Automated Phase Portrait Analysis by Integrating Qualitative and Quantitative Analysis

Toyoaki Nishida and **Kenji Mizutani** and **Atsushi Kubota** and **Shuji Doshita**
Department of Information Science
Kyoto University
Sakyo-ku, Kyoto 606, Japan
email: nishida@kuis.kyoto-u.ac.jp

Abstract

It has been widely believed that qualitative analysis guides quantitative analysis, while sufficient study has not been made from technical viewpoints. In this paper, we present a case study with PSX2NL, a program which autonomously analyzes the behavior of two-dimensional nonlinear differential equations, by integrating knowledge-based methods and numerical algorithms. PSX2NL focuses on geometric properties of solution curves of ordinary differential equations in the phase space. PSX2NL is designed based on a couple of novel ideas: (a) a set of flow mappings which is an abstract description of the behavior of solution curves, and (b) a flow grammar which specifies all possible patterns of solution curves, enabling PSX2NL to derive the most plausible interpretation when complete information is not available. We describe the algorithms for deriving flow mappings.

Introduction

In spite of tremendous research efforts, most technical developments obtained in qualitative physics are still quite naive from the standards of other disciplines and real applications. An obvious way of escaping from the trap would be to avoid building the theory from the scratch. It would deserve a serious effort to develop a computational theory on top of existing theories with keeping the spirit of qualitative physics in mind.

In this paper, we present how such a schema is instantiated in PSX2NL, a program which autonomously analyzes the behavior of two-dimensional ordinary differential equations, by integrating knowledge-based methods and numerical algorithms. PSX2NL focuses on geometric properties of solution curves of ordinary differential equations in the phase space. PSX2NL is designed based on a couple of novel ideas: (a) a set of flow mappings which is an abstract description of the behavior of solution curves, and (b) a flow grammar which specifies all possible patterns of solution curves, enabling PSX2NL to derive the most plausible interpretation when complete information is not available. Taken together, these two techniques provide a well-founded computational framework for an autonomous intelligent mathematical reasoner.

We have chosen two-dimensional nonlinear differential equations as a domain and analysis of long-term behavior including asymptotic behavior as a task. The domain is not too trivial, for it contains wide varieties of phenomena; while it is not too hard for the first step, for we can step aside from several hard representational issues which constitute an independent research subject by themselves. The task is novel in qualitative physics, for nobody except a few researchers has ever addressed it.

Analysis of Two-dimensional Nonlinear Ordinary Differential Equations

The following is a typical ordinary differential equation (ODE) we investigate in this paper:

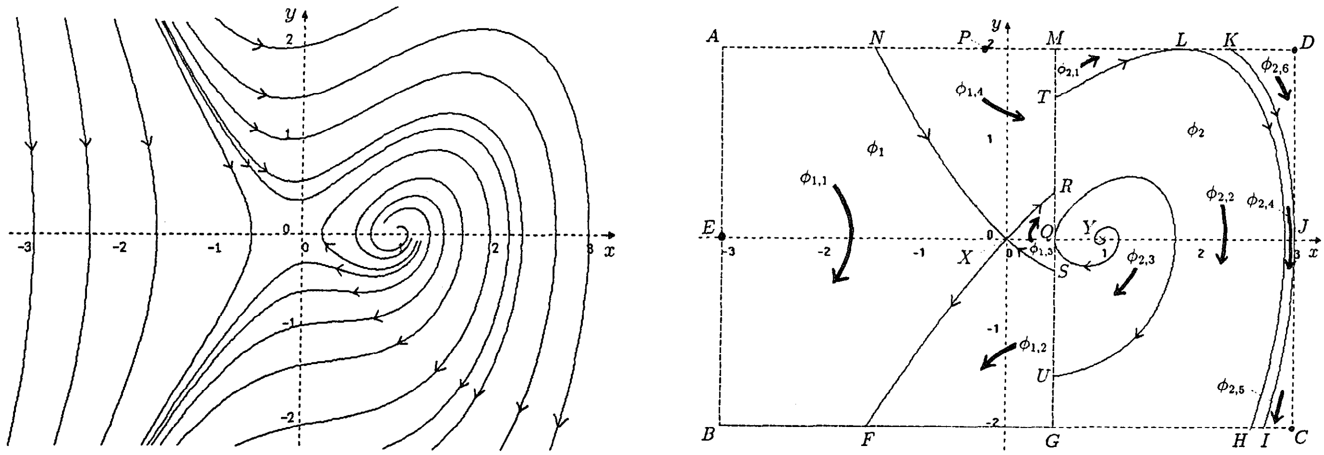
$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = x - x^2 + 0.2y + 0.3xy. \end{cases} \quad (1)$$

This is two-dimensional in the sense that it is specified by two independent state variables $x(t)$ and $y(t)$, and it is nonlinear in the sense that the right-hand sides contain nonlinear terms such as xy and x^2. Unlike linear differential equations, no general algorithm is known for solving nonlinear ODEs analytically. In addition, the behavior of nonlinear differential equations may become fairly complex in certain situations.

A strategy taken by applied mathematicians is to study geometric properties of solution curves in the phase space spanned by independent state variables [Hirsch and Smale, 1974, Guckenheimer and Holmes, 1983]. That approach has made it possible to understand the qualitative behavior of nonlinear differential equations, even when the explicit form of solution is not available. Figure 1 illustrates a portion of the *phase portrait* of (1), a collection of solution curves contained in the phase space. (a) is drawn in an rather ad-hoc manner by manually invoking a numerical integrator and (b) is one that PSX2NL has produced to understand the behavior in adjacent regions $ABGM$ and $MGCD$. Notice that only crucial orbits are drawn there. Even from (a) it would be much easier to grasp the global characteristics of

(a) manually drawn using a numerical integrator

(b) produced by PSX2NL for local analysis



Arrow-heads and some symbols are added by hand.

Figure 1: The Phase Portrait for (1)

behavior such as: y decreases as time passes, there exists some solutions which approaches to an equilibrium state $x = 1, y = 0$ as $t \rightarrow -\infty$, and so on.

The collection of solution curves is also called a *flow*, for it introduces a mapping which maps a given point in the phase space into another as a function of t. Each solution curve, called an *orbit*, represents a solution under some initial condition.

Theoretically, an orbit is a directed curve such that its tangent vector conforms to the vector field specified by a given ODE at each point in the phase space. A *fixed point* is a special orbit consisting of a single point at which $\frac{dx}{dt} = \frac{dy}{dt} = 0$. A fixed point corresponds to an equilibrium state which will not evolve forever. Fixed points are further classified into *sinks*, *sources*, *saddles*, and some other peculiar subcategories according to how orbits behave in their neighborhood. If the solution of a given ODE is unique as is often the case, orbits never cross with themselves or other orbits. An orbit must be cyclic if it intersects with itself.

Under many circumstances, it is crucial to understand the long-term behavior of a given ODE. In structurally stable two-dimensional ODEs, orbits may diverge or approach a fixed point or a cyclic orbit when $t \rightarrow \pm\infty$. An orbit is called an *attracting* orbit when all orbits in some neighborhood approach it arbitrarily after a long run. Similarly, a repelling orbit is an orbit which all orbits in some neighborhood approach as $t \rightarrow -\infty$. Conversely, when an orbit o_1 approaches orbit o_2 as $t \rightarrow \infty$ $(-\infty)$, we call o_2 the *asymptotic destination (origin)* of o_1. In this paper, we study the behavior of qualitatively coherent bundles of orbits rather than individual orbits.

It is possible to obtain an approximate picture of an orbit by using a numerical integration algorithm such as the Runge-Kutta algorithm, as was done for examples in this paper. However, numerical methods support only a small portion of the entire process of understanding the behavior, as pointed out in [Yip, 1988]. It is necessary to plan numerical simulation and interpret the result. In order for a program to carry out the entire process without much external assistance, the program has to possess sufficient knowledge about nonlinear differential equations. In order to integrate numerical and knowledge-based methods, one must address several questions concerning (a) representation of orbits, (b) algorithm for generating the representation, and (c) algorithm for reasoning about global behavior based on the representation. The central concern of this paper is the second and to demonstrate how qualitative and quantitative analysis are integrated to achieve the goal. Before describing that, we present an overview of our solution to these three questions in the next section.

Outline of our Solution

PSX2NL is a program which autonomously analyzes the behavior of two-dimensional ODEs. PSX2NL takes as input the specification of an ODE such as (1) and a region of the phase space to analyze. As output PSX2NL produces (a) a list of possible asymptotic origins and destinations and (b) a set of *flow mappings*, an abstract description of the behavior of the given ODE in the given region. In this section, we briefly describe flow mappings and their use in reasoning about long-term behavior, and then we present the overall picture of the algorithms for deriving flow mappings.

Flow Mappings as Abstract Representation of the Phase Portrait

We represent a flow as a set of *flow mappings* which specifies how the flow maps points in the phase space. For example, consider the flow in the region $MGCD$ shown in figure 1(b). Orbits transverse to segment $\overline{TQ}$ continuously map points on the segment to polyline $\overline{UGH}$. We represent the fact as $\phi_{2,2} : \overline{TQ} \to \overline{UGH}$, where $\phi_{2,2}$ is a label attached to the bundle of orbit intervals between $\overline{TQ}$ and $\overline{UGH}$. We call $\overline{UGH}$ the *destination* of $\overline{TQ}$, and $\overline{TQ}$ the *origin* of $\overline{UGH}$ (with respect to the given region). The whole flow ϕ_2 in region $MGCD$ is represented as a sum of sub-flows corresponding to bundles of orbit intervals in $MGCD$, as follows:

$$\begin{aligned}\phi_2 = \;& \phi_{2,1} : \overline{MT} \to \overline{LM} \oplus \phi_{2,2} : \overline{TQ} \to \overline{UGH} \\ & \oplus \phi_{2,3} : Y \to \overline{QU} \oplus \phi_{2,4} : \overline{KL} \to \overline{HI} \\ & \oplus \phi_{2,5} : \overline{CJ} \to \overline{IC} \oplus \phi_{2,6} : \overline{DK} \to \overline{JD}.\end{aligned} \quad (2)$$

Given a flow ϕ and a region R, we may denote the origin and destination of geometric object x with respect to R as $\phi^{-1}(x)$ and $\phi(x)$, respectively. Thus, the flow ϕ_2 in region $MGCD$ introduces relations such as $T = \phi_2^{-1}(L)$, $L = \phi_2(T)$, $\phi_2(L) = H$, and so on. Note that there are some subtle cases. For example, $\phi_2^{-1}(Q) = Y$ but $\phi_2(Y) = \overline{QU}$; or $\phi_2^{-1}(T)$ is not defined in region $MGCD$.

In order to generate flow mappings, it is necessary to identify characteristic points such as Q and characteristic orbits such as orbit $Q \rightsquigarrow U$ or $T \rightsquigarrow L \rightsquigarrow H$.

The role of flow mappings is twofold. First, flow mappings are used to reason about long-term behavior. Second, flow mappings serve as an intermediate representation for qualitative and quantitative analysis to interact. Consider Van der Pol's equation

$$\begin{cases} \frac{dx}{dt} = -2x^3 + 2x + 2y \\ \frac{dy}{dt} = -x. \end{cases} \quad (3)$$

The flow illustrated in figure 2(a) is characterized by flow mappings:

$$\begin{aligned}\phi_1 = \;& \phi_{1,1} : \overline{A\,\phi_1^{-1}(R)} \to \overline{RA} \\ & \oplus \phi_{1,2} : \overline{\phi_1^{-1}(R)\,S} \to \overline{\phi_1(S)\,\phi_1(R)} \\ & \oplus \phi_{1,3} : \overline{B\,\phi_1^{-1}(S)} \to \overline{SB} \\ & \oplus \phi_{1,4} : \overline{\phi_1^{-1}(S)\,P} \to \overline{\phi_1(P)\,\phi_1(S)} \\ & \oplus \phi_{1,5} : \overline{E\,\phi_1^{-1}(P)} \to \overline{PE} \\ & \oplus \phi_{1,6} : \overline{\phi_1^{-1}(P)\,Q} \to \overline{\phi_1(Q)\,\phi_1(P)} \\ & \oplus \phi_{1,7} : X \to \overline{Q\,\phi_1(Q)} \\ & \oplus \phi_{1,8} : \overline{FR} \to \overline{\phi_1(R)\,F}\end{aligned} \quad (4)$$

for region $ABEF$, and

$$\phi_2 : \overline{CDFQ} \to \overline{QEC} \quad (5)$$

for region $ECDF$. By combining the two,

$$\phi_2(\overline{FQ}) = \overline{\phi_2(F)\,Q} \subset \overline{\phi_1^{-1}(P)\,Q} \quad (6)$$

$$\phi_1(\overline{\phi_1^{-1}(P)\,Q}) = \overline{\phi_1(Q)\,\phi_1(P)} \subset \overline{FQ}, \quad (7)$$

and hence,

$$\phi_1 \circ \phi_2(\overline{FQ}) \subset \overline{FQ}. \quad (8)$$

(8) entails that there exists an attracting bundle of orbits containing at least one limit cycle transverse to $\phi_1 \circ \phi_2(\overline{FQ}) = \overline{\phi_1(Q)\,\phi_1 \circ \phi_2(F)}$. See figure 2(b).

Notice that qualitative and quantitative analysis are integrated in the lines of reasoning described above. The main thread is concerned with topological aspects of bundle of orbit intervals and hence it is qualitative, while the basic facts are obtained by quantitative analysis.

The global analyzer incorporated in PSX2NL recognizes attracting or repelling bundles of orbits containing a limit cycle by chaining flow mappings in turn and looking for such patterns of argumentation as described above. The domains of attraction and repelling are also determined in this process. The algorithm for the global analyzer was originally developed for PSX2PWL, the predecessor of PSX2NL, and is described in [Nishida and Doshita, 1990].

Outline of the Algorithms for Deriving Flow Mappings

PSX2NL uses multiple strategies. The most opportunistic strategy called the *algorithm* $\top$ depends on the assumption that complete information required for generating flow mappings can be obtained by available quantitative problem solvers. The most pessimistic strategy called the *algorithm* $\bot$ does not depend on such assumptions, instead it uses approximate methods to derive plausible interpretation. There are many intermediate levels between the two depending on availability of information.

The Algorithm $\top$

In order to generate a set of flow mappings, it is necessary to know the location and the type of fixed points in the given region and to classify the boundary of the region into a set of maximally continuous boundary segments so that orbits transverse to each of these segments may be coherent in the sense that the behavior of these orbits in the region is qualitatively equal. Useful clues for classifying the behavior are obtained by recognizing every *point of contact* on the boundary at which the orbit is tangent to the boundary and lies in the same side of the boundary immediately before and after contact. A point of contact is called a *concave node* if the orbit passing on it lies inside the region immediately before and after contact. Otherwise a point of contact is called a *convex node*. Note that a convex node of a region is a concave node of its adjacent region, and vice versa. Thus, Q is a convex node of region $ABGM$ in figure 1(b), while it is a concave node of region $MGCD$.

The most straightforward algorithm for generating a set of flow mappings for a given region R is this:

(step $\top$1) identify fixed points and their type in R;

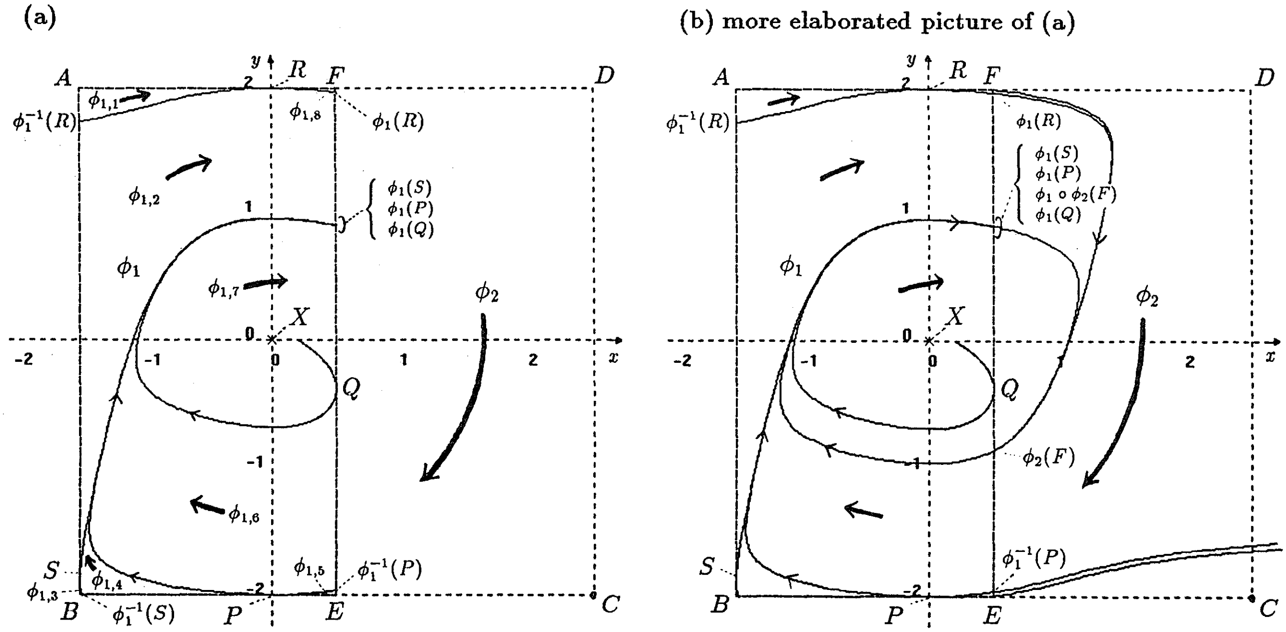


Figure 2: The Phase Portrait for Van der Pol's Equation (3)

(step T2) divide the given region into a set of subregions (called *cells*) $\{\ldots C_i \ldots\}$ so that at most one fixed point is contained in each cell;

(step T3) for each cell C_i, repeat the following:

- identify points of contact and determine their type;
- for each concave node, trace the orbit passing on it forward and backward until the traced orbit meets the boundary again, or it comes appropriately near one of the known fixed points, or it is judged that the orbit approaches a known or unknown cyclic orbit; if the traced orbit is almost cyclic, split C_i by a line crossing the detected almost-cyclic orbit, and recursively apply (step T3) to the resulting cells;
- for each saddle node in C_i, trace the orbits in stable and unstable manifolds,[1] similarly;
- generate flow mappings for C_i;

(step T4) generate flow mappings for the region by aggregating analysis obtained for each cell.

In order to pursue the steps of the algorithm T, PSX2NL has to solve varieties of mathematical problems. For example, PSX2NL has to solve simultaneous nonlinear equations $\frac{dx}{dt} = \frac{dy}{dt} = 0$, to obtain the location of fixed points. Thus, for (1) PSX2NL has to solve: $y = 0 \wedge x - x^2 + 0.2y + 0.3xy = 0$. In order to identify points of contact of a given cell, PSX2NL has to solve an (often nonlinear) equation with a single variable. PSX2NL uses a simple mathematical problem solver including an equation solver and a numerical integrator using the Runge-Kutta algorithm. When encountered with a more complex problem, PSX2NL calls Macsyma. However, there do exist many problems which cannot be solved, even by a powerful mathematical tool like Macsyma or by a numerical method. If Macsyma ends up in failure, PSX2NL will switch to a less precise but more robust algorithm such as the algorithm $\perp$.

As for ODE (1), all information is available with a simple equation solver except finding the location of fixed points. Fortunately, Macsyma can solve the remaining problem. Thus, PSX2NL performs the following steps to analyze the behavior of (1) in region $ABCD$ in figure 1(b): first, PSX2NL recognizes that the region contains two fixed points: a saddle at $(0,0)$ and a source at $(1,0)$ (designated as X and Y, respectively); PSX2NL then divides the region into two cells $ABGM$ and $MGCD$, each containing one fixed point (note that PSX2NL splits the region in the middle point between the two fixed points, but this is not essential); PSX2NL correctly recognizes four convex nodes for cell $ABGM$, and three concave nodes and three convex nodes for cell $MGCD$; PSX2NL traces orbits in stable and unstable manifolds of the fixed point X and

[1] As for the definitions, see for example [Guckenheimer and Holmes, 1983].

marks points F, S, R and N as important landmarks at which these orbits meet with the boundary; similarly, by tracing forward and backward the orbits passing on concave nodes J, L, and Q of region $MGCD$, PSX2NL identifies landmarks I, K, H, T, and U, and it also concludes that the orbit passing on Q comes from the source Y; finally, PSX2NL produces a set of flow mappings for $ABCD$ by aggregating the flow mappings for each cell.

Note that what is obtained in the above may not be a logical (or mathematical) conclusion derived from a given ODE in the sense that some of the rules used in the derivation are heuristic ones based on numerical approximation. For example, PSX2NL concludes that the asymptotic destination of an orbit is a sink if the traced orbit enters a predetermined small neighborhood of the sink. The quality and reliability of interpretation will be improved by introducing more mathematical knowledge and increasing the accuracy of numerical computation, but it may not totally solve the problem.

The Algorithm $\perp$

The algorithm $\perp$ takes care of the situation in which no information is available about the location of fixed points and points of contact due to difficulties of mathematical problems involved. The basic idea to overcome the difficulty is to enumerate all possible patterns of flow in turn, compare each pattern with observation, and pick out one which achieves the best match. To represent a pattern of flow, we use a *flow pattern* consisting of a set of flow mappings and description of geometric objects referred to by flow mappings. We have introduced a *flow grammar* to describe the set of all possible flow patterns one may encounter. The reader is referred to [Nishida and Doshita, 1991] for details of the flow grammar we use.

Like the algorithm $\top$, the algorithm $\perp$ takes ODE f and region R and produces a set of flow mappings for R as interpretation. The algorithm is this:

- (step $\perp$1) sample the flow at the boundary of R and calculate the orientation of the flow there;
- (step $\perp$2) aggregate the observed flow into a set of maximal hypothetical boundary fragments so that the orientation of flow at each boundary fragment may be coherent;
- (step $\perp$3) assume a point of contact for each pair of sample points such that the orientation of the flow has flipped from inward to outward or vice versa, and mark the pair as a delimiter of a point of contact; furthermore, infer numerically the type of each assumed point of contact;
- (step $\perp$4) construct a partial flow pattern for R by tracing an orbit forward and backward from each inferred point of contact (as a result, boundary edges may be divided by new landmarks);

  if the traced orbit exhibits periodic behavior, split the current region into two by a line cutting across the periodic portion of the orbit, and apply the algorithm $\perp$ recursively to resulting sub-regions;
- (step $\perp$5) enumerate flow patterns and look for one which matches the partial flow pattern constructed by observation;
- (step $\perp$6) if such a flow pattern is found, break the process or go to (step $\perp$4) depending on the computation resource allocated; if no flow pattern is found, repeat the whole process by increasing precision of observation;

Generally, flow patterns generated in earlier cycles are simpler and hence more probable than those produced later. Hence, we prefer flow patterns enumerated earlier. If more than one flow pattern matches in one cycle, we prefer the one with minimal number of constituents which are not supported by observation.

What PSX2NL Can Do and Cannot Do

Qualitative and quantitative analysis are integrated in PSX2NL as described above. In the algorithm $\top$, qualitative analysis determines what to compute, and quantitative analysis provides an answer. In the algorithm $\perp$, qualitative analysis generates hypotheses when complete information is not available, and quantitative analysis provides evidences for or against them.

Preliminary implementation of PSX2NL has been completed using a simple blackboard architecture. PSX2NL handles fixed point detection and classification, saddle manifold construction, flow map construction, limit cycle detection, and attractor basin detection. A bifurcation analyzer has not been implemented yet. The search process taken by PSX2NL is heuristic in nature. There is no theoretical proof that PSX2NL can find all fixed points, all limit cycles, even though we restrict the class of flows to structurally stable ones whose flow is more regular than otherwise. Thus, the result depends on the quality and variety of numerical algorithms and other mathematical tools available from PSX2NL. Limit cycle detection by PSX2NL is not weak as it may appear. The global analysis algorithm employed by PSX2NL always detects the existence of a limit cycle and identifies the asymptotic property of the bundle of orbit intervals containing the limit cycle, as far as the phase space division by boundaries of cells cuts across the limit cycle. In addition, PSX2NL tries to cut across a cyclic orbit whenever it detects symptoms.

Thus, the ability of PSX2NL should be evaluated by experimentation. Currently, we have found out that it works well for several structurally stable flows. Comprehensive evaluation is left for future.

Comparison with Related Work

Kuipers pointed out the importance of integration of qualitative and quantitative analysis and incorporated quantitative methods into QSIM [Kuipers and Berleant, 1988]. We have taken a quite different approach and base the framework on rigid mathematical theories, as was done in [Abelson *et al.*, 1989]. Yip is concerned with analyzing discrete dynamical systems represented by difference equations [Yip, 1988]. His approach is to develop a framework of intelligent numerical experimentation based on mathematical knowledge about the domain. Though in a similar spirit, the techniques presented in this paper are for analyzing continuous systems and bear quite different features from those employed by Yip. In particular, representation issues become more critical for continuous systems.

Sacks reported work on analysis of two-dimensional piecewise linear differential equations [Sacks, 1990]. Sacks uses transition diagrams as internal representation of flow [Sacks, 1990]. Unfortunately, the expressive power of transition diagram is quite limited and in order to reason about long-term behavior it is necessary to use a separate set of heuristic rules or introduce a probabilistic technique [Doyle and Sacks, 1989]. In contrast, flow mappings presented in this paper have more expressive power, enabling to reason about long-term behavior. [2]

Forbus presented a framework in which modeling language and simulation are closely integrated [Forbus and Falkenhainer, 1990]. Giving the framework an ability of analytical thinking as introduced in this paper will make his proposal more powerful.

Future Work

There are several short-term problems left for future research, other than those pointed out in the above. Among others lots should be done for improving the efficiency of the matching algorithm for the algorithm $\perp$. One obvious way is to compile a set of candidates into a discrimination net. A more long-range, and challenging issue is extension into higher-dimensional systems. We believe that insights obtained in two-dimensional phase spaces would be of great conceptual, even though not technical, help towards that goal.

[2] One reviewer of this paper pointed out that there is much overlap between our work and Elisha Sacks' recent work. According to the reviewer, Sacks' recent work was published in Computing Systems in Engineering 1:2 (p. 607) 1990 and Proc. of the 29th IRRR Conference on Decision and Automation, as well as is forthcoming from AIJ. Unfortunately we were ignorant of them, for one has not been published yet and others are not included in common readings of AI. Detailed comparison is left for future.

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.

Doyle, Jon and Sacks, Elisha P. 1989. Stochastic analysis of qualitative dynamics. In *Proceedings IJCAI-89.* 1187–1192.

Forbus, Kenneth D. and Falkenhainer, Brian 1990. Self-explanatory simulations: An integration of qualitative and quantitative knowledge. presented at 4th International Workshop on Qualitative Physics, Lugano, Switzerland.

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.

Kuipers, B. J. and Berleant, Daniel 1988. Using incomplete quantitative knowledge in qualitative reasoning. In *Proceedings AAAI-88.* 324–329.

Nishida, Toyoaki and Doshita, Shuji 1990. 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.

Nishida, Toyoaki and Doshita, Shuji 1991. A geometric approach to total envisioning. unpublished research note.

Sacks, Elisha 1990. Automatic qualitative analysis of dynamic systems using piecewise linear approximations. *Artificial Intelligence* 41:313–364.

Yip, Kenneth Man-kam 1988. Generating global behaviors using deep knowledge of local dynamics. In *Proceedings AAAI-88.* American Association for Artificial Intelligence. 280–285.