Automating Analysis of Qualitative Behaviors of Ordinary Differential Equations

Toyoaki Nishida
Department of Information Science, Kyoto University
Sakyo-ku, Kyoto 606-01, Japan
phone: 81-75-753-5386
email: nishida@kuis.kyoto-u.ac.jp

Abstract

In this working note, I discuss issues in automating qualitative analysis of systems of ordinary differential equations (ODEs). The central issues in analysis automation are the development of a computational theory for dynamical systems analysis and the accumulation of knowledge on dynamical systems analysis at the computer-executable level. I focus on integrating qualitative and quantitative methods so that a computer program can automatically make high-level decisions and derive abstract information by intelligently controlling numerical and symbolic computation. I survey several techniques developed in the PSX project and describe the current status.

Intelligent Scientific Computing and the PSX Project

In this working note, I discuss issues in automating qualitative analysis of systems of ordinary differential equations (ODEs).

In science, analysis automation is expected to release scientists from painstaking efforts of preparing, monitoring and interpreting the result of numerical computation for interesting phenomena. In engineering, analysis automation is crucial to constructing model-based problem solving systems.

Analysis automation plays a complementary role to scientific visualization. Scientific visualization assumes the existence of domain experts who have an ability of visual cognition and ample domain knowledge for interpreting the meaning of what they see. Since our ability of visual cognition is limited to low-dimensional spaces, special techniques are needed to visualize phenomena in higher-dimensional spaces. In contrast, analysis automation is aiming at developing a computational theory and its implementation that is executable by computers without human assistance.

The central issues in analysis automation are the development of a computational theory for dynamical systems analysis and the accumulation of knowledge on dynamical systems analysis at the computer-executable level. More specifically, we must address:

- encoding textbook knowledge
- elucidating the nature of experts' common sense, intuition, and expertise
- developing a computational theory of high-level decision making involving planning numerical analysis, controlling and monitoring numerical analysis, interpreting the result of numerical analysis, modifying the plan of numerical analysis, and making decisions with partial information
- interfacing with numerical and symbolic computation
- integrating heterogeneous knowledge
- having computers formulate new theories using machine learning and computational discovery techniques.

In this working note, I focus on the third issue.[1] I survey how the issue is handled in the PSX project, whose goal is building a computer program called PSX,[2] that can automatically investigate the qualitative behavior of a given system of ODEs.

In what follows, I first describe the domain and the task. I then present our approach. Finally, I briefly show the limitations of the current approach.

Qualitative Analysis of Systems of ODEs

In science and engineering, systems of first order ODEs of the form:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}) \qquad f : \mathrm{R}^n \rightarrow \mathrm{R}^n \tag{1}$$

on unknown functions $\mathbf{x}(t) = \{x_1(t), \ldots, x_n(t)\}$ ($t \in \mathrm{R}$) are often used to characterize the behavior of dynamical systems. Equation (1) is said to be linear if f is linear, and nonlinear otherwise. Systems of linear ODEs can be solved analytically in the sense that

[1] For the fifth, the reader is referred to [Nishida, 1992].

[2] PSX stands for "Phase Space eXplorer."

it is possible to compute the representation of $\mathbf{x}$ as a function of t provided that the eigenvalues and eigenvectors of an $n \times n$ coefficient matrix can be obtained. In addition, the behavior of systems of linear ODEs is simple and can be classified into a small number of categories according to the types of eigenvalues of their coefficient matrix.

Systems of nonlinear ODEs are not that simple. They cannot always be solved analytically, there are infinitely many varieties of behavior, and the behaviors themselves may become fairly complex and pathological under certain conditions. The properties of nonlinear ODEs are studied in applied mathematics, from qualitative points of view. Dynamical systems theories [Hirsch and Smale, 1974; Guckenheimer and Holmes, 1983] provide a mathematical basis for answering such questions as:

- How does a given system of ODEs behave after a long run?
- How stable is a given system of ODEs under perturbations?
- How does the behavior pattern of a parameterized system of ODEs change as some parameters are changed?

Let us introduce several basic concepts of dynamical systems theories. Consider the following system of ODEs:

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

This system of ODEs is called Van der Pol's equation, and is concerned with two independent *state variables* x and y which vary as a function of time t. Given a particular pair of values for these variables, the vector of state evolution $(\frac{dx}{dt}, \frac{dy}{dt})$ is determined from the right-hand side. In this sense, the system of ODEs (2) introduces the *vector field* in the *phase space* spanned by state variables, as shown in Figure 1(a). Equation (2) is nonlinear in the sense that the right-hand sides contain a nonlinear term x^3.

In ordinary situations, the vector field implicitly specifies the *phase portrait*, a set of non-intersecting directed curves such that each directed curve is tangent to the vector field. Part of the phase portrait for Van der Pol's equation is shown in Figure 1(b). Each curve, called an *orbit*, represents a solution to the given system of ODEs under some initial condition.

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 sub-categories according to how orbits behave in their neighborhood. If the solution of a given system of ODEs 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 system of ODEs. As for systems of ODEs on a plane, each orbit either diverges or approaches a fixed point or a cyclic orbit when $t \to \pm\infty$. 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 $t \to -\infty$. Conversely, when an orbit o_1 approaches orbit o_2 as $t \to \infty$ $(-\infty)$, we call o_2 the *generalized sink* (*source*) of o_1.

According to an applied mathematician,[3] qualitative analysis investigates the following issues in turn:

- How many attractors there are
- Their approximate locations in the phase space
- The topological structure of the attractors
- The approximate locations of basins
- The structure of the basins.

In computer science, numerical integration algorithms such as the Runge-Kutta algorithm are available which allow to trace orbits in the phase space. However, it is not enough. Numerical methods support only a small portion of the entire process of understanding the behavior. We need to develop a high-level decision making procedures for planning numerical analysis, controlling and monitoring numerical analysis, interpreting the result of numerical analysis, modifying the plan of numerical analysis, and making decisions with partial information, and integrate it with conventional numerical and algebraic packages.

The core issues are:

1. Representation of flow: we have to represent flow in an appropriate form that allows to reason about the qualitative behavior;
2. Control structure: we have to design a control structure that allows low-level computation (numerical and symbolic computation) and high-level qualitative reasoning to interact and play complementary roles.

It should be noted that neither low-level procedures nor high-level procedures are complete; high-level procedures are incomplete in the sense that they cannot yield any conclusion unless evidences are provided, whereas low-level procedures are incomplete in the sense that they cannot determine what to compute. Hence, numerical and symbolic algorithms and high-level decision-making procedures should closely interact with each other.

The Approach of the PSX Project

The goal of the PSX project is building a computer program called PSX, that can automatically investigate the qualitative behavior of a given system of

[3] H. B. Stewart in a lecture at the Department of Electrical Engineering, Kyoto University on July 11, 1989.

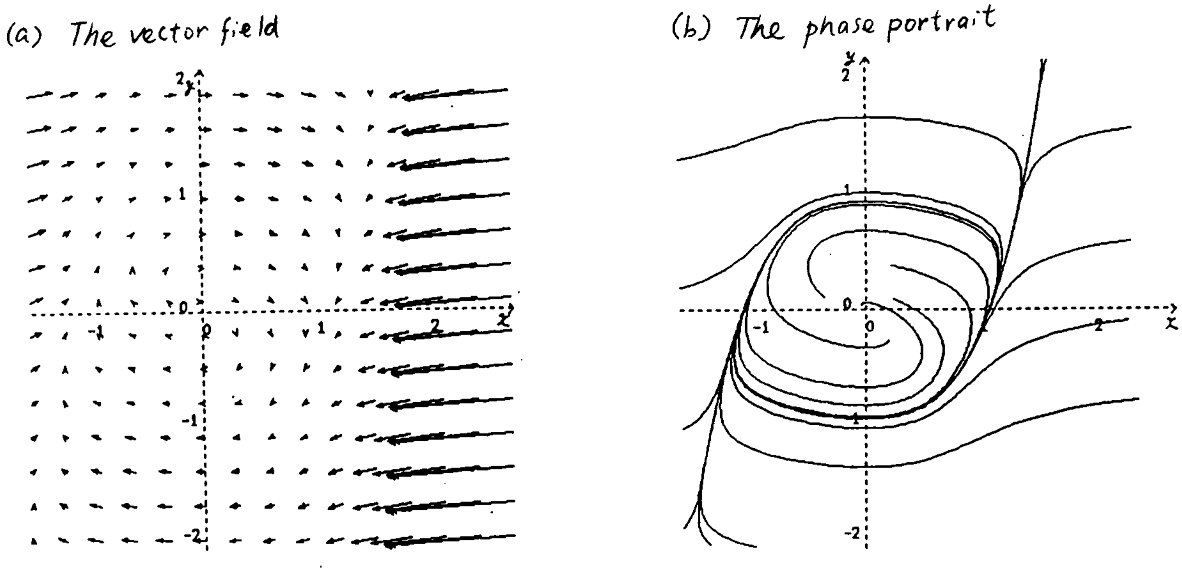

Figure 1: Vector Field and the Phase Portrait of Van der Pol's Equation (2)

ODEs. Currently, we have three separate implementations. Two of them, PSX2PWL and PSX2NL, are fully implemented which handle systems of ODEs in two-dimensional phase spaces [Nishida *et al.*, 1990; Nishida and Doshita, 1991; Nishida *et al.*, 1991]. The remaining one, PSX3, is for those in three-dimensional phase spaces and is partly implemented [Mizutani *et al.*, 1992]. Although we have employed slightly different techniques for implementation, I will emphasize the common features underlying these implementations and describe as if PSX's were a single uniform program.

What PSX Can Do

The input to PSX is:

- a system of ODEs of the form:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}) \tag{3}$$

  or a system of piecewise-linear ODEs of the form:

$$\{R_i : \langle p_i(\mathbf{x}) \mid \frac{d\mathbf{x}}{dt} = f_i(\mathbf{x})\rangle\} \qquad 1 \le i \le n, \mathbf{x} \in \mathrm{R}^n \tag{4}$$

  where, $p_i(\mathbf{x})$ is a conjunction of linear inequalities which specifies the range of linear region R_i, and $f_i(\mathbf{x})$ is a linear function[4] of $\mathbf{x}$ which specifies the flow in the linear region R_i.
- a bounded region.[5]

[4]We include those which can be transformed into linear functions by translation of coordinates.

[5]The current implementation of PSX2NL can analyze the behavior of a given system of ODEs only in a bounded region of the phase space.

The output contains both qualitative descriptions (*e.g.*, those concerning classification of orbits according to their asymptotic behaviors), and quantitative descriptions (*e.g.*, numerical characterization of important orbits). More specifically, the output includes:

- a set of locations where one or more orbit approaches as $t \to \pm\infty$
- a classification of orbits according to their asymptotic behavior
- a set of rough geometric characterization of each orbit class given in the previous item.

Current implementation of PSX is based on the following key ideas:

- representing flow as a collection of *flow mappings*, each of which represents a bundle of nearby orbits.
- combining top-down and bottom-up analysis procedures with a blackboard model as a basis.

In the rest of this section, I present an informal description of the basic algorithm employed in PSX2NL and the control structure on a blackboard model. For more details, the reader is referred to [Nishida *et al.*, 1990; Nishida and Doshita, 1991; Nishida *et al.*, 1991; Nishida, 1992].

The Basic Algorithm — Informal Description

Roughly, we characterize the flow in a bounded region in terms of how each point on the boundary is mapped by the flow in the region. A point on the boundary where an orbit is coming into the region is mapped either to another point on the boundary, or to a fixed point or a limit cycle in the region. Similar proposition holds for points where orbits come out of the region. We aggregate continuous points on the boundary provided they are mapped to or from an open continuous segment of the boundary, the same fixed point, or the same limit cycle, and represent the flow as a collection of flow mappings between these continuous segments or fixed points or limit cycles. For example, the flow in $ABEF$ in Figure 2 is represented as a sum of sixteen flow mappings, as follows:[6]

$$\begin{aligned}\Phi_1 = \ & \Phi_{1,1} : \overline{A,\, \Phi_1^{-1}(R)} \rightarrow \overline{R,\, A} \\ & \oplus \Phi_{1,2} : \overline{\Phi_1^{-1}(R),\, S} \rightarrow \overline{\Phi_1(S),\, \Phi_1(R)} \\ & \oplus \Phi_{1,3} : \overline{B,\, \Phi_1^{-1}(S)} \rightarrow \overline{S,\, B} \\ & \oplus \Phi_{1,4} : \overline{\Phi_1^{-1}(S),\, P} \rightarrow \overline{\Phi_1(P),\, \Phi_1(S)} \\ & \oplus \Phi_{1,5} : \overline{E,\, \Phi_1^{-1}(P)} \rightarrow \overline{P E} \\ & \oplus \Phi_{1,6} : \overline{\Phi_1^{-1}(P),\, Q} \rightarrow \overline{\Phi_1(Q),\, \Phi_1(P)} \\ & \oplus \Phi_{1,7} : X \rightarrow \overline{Q,\, \Phi_1(Q)} \\ & \oplus \Phi_{1,8} : \overline{F, R} \rightarrow \overline{\Phi_1(R),\, F} \\ & \oplus \Phi_1^{-1}(R) \rightarrow R \oplus R \rightarrow \Phi_1(R) \\ & \oplus \Phi_1^{-1}(S) \rightarrow S \oplus S \rightarrow \Phi_1(S) \\ & \oplus \Phi_1^{-1}(P) \rightarrow P \oplus X \rightarrow Q \\ & \oplus Q \rightarrow \Phi_1(Q) \oplus X,\end{aligned} \tag{5}$$

where $\alpha \rightarrow \beta$ means that every point in sub-region α is mapped to somewhere in sub-region β by the local flow in the cell.

In order to construct a collection of flow mappings for a cell we seek points at which the orbit is tangent to the boundary, as well as some other geometric clues such as the location and type of fixed points. Let us call a point on the boundary such that the orbit is tangent to the boundary a *point of contact*. Points of contact are further classified into *convex nodes* and *concave nodes*. At a convex node, the orbit lies outside the cell immediately before and after it reaches the point of contact. A concave node is defined similarly.

The qualitative behavior of a given system of ODEs is obtained by putting together a collection of flow mappings for each cell and examining the structure of the resulting collection of flow mappings. For example, the flow in the cell $FECD$ of the phase space for Van der Pol's equation (see the top of Figure 2) can be represented as:

$$\Phi_2 : \overline{C, D, F, Q} \rightarrow \overline{Q, E, C}. \tag{6}$$

If we combine two sets of flow mappings for the two cells $ABEF$ and $FECD$ of the phase space for Van der Pol's equation (see the center and bottom of Figure 2), we obtain

$$\begin{aligned}\Phi_2(\overline{F,Q}) &= \overline{\Phi_2(F), Q} \subset \overline{\Phi_1^{-1}(P), Q} & (7)\\ \Phi_1(\overline{\Phi_1^{-1}(P), Q}) &= \overline{\Phi_1(Q), \Phi_1(P)} \subset \overline{F, Q}, & (8)\end{aligned}$$

and hence

$$\Phi_1 \circ \Phi_2(\overline{F,Q}) \subset \overline{F,Q}. \tag{9}$$

Formula (9) means that all orbits passing through the interval $\overline{F,Q}$ never leave the interval, and hence we can conclude from the Poincaré-Bendixon theorem[7] that there exists an attracting bundle Φ_C of orbits containing at least one limit cycle that is transverse to $\Phi_1 \circ \Phi_2(\overline{F,Q})$.[8] It also follows that all orbits passing through the interval $\overline{F,Q}$ tend towards Φ_C as $t \rightarrow \infty$. See Figure 2 for the entire derivation.

PSX3 applies the ideas presented above to systems of piecewise linear ODEs in a three-dimensional phase space. Figure 3 shows important orbits PSX3 has traced to produce a set of flow mappings for a system of piecewise linear ODEs:[9]

$$\begin{array}{lll} \{ \ \mathrm{R}_1 : \langle & 0.2 \leq x \leq 2 & \\ & \wedge\, 0.2 \leq y \leq 1 - \frac{3}{2}x & \\ & \wedge\, -0.2 \leq z \leq 2 & | \\ & \frac{dx}{dt} = 2x, & \\ & \frac{dy}{dt} = -3y - \frac{1}{2}z, & \\ & \frac{dz}{dt} = -\frac{1}{2}z & \rangle, \\ \quad \mathrm{R}_2 : \langle & 0.2 \leq x \leq 2 & \\ & \wedge\, 0.2 \leq y2 & \\ & \wedge\, -0.2 \leq z \leq 2 & \\ & \wedge\, 1 - \frac{3}{2}x \leq y & | \\ & \frac{dx}{dt} = -x - 2y + 2, & \\ & \frac{dy}{dt} = 3x - y - \frac{1}{2}z - 2, & \\ & \frac{dz}{dt} = -\frac{1}{2}z & \rangle\}. \end{array} \tag{10}$$

Top-down and Bottom-up Analysis on a Blackboard Model

In order to facilitate intensive coupling of the processes at different levels of abstraction, we have employed a blackboard model as a basis of a control scheme. Thus, the system has a shared memory space (the blackboard) and a library of processes (KSs: knowledge sources). KSs interact indirectly by updating the contents of the blackboard.

The blackboard model makes it possible to implement multiple strategies. In normal situations, PSX2NL examines the flow in a bottom-up manner, attempting to build a qualitative description according to a prescribed fixed sequence: PSX2NL first examines whether the given flow has one or more fixed points; if so, it determines their type. If there is more than one fixed point in the given region, PSX2NL partitions the given region into several cells so that at most one fixed point may be contained in each cell; it then examines the geometric features of the flow, traces key

[6]The first eight are essential. The others involve some subtlety, but this is not critical to our discussion.

[7]See p. 248 of [Hirsch and Smale, 1974] for more details.

[8]Note that $\Phi_1 \circ \Phi_2(\overline{F,Q}) = \overline{\Phi_1(Q), \Phi_1 \circ \Phi_2(F)}$.

[9]PSX3 automatically produces a command sequence which allows Mathematica to produce a three-dimensional graphics for the user.

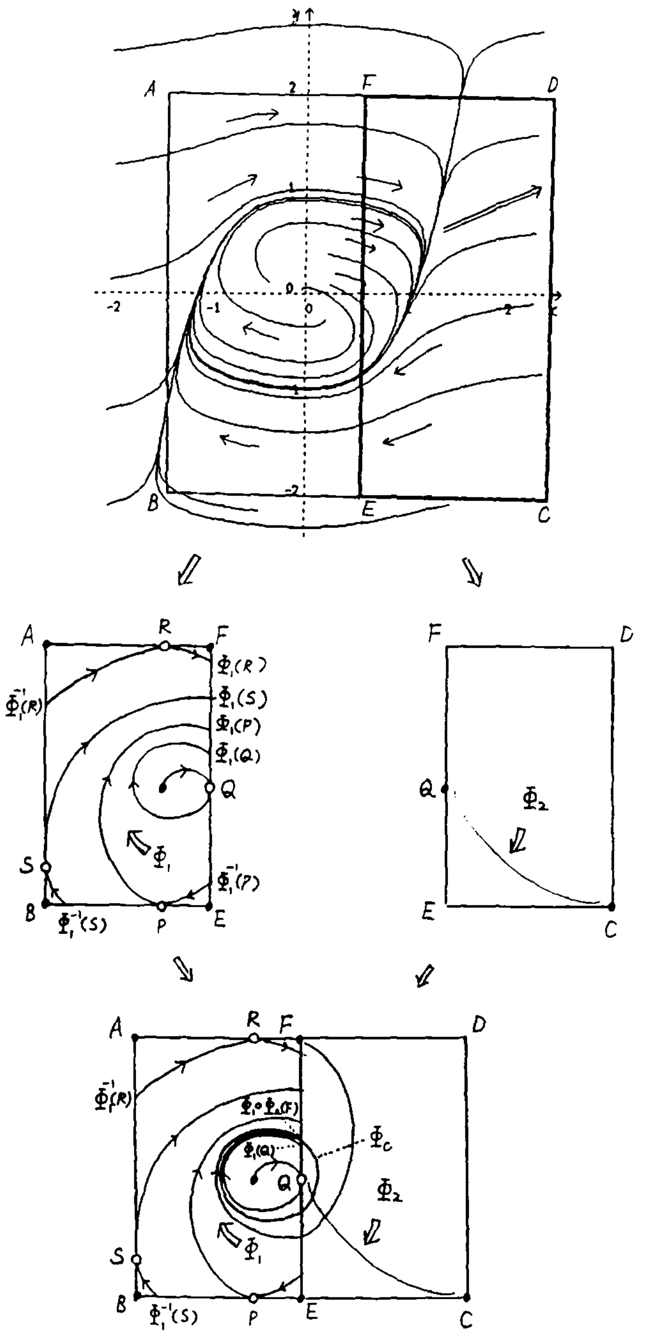

Figure 2: Reasoning about Qualitative Behavior with Flow Mappings

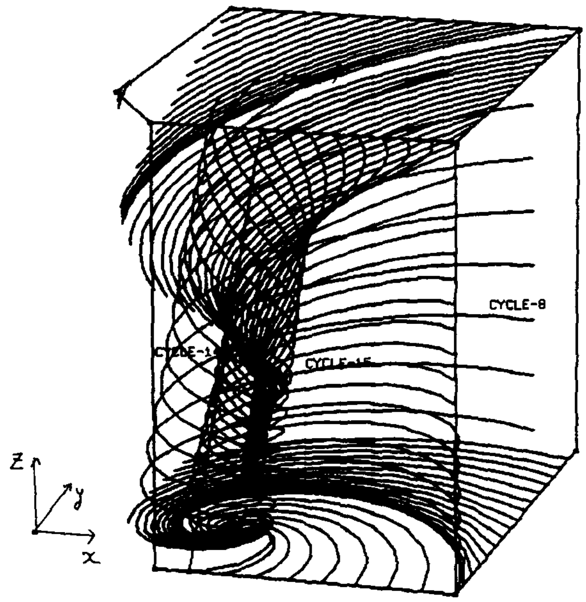

Figure 3: Trace of Important Orbits PSX3 Traced for (10)

orbits by numerical integration if necessary, generates a set of flow mappings for the given region, investigates the properties of the flow mappings, and derives conclusions about the qualitative behavior of the given region. Thus, an abstract description is gradually constructed from less abstract descriptions.

If something goes wrong and the standard sequence turns out to be intractable, the analysis process switches to the top-down mode, trying to find the most plausible interpretation that matches the observations made so far. As a knowledge source, PSX2NL uses a *flow grammar*, which is a grammatical description of all possible behavior patterns.

The flow grammar provides PSX2NL with theoretical constraints, allowing it to operate in a top-down manner. More specifically, the flow grammar allows PSX2NL

1. To predict the existence of key orbits and to plan numerical computation to find their location:
2. To focus numerical computation: for example, if there is more than one possible interpretation of an observation, PSX2NL will plan numerical computation that is expected to resolve the ambiguity;
3. To rearrange the analysis process when an unexpected result is obtained: for example. when a symptom of an unexpected limit cycle is observed, PSX2NL will divide a region into cells and try to prove the existence of a limit cycle by numerical and symbolic computation;

4. To detect inconsistent numerical results and propose a plausible interpretation: for example, non-intersection constraints of orbits may be violated because of numerical errors; in such cases, PSX2NL suggests the most plausible interpretation;

Discussion

Our approach to designing and implementing PSXs has the following limitations:

1. All knowledge used for problem solving must be represented in a procedural form as KSs;
2. The terminology for representing objects and relations has not been carefully designed;
3. No attempt has been made to incorporate intuitive guidance, possibly obtained from statistics.

Research on automating qualitative analysis of dynamical systems has just begun [Sacks, 1990; Sacks, 1991; Yip, 1991; Zhao, 1991; Kalagnanam, 1991]. To make significant progress beyond what has been achieved in PSX and its siblings, it is necessary to incorporate the large-scale knowledge on dynamical systems and their analysis techniques that has been set down in handbooks (e.g., [Zwillinger, 1989]).

References

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

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

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

Mizutani, Kenji; Nishida, Toyoaki; and Doshita, Shuji 1992. Automated behavior analysis of piecewise linear ordinary differential equations in three-dimensional phase space. Technical Report AI91-65, IEICE.

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

Nishida, Toyoaki; Fujiwara, Hiroshi; and Doshita, Shuji 1990. Abstracting flow as mapping. In *Proceedings of PRICAI '90.* 384–389.

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

Nishida, Toyoaki 1992. Towards integration of heterogeneous knowledge for highly autonomous analysis of dynamical systems — preliminary report from the PSX project. (*Journal of Information Processing*, to appear).

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

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

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.

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

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

Zwillinger, Daniel 1989. *Handbook of Differential Equations.* Academic Press.