant n-1} & \left(\sum_{k=0}^{i_1 - 1} P_C(k) P_C(i_1 - k - 1) \right) \\ & \times \left(\prod_{j=1}^{p-1} P_C(i_{j+1} - i_j - 1) \right) \\ & \times P_C(n - i_p - 1) \\ + \sum_{i=0}^{n-1} P_C(i) P_C(n-i-1) & \quad \text{where } n \geqslant 2. \end{aligned}$$

The values of $P_C(n)$ for small n are:

$$P_C(2) = 3, \quad P_C(3) = 11, \quad P_C(4) = 45,$$
$$P_C(5) = 197, \quad P_C(6) = 903, \quad P_C(7) = 4279, \ldots .$$

5.3. Properties of local flow in open cells

Local flow in open cells are slightly harder to capture. In this paper, we reduce the problem of analyzing local flow in open cells to that of closed cells. The idea is to extend a boundary segment list with additional virtual boundary entities such as *point at infinity* or an *edge at infinity* and to give them appropriate attributes so that the following condition may be satisfied:

(1) the number of convex segment nodes is that of concave nodes plus *two*,

(2) there is at least one entrance segment and exit segment in the extended boundary segment list, and

(3) flow at both sides of every singular segment is in the opposite direction; if flow is inward at one side, flow should be outward at the other side.

Before describing general decision rules, let us study several examples. First, consider an open cell shown at the left side of Fig. 12. This cell consists of three real boundary edges. Two of them are part of invariant manifolds. The remaining one is an exit

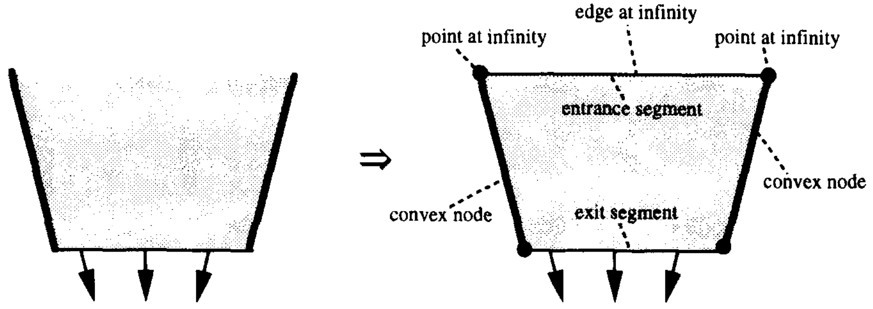

Fig. 12. Example of local flow in an open cell (1).

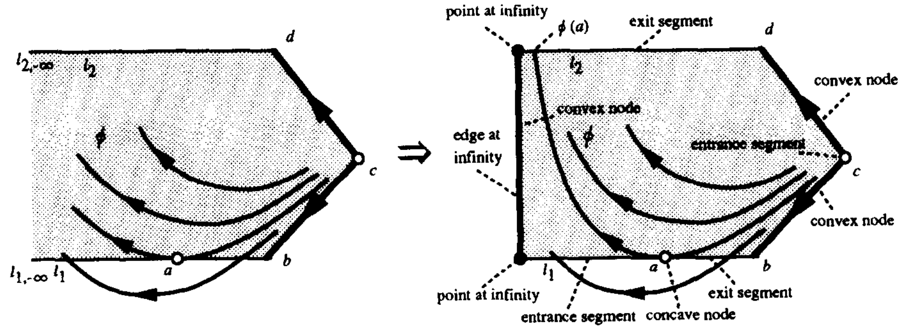

Fig. 13. Example of local flow in an open cell (2).

segment. Three additional virtual boundary entities, two points at infinity and an edge at infinity, are introduced. The constraint on the number of concave nodes and convex node segments is already satisfied. But as there is no entrance segment, we regard the edge at infinity as an entrance segment. This interpretation conforms to the fact that all of the orbits crossing the real exit segment diverge to the direction delimited by the two invariant manifolds as $t \to -\infty$.

Consider the local flow in an open cell shown in Fig. 13, which has a concave node a on the boundary. Although both entrance segments and exit segments exist on the boundary, the number of convex node segments is one less than what is required. We regard the edge at infinity a convex node segment, and characterize the local flow in this cell as the following set of mappings:

$$\overline{l_{1,-\infty}\,a} \to \overline{l_{2,-\infty}\,\phi(a)}, \qquad c \to \overline{a\,b}, \qquad c \to \overline{\phi(a)\,d},$$

at the inter-boundary segment level.

Consider the flow in an open cell shown at the left side of Fig. 14. Both of the two boundary edges comprising the boundary of the cell are portions of invariant manifolds connected by a saddle node. All of these boundary entities are viewed as one convex node segment. There is no entrance segment nor an exit segment on the boundary. Three additional virtual boundary entities are introduced. In order to satisfy the constraints on boundary entities, we divide the edge at infinity into three segments: an entrance segment,

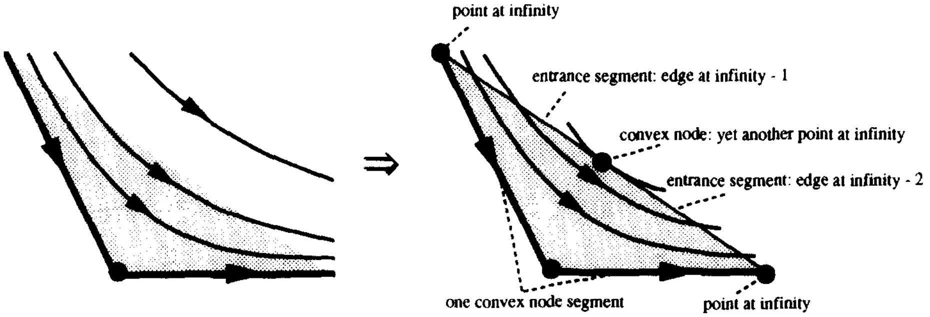
point at infinity
entrance segment: edge at infinity - 1
convex node: yet another point at infinity
entrance segment: edge at infinity - 2
one convex node segment
point at infinity

Fig. 14. Example of local flow in open cell (3).

an exit segment, and a convex node between the two.

Fig. 15 shows general rules for assigning attributes to the edge at infinity e_∞ and two points of infinity $p_{\infty,1}$ and $p_{\infty,2}$.

Attribute assignments resulting from these rules have the following properties:

- When the edge of infinity is regarded as an entrance segment, there are orbits which diverge to the corresponding direction as $t \to -\infty$.
- When the edge of infinity is regarded as an exit segment, there are orbits which diverge to the corresponding direction as $t \to \infty$.
- When the edge of infinity is regarded as a singular segment, no orbit diverges to the corresponding direction as $t \to \pm\infty$.

6. The algorithm

We have implemented a program PSX2PWL based on the ideas presented above. PSX2PWL takes a specification of a system of piecewise linear ODEs of the form (2) and produces (a) a list of g-sources and g-sinks together with the type and geometric features, and (b) a set of transitions from g-sources to g-sinks.

The top level algorithm of PSX2PWL is shown in Fig. 16. In the beginning, PSX2PWL partitions the phase space into a collection of cells, by boundaries between linear regions and invariant manifolds of fixed points with real eigenvalues. The rest of the processing is divided into local analysis and global analysis. In local analysis, PSX2PWL analyzes the local flow in each cell, identifies g-sinks and g-sources in each cell, and characterizes the flow in each cell as a collection of flow mappings. In global analysis, PSX2PWL puts together results obtained from local analysis of each cell, and examines the properties of flow mappings as a whole, by computing possible compositions of flow mappings.

[9] The result does not depend on which one of the two points of infinity is chosen.

(1) If $n_v = n_c + 2$, then:
   (a) If the open cell has no entrance segment, then e_∞ is an entrance segment;
   (b) if the open cell has no exit segment, then e_∞ is an exit segment;
   (c) otherwise, e_∞ is a convex node segment or its portion.

(2) If $n_v = n_c + 1$, then:
   (a) If the cell has both an entrance segment and an exit segment, then e_∞ is a convex node segment.
   (b) If the cell has an entrance segment but no exit segment, then e_∞ is an exit segment.
   (c) Similarly, if the cell has an exit segment but no entrance segment, then e_∞ is an entrance segment.
   (d) Otherwise (i.e., the cell has no entrance segment nor exit segment), then divide e_∞ into two by introducing a new virtual vertex and regard the two subsegments resulting from subdivision as an entrance segment and an exit segment, respectively. The sign of eigenvalue associated with the adjacent invariant manifold is examined to determine the type of the two boundary segments: the boundary segment adjacent to an unstable manifold (an invariant manifold associated with a positive eigenvalue) is regarded as an exit segment, and the one adjacent to the stable manifold (an invariant manifold associate with a negative eigenvalue) is regarded as an entrance segment.

(3) If $n_v = n_c$, then:
   (a) If the cell has an entrance segment but no exit segment, then e_∞ is an exit segment.
   (b) Similarly, if the cell has an exit segment but no entrance segment, then e_∞ is an entrance segment.
   (c) Otherwise, regard $p_{\infty,1}$ and $p_{\infty,2}$ as convex nodes, and regard e_∞ as an entrance segment (exit) if the boundary segment on the other side of a point at infinity is an exit (entrance) segment.[9]

The type of points of infinity p_∞, if not determined in the above, is determined based on the type of adjacent boundary segments, as follows:

(1) If the type of the two adjacent boundary segments of p_∞ are the same, so is the type of p_∞.
(2) If one of the adjacent boundary segments of p_∞ is a singular segment, so is p_∞.
(3) Otherwise, p_∞ is a convex node segment or its portion.

where n_c and n_v are the numbers of "real" concave nodes and convex node segments on the boundary, respectively.

Fig. 15. General rules for assigning attributes to virtual boundary segment list items.

7. Complete example

We have implemented the idea as a program called PSX2PWL. In this section, we use an example to illustrate how PSX2PWL works. For a more detailed description of the algorithm, the reader is referred to [4].

Consider an unstable multivibrator shown in Fig. 17. By modeling the two transistors as ideal switching elements, we obtain a set of circuit equations:

```
procedure PSX2PWL($s$: system of PWL-ODEs):
  begin
    partition the phase space into cells ($C = \{c_i\}$);
    ; local analysis
    for each $c_i \in C$
       begin
         instantiate boundary segments;
         generate inter-boundary segment mappings;
         generate inter-boundary edge mappings;
         generate local flow mappings $m_i$ for $c_i$;
       end
    ; global analysis
    let $M = \bigcup_{c_i \in C} m_i$;
    examine the properties of $M$
     by computing possible compositions
     $m_{i_1} \circ \cdots \circ m_{i_j}$ of flow mappings $m_{i_k} \in M$;
  end
```

Fig. 16. The top level algorithm of PSX2PWL.

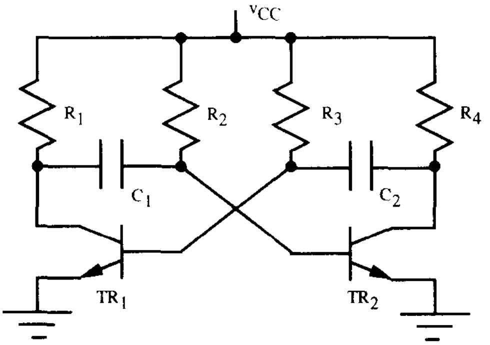

Fig. 17. Unstable multivibrator.

$$\left\{ \begin{array}{l} R_{1,0} : \left\langle x \leqslant v_\sigma \mid \dfrac{\mathrm{d}x}{\mathrm{d}t} = \dfrac{v_{CC} - x}{C_1 R_2}, \dfrac{\mathrm{d}y}{\mathrm{d}t} = \dfrac{-v_{CC} + v_\sigma - y}{C_2 R_4} \right\rangle, \\ R_{0,1} : \left\langle y \leqslant v_\sigma \mid \dfrac{\mathrm{d}x}{\mathrm{d}t} = \dfrac{-v_{CC} + v_\sigma - x}{C_1 R_1}, \dfrac{\mathrm{d}y}{\mathrm{d}t} = \dfrac{v_{CC} - y}{C_2 R_3} \right\rangle, \\ \quad \dots \end{array} \right\}, \qquad (5)$$

where variables x and y denote voltage across capacitors C_1 and C_2, respectively. As for other parameters,

$$v_\sigma = 0.7, \qquad v_{CC} = 5, \qquad \{C_1, C_2, R_1, R_2, R_3, R_4\} > 0.$$

For simplicity, let us assume that we have only two linear regions: $R_{1,0}$ and $R_{0,1}$. The former stands for a state in which TR_1 is ON and TR_2 is OFF, while the latter stands for the state in which the activation pattern of the two transistors are exactly the opposite

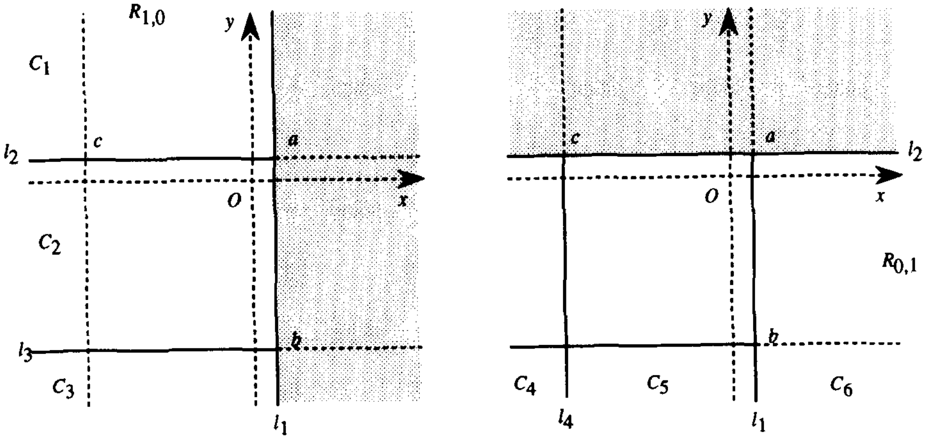

Fig. 18. Partitioning the phase space for (5) into cells.

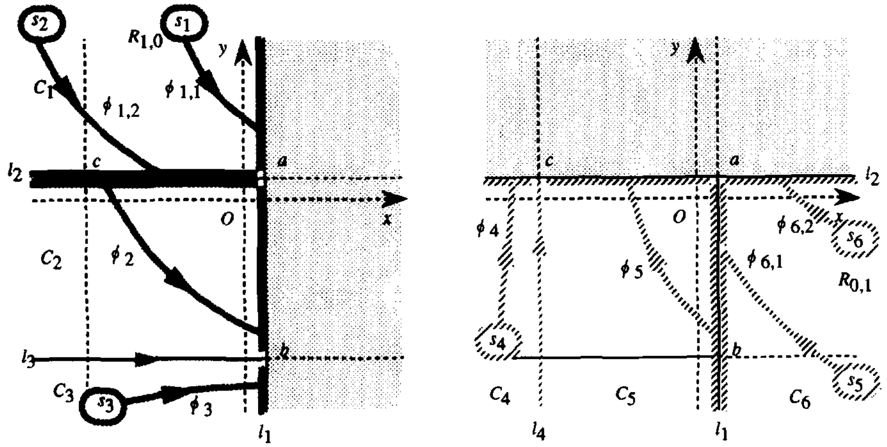

Fig. 19. Local analysis that PSX2PWL has produced for (5).

of that in $R_{1,0}$. Note that these two linear regions overlap with each other and hence the uniqueness of the solution does not hold at boundaries.

PSX2PWL first partitions linear regions $R_{1,0}$ and $R_{0,1}$ into cells, as shown in Fig. 18. Local analysis identifies local flow in each cell, as shown in Fig. 19. By global analysis, PSX2PWL finds the following contracting recursive mapping:

$$\phi_{5,2} \circ \phi_2 : \overline{l_{2,-\infty}\, a} \rightarrow \overline{\phi_5(b)\, a} \subset \overline{l_{2,-\infty}\, a}.$$

This convinces PSX2PWL of the existence of an attractor (in weaker sense) that is transverse to $\overline{\phi_5(b)\, a}$, where

$$l_{2,-\infty} \prec_{c_2} c \prec_{c_2} \phi_5(b) \prec_{c_2} a \prec_{c_2} l_{2,\infty}.$$

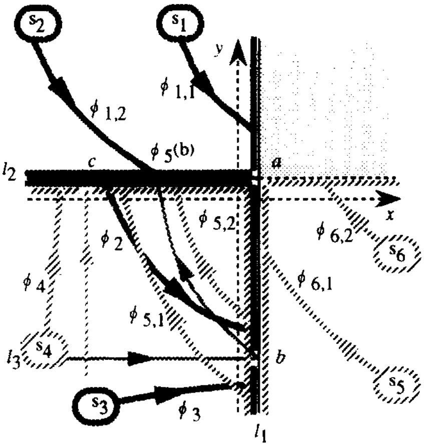

Fig. 20. Global analysis that PSX2PWL has produced for (5).

Furthermore, PSX finds out that all orbits except those originated from sources S_1 and S_6 that are located in place at infinity will eventually be absorbed into the attractor found above. Fig. 20 illustrates global analysis by PSX2PWL. The analysis correctly captures the qualitative behavior of the unstable multivibrator.

8. Related work

In general, there are two extremes in the spectrum of approaches to qualitative phase space analysis. One extreme is to heavily rely on numerical computation: running a numerical simulator and interpreting the result in an intelligent way. One can constrain search by incorporating general knowledge about dynamical systems. One can also use techniques from computer vision and computational geometry to interpret phase portraits partly drawn by a numerical simulator. Yip [7] took this approach to analyze discrete systems. Sacks [6] presented a method for analyzing continuous systems. Abelson [1] provided a global picture of activities at MIT.

The other extreme is to rely more on qualitative methods. By intensive use of knowledge about dynamical systems, we can strongly guide the search process and the result is less likely to be affected by numerical errors. It is possible to avoid combinatorial explosion, by invoking a numerical computation as soon as ambiguity is detected. The approach we have taken in this paper is closer to the second extreme.

Sacks [5] took a similar approach, too. Our approach has two advantages over Sacks' POINCARE. First, PSX2PWL partitions the phase space into linear regions in a more general way than POINCARE. Although POINCARE partitions it only by straight lines that are parallel to the x- or y- axis, PSX2PWL partitions the phase space by any polyline, allowing to partition the phase space by eigenspaces.

The second and more essential difference is the internal representation used for analysis. Sacks used transition graphs as internal representation. With transition graphs alone,

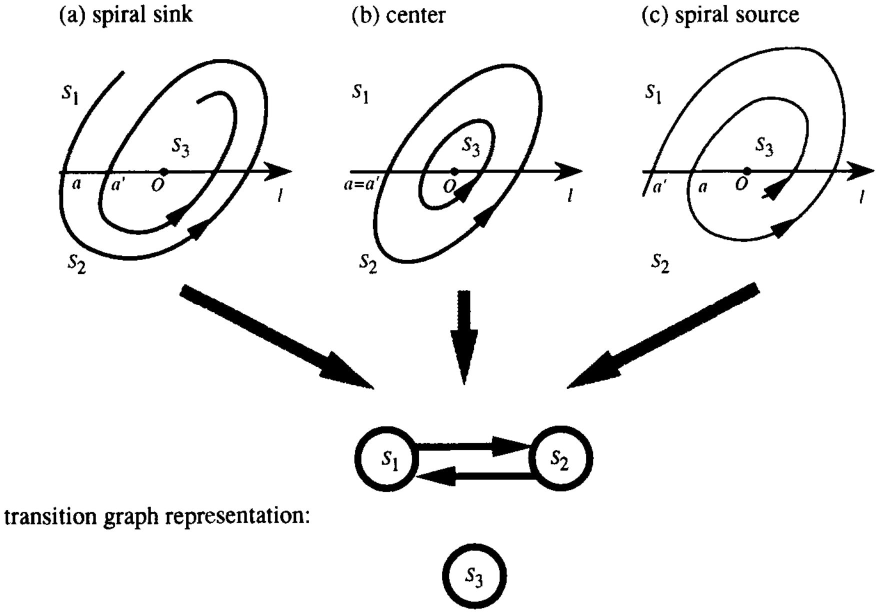

Fig. 21. Transition graph representation of spiral sink, spiral source, and center.

(a) f such that $f(\overline{Oa}) \subset \overline{Oa}$.
(b) f such that $f(\overline{Oa}) = \overline{Oa}$.
(c) f such that $f(\overline{Oa}) \supset \overline{Oa}$.

Fig. 22. Flow map representation of spiral sink, spiral source, and center.

it is impossible to make the important distinction between spiral sinks, spiral sources, and centers, as shown in Fig. 21. In contrast, it is possible to distinguish the three by the mapping representation employed in this paper, as shown in Fig. 22 (a) through (c).

Unfortunately, what has been done is severely limited in the number of the dimensionality of phase spaces. The bottleneck is the algorithm for local analysis. An interesting open question is how to develop a general algorithm for generating flow mappings for n-dimensional flow.[10]

9. Concluding remark

In this paper, we have presented a program PSX2PWL that explores phase portraits of systems of PWL-ODEs with two state variables. The program is based on four simple ideas: reasoning about a bundle of orbits, representing a bundle of orbits as a flow mapping between hyperplanes, deriving global behavior by analyzing the structure of

[10] The first author has developed an algorithm for three-dimensional piecewise linear flow. It is still limited, but a big progress because the current version can analyze a system of ODEs that manifest chaotic behavior.

flow mappings, and taking advantage of strong constraints of two-dimensional piecewise linear flow. We have given a detailed analysis of local flow in a cell, which brings about insights in designing the algorithm. We have implemented the ideas as a program called PSX2PWL and tested it against several examples.

Acknowledgment

We would like to thank the late Mr. Yoichi Shimada for discussion. We also thank Mr. Kenji Satoh for writing a program which counts the number of ambiguities based on the formula given in this paper.

References

[1] H. Abelson, M. Eisenberg, M. Halfant, J. Katzenelson, E. Sacks, G.J. Sussman, J. Wisdom and K. Yip, Intelligence in scientific computing, *Comm. ACM* **32** (1989) 546–562.

[2] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer-Verlag, Berlin, 1983).

[3] M.W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra* (Academic Press, New York, 1974).

[4] T. Nishida and S. Doshita, PSX: a program that explores phase portraits of two-dimensional piecewise linear differential equations, *Memoirs of the Faculty of Engineering, Kyoto University* **52** (4) (1990) 311–355.

[5] E. Sacks, Automatic qualitative analysis of ordinary differential equations using piecewise linear approximations, Tech. Report 1031, MIT Artificial Intelligence Laboratory (1988).

[6] E. Sacks, Qualitative analysis of continuous dynamic systems by intelligent numeric experimentation, in: *Third International Workshop on Qualitative Physics*, Stanford, CA (1989).

[7] K. M.-k. Yip, Generating global behaviors using deep knowledge of local dynamics, in: *Proceedings AAAI-88*, St. Paul, MN (1988) 280–285.

Postscript

This translation represents work that was accomplished by June 1990. After that, several versions of PSXs have been constructed and much progress has been made. (Nishida, 1991a) describes automated analysis of general nonlinear ODEs with two state variables. (Nishida, 1991b) introduced a notion of flow grammar to reason about topology of phase portraits. These ideas were incorporated into PSX2NL. (Nishida, 1993a) gives a general framework of the approach. (Nishida, 1993b) is an extension in the number of dimensionality from two to three, which enables to reason about chaotic behaviors as reported in (Nishida, 1994). The ideas were incorporated into a program called PSX3. Extension into n-dimensional phase space still is a big challenge to us, though.

References

(Nishida, 1991a) T. Nishida, K. Mizutani, A. Kubota and S. Doshita, Automated phase portrait analysis by integrating qualitative and quantitative analysis, in: *Proceedings AAAI-91*, Anaheim, CA (1991) 811–816.

(Nishida, 1991b) T. Nishida and S. Doshita, A geometric approach to total envisioning, in: *Proceedings IJCAI-91*, Sydney, NSW (1991) 1150-1155.

(Nishida, 1993a) T. Nishida, Generating quasi-symbolic representation of three-dimensional flow, in: *Proceedings AAAI-93*, Washington, DC (1993) 554-559.

(Nishida, 1993b) T. Nishida, K. Mizutani and Shuji Doshita, Automated analysis of qualitative behaviors of piecewise linear ordinary differential equations, *New Gen. Comput.* **11** (2) (1993) 159-177.

(Nishida, 1994) T. Nishida, Qualitative reasoning for automated exploration for chaos, in: *Proceedings AAAI-94*, Seattle, WA (1994) 1211-1216.