A Geometric Approach to Total Envisioning

Toyoaki Nishida and Shuji Doshita
Department of Information Science
Kyoto University
Sakyo-ku, Kyoto 606. Japan
email: nishid ay kuis. k yo to-u. ac. jp

Abstract

Conventional envisioners proposed in qualitative physics have two difficulties in common: ambiguities in prediction and inability of reasoning about global behaviors. We take a geometric approach to overcome these difficulties and have implemented a program PSX2NL which can reason about global behaviors by analyzing geometry and topology of solution curves of ordinary differential equations in the phase space.

In this paper, we highlight a *flow grammar* which specifies possible patterns of solution curves one may see in the phase space. The role of a flow grammar in PSX2NL is twofold: firstly, it allows PSX2NL to reason about complex patterns in a uniform manner; secondly, it allows PSX2NL to switch to an approximate, top-down algorithm when complete geometric clues are not available due to the difficulty of mathematical problems encountered.

1 Introduction

One of the core issues in qualitative physics is to develop an envisioner for deriving the behavior from the structure of given dynamical systems. The intended advantage of total envisioners (hereafter, simply "envisioners") over numerical simulators is an ability of automatically deriving an abstract, qualitative description of behaviors of a given dynamical system under various initial conditions. Constraint propagation and satisfaction on symbolically represented quantity space have been a common technique of implementing envisioners [Weld and de Kleer, 1989].

However, as is often pointed out, envisioners based on constraint propagation and satisfaction have two problems: ambiguity in prediction and inadequate global analysis. Both of these problems are attributed to the local nature of constraint propagation and satisfaction.

A promising direction would be to take a geometric approach. In fact, mathematicians have long been using geometric methods to study complex behaviors of nonlinear differential equations. Why not build an envisioner on a firm ground?

In this paper and related work [Nishida and Doshita, 1990; Nishida *d ai.* 1991], we propose a geometric method of reasoning about *phase portraits,* collections of all solution curves of ordinary differential equations in the phase space. The outline of the method is this:

- collect geometric features of solution curves using varieties of quantitative techniques,
- infer topology of the phase portrait from geometric cues, and
- reason about the global behavior by analyzing topology of the phase portrait.

This method solves the problems raised above in the following way: firstly, the problem of ambiguity is much reduced because the geometry and topology of phase portraits are determined based on quantitative information; and secondly, global analysis based on geometric and topological analysis of phase portraits is theoretically supported by dynamical systems theories [Hirsch and Smale, 1974; Cuckenheimer and Holmes, 1983], Yet, the method satisfies a general requirement to envisioners, for it produces symbolic description of total behavior. The method is incorporated in a program PSX2NL which works on ordinary differential equations defined on a two-dimensional phase space.

In developing PSX2NL, we have argued the importance of representation and accordingly we have introduced *flow patterns* to represent the geometric and topological features of phase portraits. Notion of flow patterns, algorithm for deriving flow patterns, and algorithm for reasoning about global behavior are presented elsewhere [Nishida and Doshita, 1990; Nishida *et ai,* 199l].

In this paper, we highlight a *flow grammar,* which specifies all possible flow patterns one may see in the phase space. The role of a flow grammar in PSX2NL is twofold: firstly, it allows PSX2NL to reason about complex patterns in a uniform and systematic manner; secondly, it allows PSX2NL to switch to an approximate, top-down algorithm when complete geometric clues are not available dne to the difficulty of mathematical problems encountered.

We begin with the background of this work, introducing mathematical notions and flow patterns. Next, we describe a flow grammar and related issues in some detail. Finally, we compare our work with related work and show the future direction.

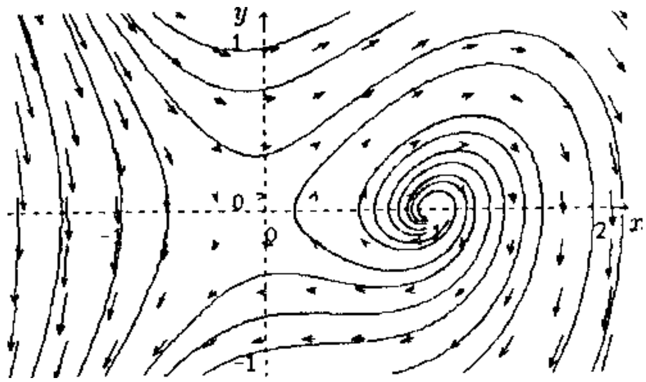


Figure 1: the Vector Field (arrows) and Phase* Portraits (curves) of (2) (partly drawn)

2 Qualitative Theory of Differential Equations

In this paper, we consider planar ordinary differential equations (planar ODEs for short)

$$\begin{cases} \frac{dx}{dt} = f(x, y) \\ \frac{dy}{dt} = g(x, y) \end{cases} \quad (1)$$

on *stale variables x(t)* and y(+) that vary with time The right-hand side of (1) specifies the velocity of state change at each point in the phase space. In other words, formula (1) introduces a *rt ctor field* in the planar *phast space* spanned by *x* and *y.* In normal situations, a vector field implicitly specifies a *phase port mils,* a collection of non-intersecting directed curves such that each directed curve is tangent to the vector field. Each directed curve- is called an *orbit or* a *solution caret* or a *trajeitory.* and corresponds to a solution under a certain initial condi tion.

For example, figure 1 shows (a) the vector field and (b*)* the p 11 a se portrait of an ODE:

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = x - x^2 + 0.2y + 0.3xy. \end{cases} \quad (2)$$

State change of a dynamical system occurs along with orbits. To put it another way, each orbit can be identified with a function from the phase space to the phase space. In this sense, the phase portrait is said to define a *flow* in the phase space. Dynamical systems theories suggest that understanding the behavior begins with qualitative analysis by identifying regions in the phase space which orbits approach as $t \rightarrow \pm\infty$, and classifying orbits by the regions they tend towards (or by *asymptotic behavior).* As for two-dimensional planar ODEs, it is proved that orbits may either diverge to place at infinity, or tend towards a *fixed point* (an orbit consisting of a point which makes the right-hand side of (1) *zero)* or a *limit cycle* (a cyclic orbit which attracts or repells nearby orbits).[1] Methods of dynamical systems theories are powerful enough to provide useful information

*It follows from Poincare-Bendixson theorem. See p. 218 of [Hirsch and Smale, 1974] for more details.

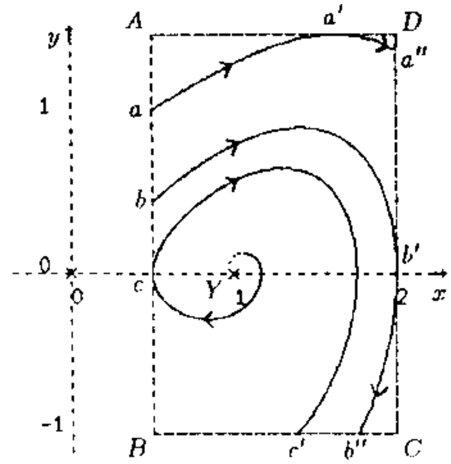


Figure 2: Key Orbits Characterizing the Behavior of (2) in $ABCD = \{(x, y) \mid 0.5 \leq x \leq 2, -1 \leq y \leq 1.5\}$

about the behavior even when nonlinear ODEs cannot be solved analytically and it is not possible to represent orbits as an explicit analytical function *of f.*

3 Flow Patterns and Global Analysis

In order to derive global behavior, we partition the phase space into regions, and characterize the local flow in each region, and put together the results of local analysis to reason about global behavior.[2]

We use *flow mappings* to represent local flow. For example, the local flow of (2) in region *ABC D* — {(x, *y) * $0.5 \leq x \leq 2, -1 \leq y \leq 1.5\}$ can be characterized by fixed points and several other key orbits as shown in figure 2. Flow mappings corresponding to this characterization is

$$\begin{aligned} &\overline{Aa} \rightarrow \overline{a'A} + \overline{Da'} \rightarrow \overline{a''D} \\ &+ \overline{ab} \rightarrow \overline{b'a''} + \overline{bc} \rightarrow \overline{c'b''} \\ &+ \overline{Y} \rightarrow \overline{cc'} + \overline{Cb'} \rightarrow \overline{b''C}. \end{aligned}$$

The first term of the above says that orbits transverse to boundary edge *A a* continuously maps points there onto boundary edge *a' A.* Either side of the arrow may as well be a fixed point as in the fifth term above, or a limit cycle.

Flow mappings can be obtained by searching for *points of contact,* such as a'. *b'. c*, ,4, *C. D,* where the orientation of flow flips from inward to outward or *vice versa.* We further classify points of contact into *concave nodes.* such as a', b', *c* where the orbits passing on these points lie inside the region immediately before and after the contact, and *convex nodes,* such as *A,* C. D, where the orbits lies outside the region before and after the contact.

Global analysis requires only topological aspects of flow. We use *flotv patterns* to represent the topology of flow. Figure 3(a) shows a symbolic representation of the flow in region *ABCD* in figure 2 and (b) gives a schematic representation.

[2]Currently, our technique cannot handle flow in open regions. This means that, our method of global analysis is limited to flow in a bounded region.

(a) a flow pattern:

constituents:	Y : type = fixed-point $\overline{A\,a}$: type = boundary-segment a : type = landmark ... c : type = concave-node ...
boundary-list:	$[\overline{A\,a}, a, \overline{a\,b}, \ldots]$
flow-mappings:	$\overline{A\,a} \rightarrow \overline{a'\,A}$ ... $Y \rightarrow \overline{c\,c'}$ ...

(b) schematic representation of (a)

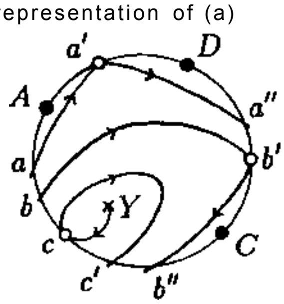


Figure 3: A Flow Pattern Representing the Local Flow in Region *ABCD* in figure 2

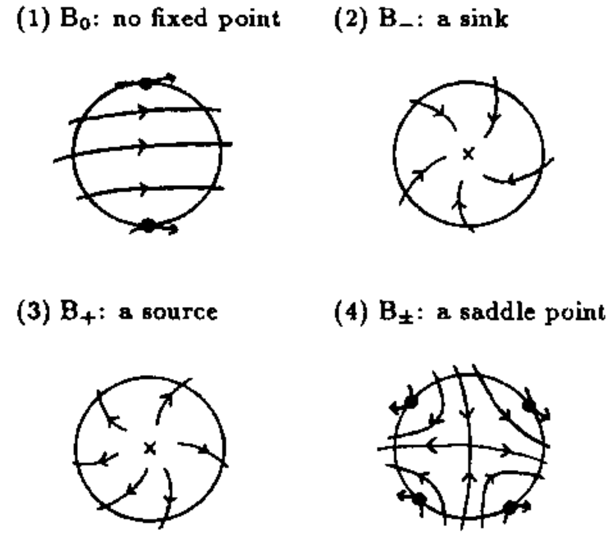


Figure 4: Basic Flow Patterns

The computability and complexity of generating flow patterns depend on the class of ODEs. For two-dimensional piece wise linear differential equations, which result from approximating each occurrence of nonlinear terms in a nonlinear differential equation by a set of connected pieces of linear functions, the almost all process is computable [Nishida and Doshita, 1990]. For more complex classes of ODEs, difficulties arise mainly because complete information may not be available due to the complexity of mathematical problems encountered [Nishida *et al.*, 1991]. This implies that the procedure may fail when it tries to divide the phase space into uniform regions or to annotate flow at the boundary of regions. Moreover, handcrafting procedures which can deal with these situations would be quite painstaking because of unmanageably many combination of possibilities. To overcome these difficulties, we take a grammatical approach, which will be presented in the next section.

The main stream of global analysis is to merge flow patterns in turn and examine topological properties of resulting flow patterns for larger regions. Limit cycles, if any, can be detected in finite steps as far as it is not totally contained in a single region as a result of phase space partition and the local flows in related regions are properly analyzed [Nishida *et al.*, 1991]. We use several heuristics to back up the incompleteness. Since attracting and repelling sets of planar ODEs are either fixed points or limit cycles, and since fixed points are determined in analysis of local flow, the above gives a complete process for global analysis in theory. Of course, there is a chance that the above method may fail or produce an incorrect result, due to the failure in characterizing the local flow or numerical error. This is a common difficulty we encounter in addressing nonlinear problems.

4 Flow Grammar

A flow grammar specifies all possible flow patterns in closed regions.[3] Generally, a flow grammar is a system $G = \langle B, C, D \rangle$, where B is a set of *basic flow patterns,* C a set of *composition rules,* and D a set of *distortion rules.*

In the rest of this paper, we will describe one particular flow grammar, though other formulation might be possible as well. Particularly, we assume that closed regions do not have fixed points on the boundary and that the given flow is *structurally stable.*[4] Moreover, the flow grammar presented below does not generate limit cycles for efficiency. Limit cycles are handled in the algorithm which calls for the flow grammar [Nishida *et al.*, 1991].

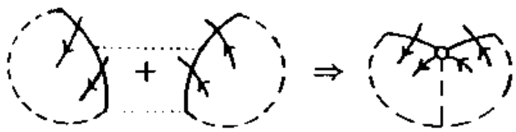
Basic flow patterns specify flow patterns which contain at most one fixed point. We have four basic flow patterns as illustrated in figure 4. Basic flow patterns have at most one type of points of contact on the boundary.

Composition rules specify the way the flow patterns with more than one fixed point are computed. Suppose we are to "fuse" a couple of flow patterns P_1 and P_2 together at boundary segments $a_1\,b_1$ of P_1 and 02 62 of P_2 In order for flow patterns P_1 and P_2 to be properly

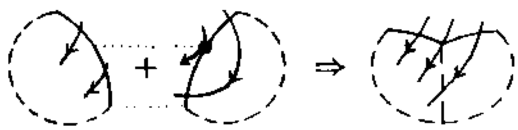
[3] The notion of flow grammar was inspired by Process Grammar [Leyton, 1988].

[4] Structurally stable flows are those which persist under an infinitesimal perturbation to their parameters. If the purpose is to analyze ODEs for physical systems, only structurally stable systems may be observed. Peixoto's theorem suggests that fixed points which may appear in structurally stable flows are either *sinks, sources,* or *saddle points* (see p. 60 of [Guckenheimer and Holmes, 1983] for more detail).

(1) introduction of a concave node: $cc \rightarrow c$

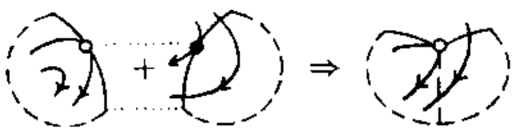


(2) deletion of a convex node: $cv \rightarrow c$



(3) merge into a convex node: vv — v

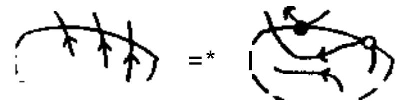
(4) merge into a concave node: $cv \rightarrow c$



• : convex node, o : concave node

Figure 5: Four Patterns Governing Fusion of Flow at End Points

(1) cv-distortion



(2) vc-distortion

Figure 6: Distortion Rules

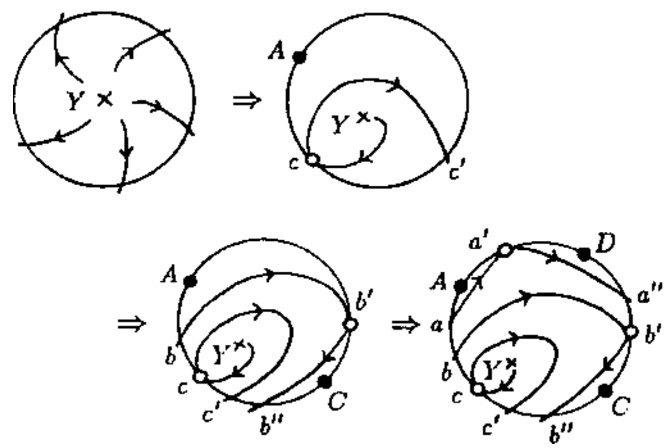
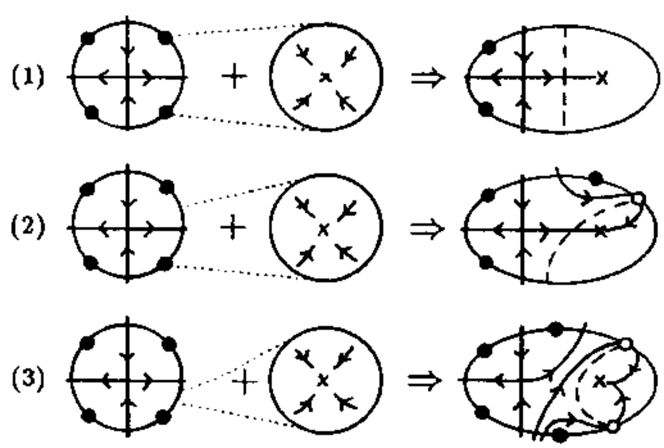


Figure 7: Derivation of Flow Patterns shown in Figure 3 (drawn schematically)



; (12 flow patterns in total)

Figure 8: Flow Patterns Formed From Basic Flow Patterns $B_\pm$ and B_

fused, the flow at the open intervals (ai, b_1) and (a_2, b_2) should be complementary: outgoing (incoming) flow of Pi should correspond to incoming (outgoing) flow of P_2, and each convex (concave) node of P_1 should correspond to a convex (concave) node of P_2. At the end points a, and 6,, flow should be either one of four patterns shown in figure 5.

Distortion rules specify flow patterns arising when the relative geometric relation between the flow and the boundary segment becomes complex. We have two distortion rules as shown in figure 6. One introduces a sequence of a concave node and a convex node into a boundary segment and the other introduces them in the reverse order.

Figure 7 illustrates how the flow pattern shown in figure 3 can be obtained by applying a sequence of distortion rules to the basic flow pattern B+. Figure 8 illustrates flow patterns formed by applying a composition rule to basic flow patterns B± and B_.

We classify flow patterns by the number of fixed points and points of contact involved: we use a tuple $(n_{sa}, n_{ss}, n_c, n_v)$ as an index to the finite set of flow patterns which have n_{sa} saddle points, n_{ss} sinks or sources, n_c concave nodes, and n_v convex nodes. Although the index appears four-dimensional, it is actually three-dimensional, as shown in the following theorem:

Theorem 1 For any C^1 structurally stable flow and a closed region R, let the number of saddle points and sinks or sources in R be n_{sa} and n_{ss}, respectively. Lit the number of convex nodes and concave nodes at the boundary dR of R be n_v and n_c. respectively. Then wt havt the following relation:

$$n_v - n_c = 2 \times (n_{sa} - n_{ss} + 1). \tag{4}$$

Outline of proof The theorem can be proved by the following facts: (a) the flow grammar G can produce all possible flow patterns which do not contain limit cycles, (b) introduction of limit cycles does not affect the index, (c) (4) holds for basic flow patterns, (d) application of distortion rules trivially retains (4). and (e) application of composition rules also retains (4), for the number of concave nodes and convex nodes should be equal at the boundary segments to be fused except the two end points, and $n_v - n_c$ decreases by one at the two end points of fusion in each of four patterns governing fusion (see figure 5). □

Since n_{sa}, n_{ss}, n_c, and n_v, are computed independently, the above theorem can be used as a constraint for detecting numerical errors or missing information. It can also be used to enumerate flow pattern efficiently. For example, if both n_v and n_c are 2, it follows from (4) that $n_{sa} - n_{ss} < -1$. Thus, we can start enumerating flow patterns from those with index (0,1,2,2).

5 Enumerating Flow Patterns

The flow grammar presented in the previous section is ambiguous in the sense that there usually exists more than one derivation which produces the same flow pattern. In order to cope with this unfortunate property, we have to rely on a generate-and-test method. In order to decrease the cost of enumeration, we use two techniques.

The first technique is to pose a constraint on the sequence of derivations so as to suppress a sequence of derivations which eventually produces a flow pattern to be generated otherwise. Flow pattern P is minimal if there is no other flow pattern Q such that P results from applying a distortion rule to Q. If is easy to determine whether a given flow pattern is minimal or not. We have found the following property:

Theorem 2 For any derivation sequence S which involves more than one application of distortion rules, there exists a derivation sequence S' such that S' produces the same flow pattern as S and application of distortion rules comes later than application of other rules in S'.

Thus, it suffices to first generate minimal flow patterns and then apply distortion rules.

The second technique is to use a short-hand representation of flow patterns to reduce the cost of comparing flow patterns. An f~rep is a cyclic list[5] of form $\langle\langle \ldots, s_i[x_{i,1}, \ldots, x_{1,n_i}], \ldots \rangle\rangle$. Each element of an f-rep corresponds to a boundary segment delimited by a couple of points of contact and qualitatively denotes in the counterclockwise order how points on the boundary segment are mapped by orbit intervals involved in the flow pattern. If the interval corresponding to X_{ij} is mapped from/to another boundary segment B, we use a positive integer indicating the relative position of B counted in the counterclockwise order from the current boundary segment. If it is either a fixed point or a limit cycle, we use a negative integer. s_i denotes the orientation of the flow there; it is + if the flow comes from the outside, and — if it leaves for the outside.

[5]We denote a cyclic list consisting of $x_1, \ldots, x_n$ as $\langle\langle x_1, \ldots, x_n \rangle\rangle$. By definition, $\langle\langle x_1, \ldots, x_n \rangle\rangle = \langle\langle x_2, \ldots, x_n, x_1 \rangle\rangle = \ldots = \langle\langle x_n, x_1, \ldots, x_{n-1} \rangle\rangle$.

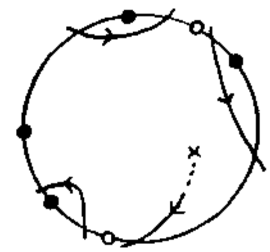


Figure 9: Observation (drawn schematically)

For example, an f-rep for the flow pattern in figure 3 is:

$$\langle\langle +[5,3,1], -[-1,5,1], +[5], -[3,1], +[5], -[1] \rangle\rangle \tag{5}$$

where, the first element is for boundary segment $\overline{Ac}$ and the second is for $\overline{cC'}$, and so on. Note that f-rep for a set of flow mappings is uniquely defined except the existence of variants which only differ from each other in the way fixed points are numbered. And importantly it seems that different flow mappings give different f-reps. More study is left for future.

6 Utility of Flow Grammar

The utility of a flow grammar is twofold: firstly, it enables to reason about complex patterns in a uniform manner, and secondly, it provides constraints. Both of these features are implemented in PSX2NL.

When no information is available about fixed points, PSX2NL switches to an approximate algorithm based on a generate-and-test method. The input is a set of observations consisting of partially traced orbits and estimated location of convex and concave nodes. A flow pattern P is an interpretation of an observation O if there exists (possibly empty) a set of assumptions A such that $O = P \cup A$. An interpretation is minimal when there is no other interpretation which explains the observation with smaller set of assumptions. PSX2NL uses the enumerator described in the previous section to generate flow patterns in turn and seeks for a minimal interpretation.

For example, given the observation shown in figure 9, PSX2IML produces twelve minimal interpretations, two of which are shown in figure 10. Currently, PSX2NL will simply increase the number of observations when

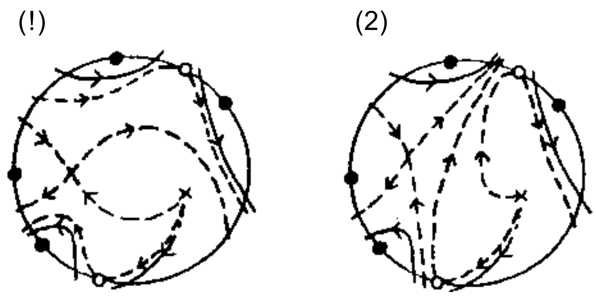


Figure 10: Minimal Interpretations of the Observation in Figure 9

more than one minimal interpretation is found Whenever PSX2NL detects a symptom suggesting that a limit cycle is contained in the given region, it will divide the region into two by a line across the limit-cycle-like orbit Focusing observation for resolving ambiguity would be an interesting problem left for future.

7 Related Work

Attempts to incorporate global information have been made by several authors [Lee and Kuipers, 1988; Struss, 1988]. Unfortunately, mast computational methods developed so far only make of the non-intersection constraint of orbits and some other partial constraints, and hence the ability of reasoning about global behaviors is still quite limited. Their weakness mostly comes from the lack of adequate representation. In contrast, flow patterns and a flow grammar provide a means for reasoning about various aspects of geometric constraints, allowing an envisioner to symbolically reason about the structure of global and asymptotic behaviors.

Intelligent analysis of nonlinear ODEs is quite a new field. POINCARE [Sacks, 1991] would be the first program addressing intelligent analysis of nonlinear ODEs. POINCARE integrates qualitative and quantitative methods, as PSX2NL does. Both POINCARE and PSX2NL work on planar ODEs including nonlinear ODEs, though POINCARE supports bifurcation analysis which is not yet implemented in PSX2NL. The difference in phase portrait analysis is that PSX2NL makes use of more representation than POINCARE and other programs based on the conventional simulation technology. This leads to three consequences. First, PSX2NL saves computational resources, for it keeps geometric and topological information in an more abstract form. For example, PSX2NL keeps information about only a few essential points on orbits, while POINCARE has to keep the location of all points on orbits. Second, PSX2NL can derive richer conclusion from the same observation obtained by quantitative analysis, as demonstrated in the previous section. Third, PSX2NL is more robust from incompleteness of information and numerical errors. For example, POINCARE rely on an external package in locating fixed points. If the package fails POINCARE fails, too. In contrast, PSX2NL can switch to a robust, approximate method based on a flow grammar.

A stochastic approach [Doyle and Sacks, 1989] is another candidate of uniform treatment of global behavior supported by mathematical theories. Since the stochastic approach possesses a complementary nature to ours, it would be interesting to seek a way for combining the two.

8 Further Work

An important work left for the future research is extension into higher dimensional flow. The research in that direction is quite challenging both theoretically and practically. Unfortunately, the extension of this work into higher dimensional flows is not trivial, for firstly flow patterns become far more complicated, secondly, representing higher dimensional geometric objects is hard, and thirdly it becomes subtle to characterize the topological structure of flow. However, we believe that the concepts exploited in this paper would be of much help in such extension.

References

[Doyle and Sacks, 1989] Jon Doyle and Elisha P. Sacks. Stochastic analysis of qualitative dynamics. In *Proceedings UCAI-89*. pages 1187-1192, 1989.

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

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

[Lee and Kuipers. 1988] Wood W. Lee and Benjamin J. Kuipers. Non-intersection of trajectories in qualitative phase space: A global constraint for qualitative simulation. In *Proceedings AAA 1-88,* pages 286-290. American Association for Artificial Intelligence, 1988.

[Levton, 1988] Michael Leytori. A process-grammar for shape. *Artificial Intelligence,* 34:213 247, 1988.

[Nishida and Doshita, 1990] Toyoaki Nishida and Shuji 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):311 355, 1990.

[Nishida *e1 al* 1991] Toyoaki Nishida, Kenji Mizutani, Atsushi Kubota. and Shuji Doshita. Automated phase portrait analysis by integrating qualitative and quantitative analysis, to be presented at AAA1-91, 1991.

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

[Struss, 1988] Peter Struss. Global filters for qualitative behaviors. In *Proceedings AAAI-88,* pages 275-279. American Association for Artificial Intelligence, 1988.

[Weld and de Kleer, 1989] Daniel S. Weld and Johan de Kleer, editors. *Readings in Qualitative Reasoning about Physical Systems.* Morgan-Kaufmann, 1989.