MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1274 March 1991

Extracting and Representing Qualitative Behaviors of Complex Systems in Phase Spaces

Feng Zhao

Abstract

We develop a *qualitative* method for understanding and representing phase space structures of complex systems. To demonstrate this method, a program has been constructed that understands qualitatively different regions of the phase spaces and represents and extracts geometric shape information about these regions, using deep domain knowledge of dynamical system theory. Given a dynamical system specified as a system of governing equations, the program applies a successive sequence of operations to incrementally extract the qualitative information and generates a *complete, high level symbolic description* of the phase space structure, through a combination of numerical, combinatorial, and geometric computations and spatial reasoning techniques. The high level description is *sensible* to human beings and *manipulable* by other programs. We are currently applying the method to a difficult engineering design domain in which controllers for complex systems are to be automatically synthesized to achieve desired properties, based on the knowledge of the "shapes" of the systems.

Keywords: qualitative reasoning about complex systems, nonlinear dynamical systems, knowledge representation, automated control analysis and synthesis.

This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-89-J-3202, and in part by the National Science Foundation grant MIP-9001651. The author is also supported by a G.Y. Chu Fellowship.

Extracting and Representing Qualitative Behaviors of Complex Systems in Phase Spaces*

Feng Zhao

MIT Artificial Intelligence Laboratory

545 Technology Square, Room 438

Cambridge, MA 02139

December 1990

(revised March 1991)

Abstract

We develop a *qualitative* method for understanding and representing phase space structures of complex systems. To demonstrate this method, a program has been constructed that understands qualitatively different regions of the phase spaces and represents and extracts geometric shape information about these regions, using deep domain knowledge of dynamical system theory. Given a dynamical system specified as a system of governing equations, the program applies a successive sequence of operations to incrementally extract the qualitative information and generates a *complete, high level symbolic description* of the phase space structure, through a combination of numerical, combinatorial, and geometric computations and spatial reasoning techniques. The high level description is *sensible* to human beings and *manipulable* by other programs. We are currently applying the method to a difficult engineering design domain in which controllers for complex systems are to be automatically synthesized to achieve desired properties, based on the knowledge of the "shapes" of the systems.

*This paper is to appear in IJCAI-91.

1 Introduction

Analysis of dynamical systems via phase space structures plays an increasingly important role in experimenting, interpreting, and controlling complex systems [1, 2]. Nonlinear systems usually fall outside the domain of traditional analysis, such as Fourier analysis for linear systems. However, most of the important qualitative behaviors of a nonlinear system can be made explicit in the phase spaces with a phase space analysis.

We have constructed a program for understanding and representing qualitative structures of phase spaces through a combination of numerical, combinatorial, and geometric computations and techniques of spatial reasoning. The program uses theoretical knowledge about nonlinear dynamical systems. We will illustrate our techniques of extracting and representing the qualitative features of the phase space structures with two and three dimensional systems. The techniques presented in this paper also apply to higher dimensional dynamical systems.

Complex systems are usually nonlinear and high dimensional. Our theoretical knowledge about nonlinear dynamical systems is far from complete. Therefore, many engineering applications reply on extensive numerical experiments. A numerical simulation typically generates an immense amount of quantitative information about a complex system. To interpret the numerical result and to use the information for engineering designs, it is essential to develop qualitative methods that automatically analyzes the system, extracts the qualitative features, and represents them in a high level description sensible to human beings and manipulable by other programs.

This paper demonstrates a qualitative method for automatically understanding and representing the "shapes" of nonlinear dynamical systems. Our ultimate goal is to develop a class of intelligent and autonomous controllers that understand the phase spaces of complex systems, sense the world, synthesize control commands, and affect the processes. For example, an intelligent controller would balance an inverted pendulum that is mounted on a moving cart pulled by a motor, through qualitatively analyzing the pendulum system, monitoring the motion of the system, and commanding the motor, much like what we would do to balance a broom on its end with a hand. Accomplishing such difficult tasks by autonomous robots would be hard to imagine without their understanding of the qualitative behaviors of the systems, especially when the systems are of *high* order and operate in *nonlinear* regimes. We are particularly interested in automating the control analysis and synthesis for a class of nonlinear systems that do not lend themselves easily to traditional analysis and design techniques.

2 Automated Qualitative Analysis of Phase Space Structures

The phase spaces of nonlinear dynamical systems often consist of qualitatively different regions. The "shapes" of dynamical systems refer to the geometric information about the structures and spatial arrangements of these regions. A key component of the qualitative analysis of the "shapes" is to determine the stability regions of the dynamical system. The geometric information about the stability regions is extremely useful in analyzing stabilities of control designs for complex systems, such as electric power systems and mechanical control systems, as well as in economics, ecology, etc.

Our program understands qualitatively different regions and extracts and represents geometric shape information about these regions. Given a dynamical system specified as a system of governing equations, the program generates a complete, high level symbolic description of the phase space structure as the result of the analysis. The high level description can be used as input to other programs for further computations. We are currently using the result to automatically synthesize control laws for nonlinear dynamical systems.

2.1 The qualitative phase space structures

We are interested in representing the qualitative features of dynamical systems for engineering analysis and design. For this purpose, the qualitative phase space structure of a dynamical system within the phase space region of interest is characterized by the equilibrium points and limit cycles and their stability types, the geometric structures of stability regions associated with the attractors, and the spatial arrangement of the equilibrium points, limit cycles, and stability regions.

We review some of the basic concepts in dynamical system theory in order to describe the qualitative phase space structures. Let us consider a single pendulum perturbed from its downward resting position. It swings around its vertical axis with a smaller and smaller amplitude, and eventually settles to the resting position

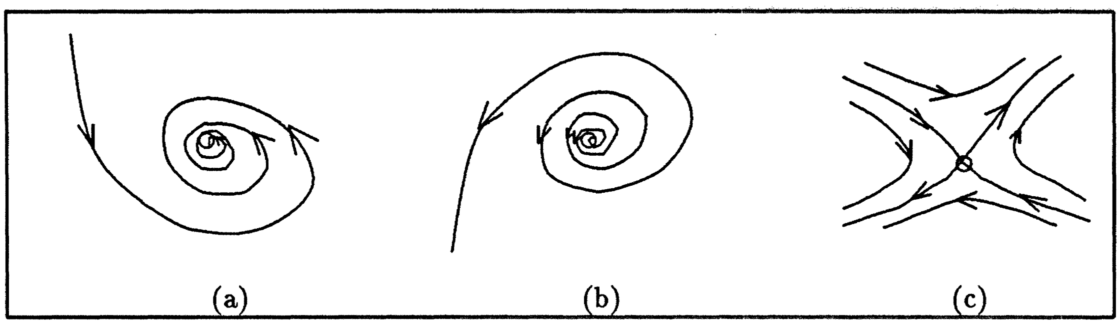


Figure 1: Equilibrium points: (a) attractors, (b) repellors, and (c) saddles.

due to frictions. We call such a resting state a stable *equilibrium point.* We say that an initial state of the pendulum is in the *stability region* of the stable equilibrium point if the pendulum starting from that state eventually settles to the stable equilibrium point. In our example, the stability region of the pendulum includes all the possible initial states.

In general, the equilibrium points of a dynamical system $x' = f(x, u)$, where u is a parameter, are the zeros of the vector field $f(x, u) : R^n \rightarrow R^n$. Structurally stable systems [6] can have equilibrium points of three types: stable equilibrium points (attractors), unstable equilibrium points (repellors), and nonstable equilibrium points (saddles), whose local behaviors in phase spaces are shown in Figure 1. An attractor is an equilibrium point that nearby trajectories approach in forward time. In our pendulum example, the downward resting state is an attractor. A repellor is the one that repels nearby trajectories. One can think of it as an attractor in reverse time. A saddle has trajectories approach it in some directions and trajectories leave from it in the other directions. The upward state of the pendulum is a saddle. Start the pendulum resting at this position. A slight perturbation would move it from the position. On the other hand, the pendulum with just the right amount of energy could swing to the upward position and rest there, although it will be extremely rare and will take an infinite amount of time to do so. From our discussion it is clear that the equilibrium points are asymptotic behaviors of dynamical systems. The other class of asymptotic behaviors are limit cycles, quasi-periodic orbits, and chaotic attractors. Although our techniques apply to limit cycles and quasi-periodic orbits, we shall not discuss them in details in this paper.

The important concept of stability is associated with the stability regions. The collection of trajectories approaching an equilibrium point is called the stable trajectories of the point; and the collection of trajectories leaving from an equilibrium point is called the unstable trajectories of the point. A saddle has stable trajectories along some directions and unstable trajectories along the other directions. The union of the stable trajectories of an attractor is its stability region, often called the basin of attraction for the attractor. The stability region of an attractor has the following properties. It is a simply connected and open region, either bounded or unbounded. Every trajectory starting in the region will be attracted to the attractor, by definition. The boundary of the region contains saddles or repellors only. The region is unbounded if the boundary contains no repellors. In planar systems, the stability region of an attractor contains no holes.

2.2 Automated qualitative analysis

One component of our program is to determine the boundaries of stability regions—the stability boundaries. Our algorithm for determining stability boundaries is based on a crucial theoretical result characterizing the stability boundaries of a fairly large class of dynamical systems. Under certain weak conditions that will be discussed in Section 4.1, the result of Chiang et al. [5] shows that the stability boundary for an attractor consists of the stable trajectories of equilibrium points and limit cycles whose unstable trajectories approach the attractor. This allows us to numerically determine a collection of trajectory points on the stability boundary through calculations of the stable and unstable trajectories. In planar systems, the boundaries consist of curve segments obtained from trajectories. In higher dimensions, we can approximate the boundary surfaces with a set of trajectories on the boundary.

We need to extract the geometric information about the stability regions from the numerical results about the stability boundaries, and represent it parsimoniously such that it facilitates further computations. For examples, the representation is to be used to estimate the volumes of the stability regions, to reason about the spatial relations with the stability boundaries, to compute topological properties of the regions, to extract information about trajectory flows, etc. Given a set of trajectory points on the stability boundary, the minimal representation—the one with fewest edges and preserving topological structures—is the polyhedron having those boundary points as vertices. Furthermore, the polyhedron is contained in the convex hull of the boundary points.

The geometric information about a stability region is represented as a polyhedron, tightly stretched over the trajectory points on the stability boundary. Extraction of the polyhedral approximation proceeds in two steps: computing a triangulation of the convex hull containing the polyhedron, and eliminating exterior triangles. The convex hull is computed and tessellated with simplices (triangles, tetrahedra, etc.) by a triangulation method—the Delaunay triangulation. The polyhedral approximation is then extracted by a "sculpture method" used in visual information representation [4]. Simplices exterior to the polyhedron are eliminated by heuristic rules.

2.3 The algorithm

We present the following algorithm for analyzing, extracting, and representing qualitative features of a dynamical system of any order in the phase space.

(1) **Identify qualitative behaviors:**

- (a) locate equilibrium points/limit cycles and classify their stability types;
- (b) compute stable and unstable trajectories for each saddle/limit cycle;
- (c) identify those saddles/limit cycles whose unstable trajectories approach an attractor;
- (d) the stability boundary for the attractor is the union of the stable trajectories of those saddles/limit cycles identified in (c);
- (e) check if consistency rules are violated. If yes, look for missing equilibrium points/limit cycles and go to step (a). Otherwise, go to the next step.

(2) **Extract geometric structures:**

- (a) for each attractor, collect stability boundary points;
- (b) tessellate the convex hull of the boundary points with a triangulation;
- (c) extract a polyhedral approximation to the stability region.

(3) **Summarize qualitative behaviors and generate a high level description:**

- (a) compile the phase space data structure from step 1 into a relational graph;
- (b) augment the graph with the geometric structure from step 2;
- (c) report the graph as the output.

The set of consistency rules specify the conditions for the stability boundaries and are used in the algorithm to automatically locate missing saddles.

1. *The Existence Rule*: Every stability region of an attractor has a boundary in a phase space with multiple attractors;

2. *The Separation Rule*: Separatrices either form a closed surface or become unbounded on all ends.

The first rule states the existence of stability boundaries in a phase space with multiple attractors. The second rule describes the separation property of multiple stability regions. The separatrices are stability boundaries that separate two stability regions.

The first step of our algorithm is based on a numerical method proposed by Parker and Chua [8] for numerically determining stability boundaries of planar systems. We have augmented their method with the set of consistency rules they suggested to automate the locating of saddles. Since the Newton-Raphson method used in finding equilibrium points requires an initial guess, Parker-Chua's method uses a grid to set up initial guesses and is able to find all the stable and unstable equilibrium points under normal circumstances. However, they require that the initial guesses for saddles be provided manually by the user. We seek to automate saddle locating by focusing the search for missing saddles on the most likely places using partial boundary information already obtained, or by refining the initial guesses for the Newton-Raphson method. We want to emphasize that our algorithm is valid for higher dimensional systems as well and generates a symbolic description of the phase space structure. Parker-Chua's method is designed for numerically analyzing planar systems only.

We have constructed a program using the above algorithm to analyze the qualitative behaviors of nonlinear dynamical systems. The program is implemented in Scheme, a dialect of LISP. All the numerical routines are implemented in Scheme as well. The flow chart of our program is shown in Figure 2.

The input to the program is a system of governing equations for a dynamical system. We could also start with a set of measured states from experiments. The phase space could then be reconstructed using standard methods for phase space reconstruction from time-series.

2.4 The main illustration

We illustrate how the algorithm is used to compute the high level description of a dynamical system with an example. Consider a 2nd order nonlinear system

$$\begin{cases} x' = -3x + 4x^2 - xy/2 - x^3 \\ y' = -2.1y + xy + u \end{cases} \tag{1}$$

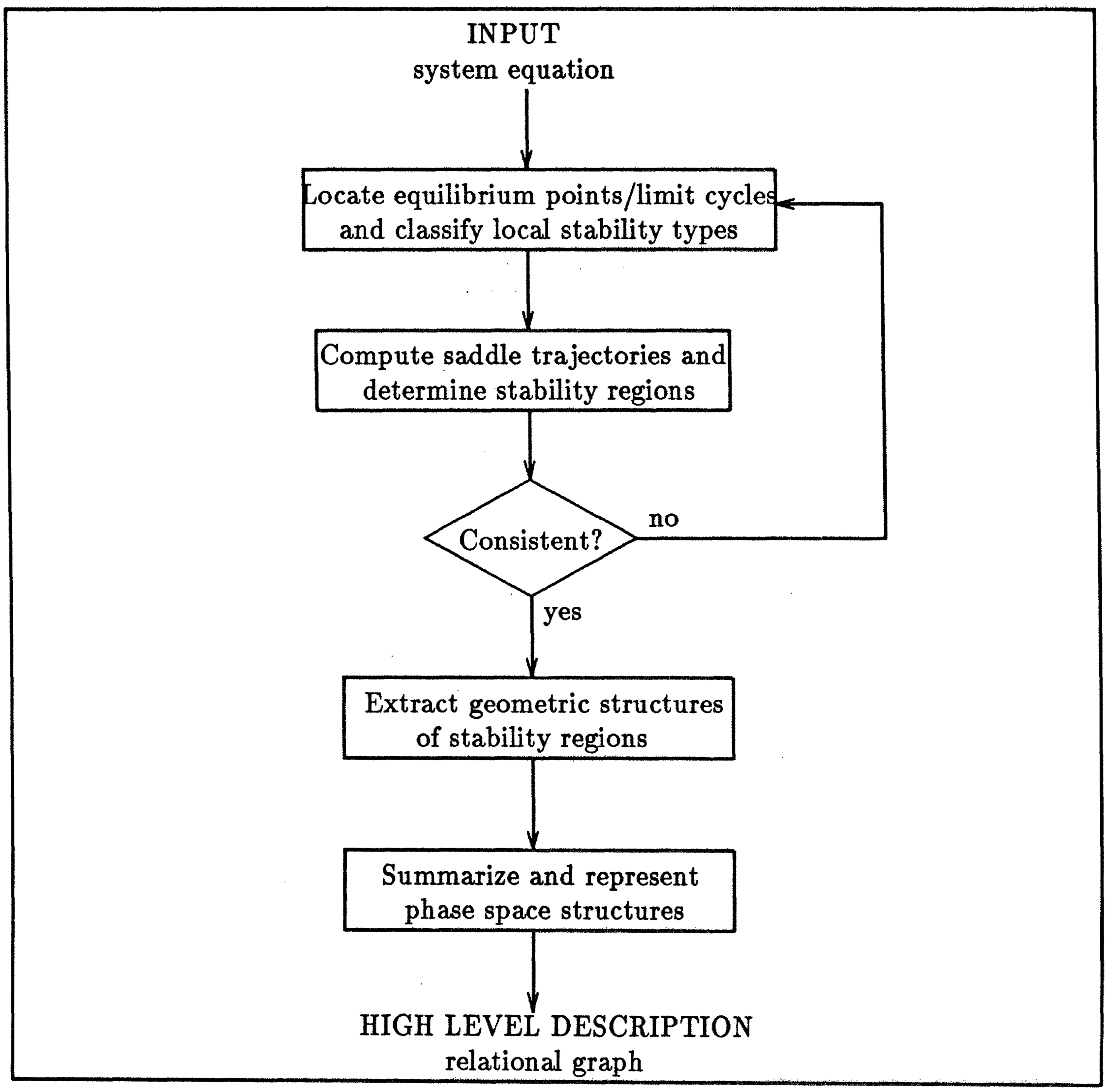


Figure 2: The flow chart of the program.

where u is a parameter. For the parameter value $u = 0.2$, the vector field within the region $-1.0 \leq x \leq 4.0$ and $-1.0 \leq y \leq 4.0$ is shown in Figure 3(a). We use the algorithm to analyze the qualitative behaviors of the system within this region.

The equilibrium points of the system are found by a zero finding method on $f(x, u)$—the Newton-Raphson method. The program locates four equilibrium points within the region and classifies their stability by inspecting the eigenvalues of Jacobians at the equilibrium points: two attractors at $(0.0, 0.095)$ and $(2.0, 2.0)$ and two saddles at $(1.05, 0.19)$ and $(3.05, -0.21)$, all shown in Figure 3(b) (Note that the attractors are represented with the symbol $+$ and the saddles are represented with the symbol $\oplus$.). The stable and unstable trajectories of the saddle are then computed by integrating the system from a small neighborhood of the saddle in the directions of the stable and unstable eigenvectors backwards and forwards, respectively.

Since one of the unstable trajectories of each saddle goes to the attractor at $(2.0, 2.0)$, the stability boundary of the attractor consists of the stable trajectories of both saddles. Similarly, the stability boundary of the attractor at $(0.0, 0.095)$ consists of the stable trajectories of the saddle at $(1.05, 0.19)$, one of whose unstable trajectories goes to the attractor. However, within the region of interest there are trajectories that leave the bounding box. These trajectories can be conveniently thought of as the stable trajectories of an attractor at infinity. Therefore, the stable trajectories of the saddle at $(3.05, -0.21)$ form the stability boundary for the attractor at infinity, for one of the unstable trajectories of the saddle leaves the bounding box. Consistency rules are checked and satisfied. At the end of this step, the program finds three qualitatively different regions associated with the three attractors and internally represents the phase space structure in a data structure: the attractors are connected with each other via saddles and associated with stability boundaries (Figure 3(c)).

The second step extracts a polyhedral approximation to each stability region preserving the gross features of the shape of the region. Consider the stability region of the attractor at $(2.0, 2.0)$. The stability boundary is numerically approximated by a collection of trajectory points on the boundary (56 in total for this example), see Figure 3(d). We choose this set of points such that they are relatively uniform and dense on the boundary.

A Delaunay triangulation is performed on this set of points. As the result, the convex hull of the points is tessellated with triangles, as in Figure 3(e). Under

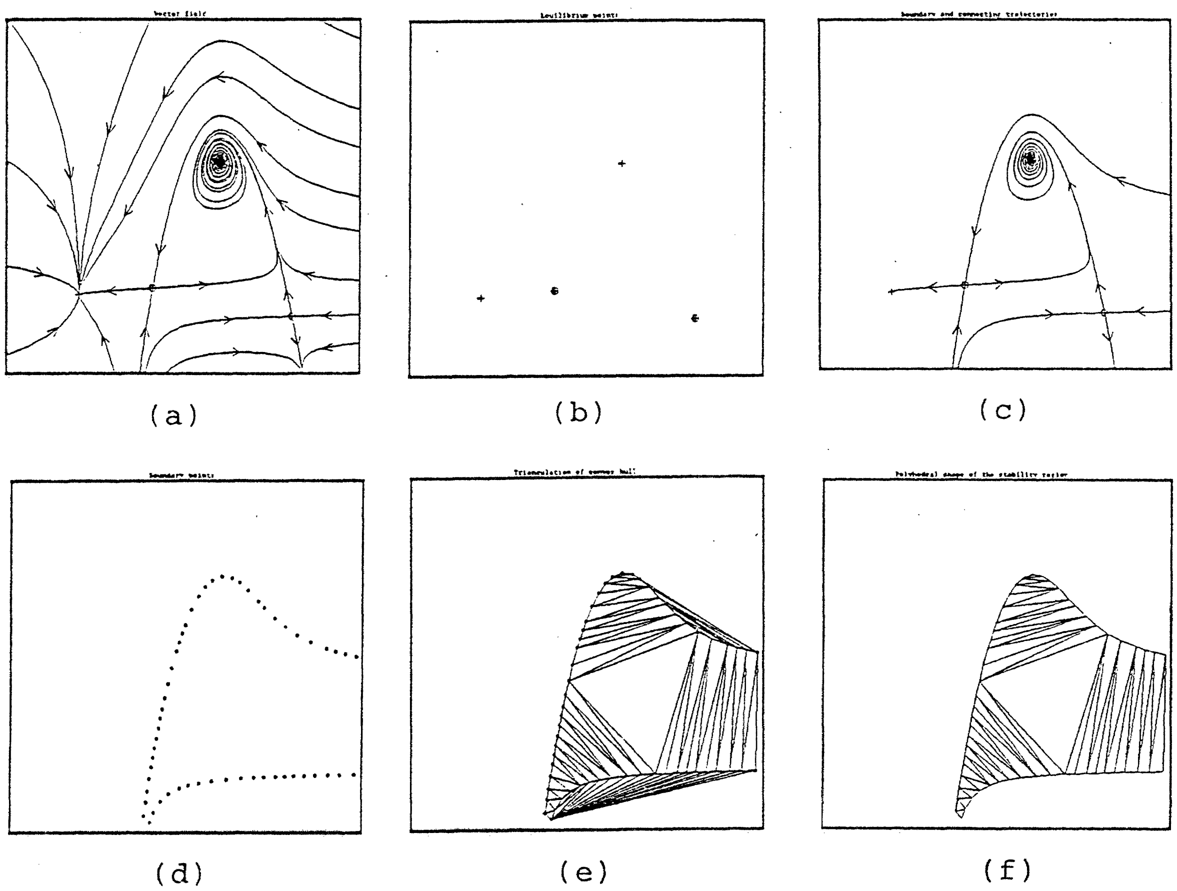


Figure 3: The analysis of a 2nd order nonlinear system: (a) vector field; (b) equilibrium points; (c) boundary and connecting trajectories; (d) points on the stability boundary for one of the attractors; (e) triangulation of the convex hull; (f) polyhedral approximation.

our condition that the boundary points are reasonably dense and uniform on the boundary, the polyhedral shape of the stability region is contained in the triangulation of the convex hull. In order to extract the polyhedron, triangles exterior to the polyhedron have to be eliminated. We note two facts. First, the circumcircles of exterior triangles are larger than the circumcircles of the interior triangles [4], since the interior triangles respect the local geometries of the boundary they approximate, whereas the exterior ones do not. Second, only certain type of triangles are the candidates for elimination: the triangles with exactly one edge and two vertices on the boundary of the convex hull in two dimensions. Therefore, we can focus the search and eliminate the exterior triangles by deleting the candidate triangle with the largest circumcircle, until the number of points on the boundary of the polyhedral approximation is the same as that of the original set of the boundary points or there is no more candidates for elimination (Figure 3(f)). We reiterate that the condition on the distribution and density of the boundary points has to be checked with respect to the shape of a region, to ensure that the circumcircle heuristic rule for elimination works. For example, more points are needed to approximate the boundary of a region with finger-like narrow parts.

The program compiles the data structure from step 1 into a relational graph, augments it with the polyhedral approximation, and reports to the user the following findings:

```
<equilibrium-points:
 1. saddle at:(3.05124921972504 -.2102498439450078)
 2. attractor at:(2.0000000000000004 1.9999999999999996)
 3. saddle at:(1.0487507802749587 .19024984394500213)
 4. attractor at:(-6.856874922766096e-15 .09523809523809473)>

<relational-graph:
 1. stability-region for attractor at:*infinity*:
   stability-boundary:
     trajectory 1:(from *infinity* to (3.05124921972504 -.2102498439450078))
     trajectory 2:(from *infinity* to (3.05124921972504 -.2102498439450078))
   connecting-trajectories:
     trajectory 3:(from (3.05124921972504 -.2102498439450078) to *infinity*)
 2. stability-region for attractor at:(2.0000000000000004 1.9999999999999996):
   stability-boundary:
     trajectory 4:(from *infinity* to (1.0487507802749587 .19024984394500213))
     trajectory 5:(from *infinity* to (1.0487507802749587 .19024984394500213))
     trajectory 1:(from *infinity* to (3.05124921972504 -.2102498439450078))
     trajectory 2:(from *infinity* to (3.05124921972504 -.2102498439450078))
```

```
  connecting-trajectories:
    trajectory 6:(from (1.0487507802749587 .19024984394500213)
                 to (2.0000000000000004 1.9999999999999996))
    trajectory 7:(from (3.05124921972504 -.2102498439450078)
                 to (2.0000000000000004 1.9999999999999996))
3. stability-region for attractor at:(-6.856874922766096e-15 .09523809523809473):
  stability-boundary:
    trajectory 4:(from *infinity* to (1.0487507802749587 .19024984394500213))
    trajectory 5:(from *infinity* to (1.0487507802749587 .19024984394500213))
  connecting-trajectories:
    trajectory 8:(from (1.0487507802749587 .19024984394500213)
                 to (-6.856874922766096e-15 .09523809523809473))>
```

2.5 Other examples

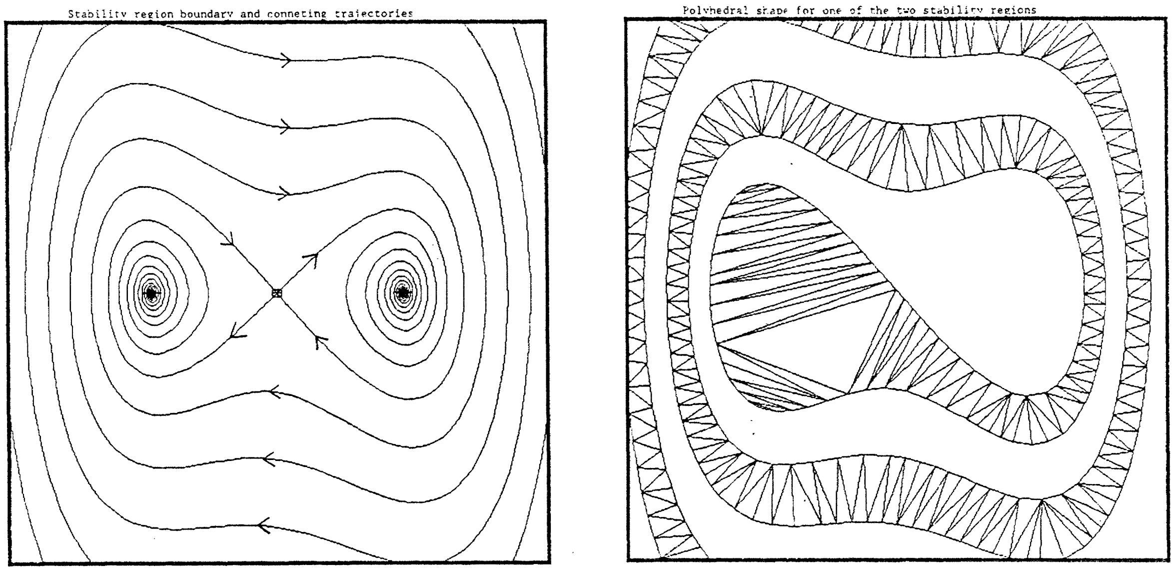
We have run the program on several other nonlinear examples either with greater complexity or of higher order.

The dynamical system for a buckling column under compressive force [11]

$$\begin{cases} x' = y \\ y' = ax^3 + bx + cy \end{cases}$$

is a 2nd order system that exhibits more complicated phase space geometries than the previous example. The dynamical model is also closely related to a pendulum suspended over two magnets. For the parameter values $a = -1.0$, $b = 2.0$, and $c = -0.2$ and the phase space region $-3.0 \leq x \leq 3.0$ and $-4.0 \leq y \leq 4.0$, the program finds two attractors at $(1.41, 0.0)$ and $(-1.41, 0.0)$ and a saddle at the origin, and generates a description about the phase space geometries: two banded stability regions associated with the two attractors, separated by the stable trajectories of the saddle at the origin. Figure 4(a) shows stability boundaries and connecting trajectories of two stability regions, and Figure 4(b) shows the polyhedral approximation to one of the region.

Since the stability regions are interleaved with each other, the geometric extraction algorithm using the circumcircle heuristic described earlier terminates before all the exterior triangles are eliminated. A more expensive procedure is then used to eliminate the remaining exterior triangles: a triangle is in a stability region of an attractor if the trajectory starting at the centroid of the triangle approaches the attractor in the limit or enters another triangle already in the stability region.



(a) (b)

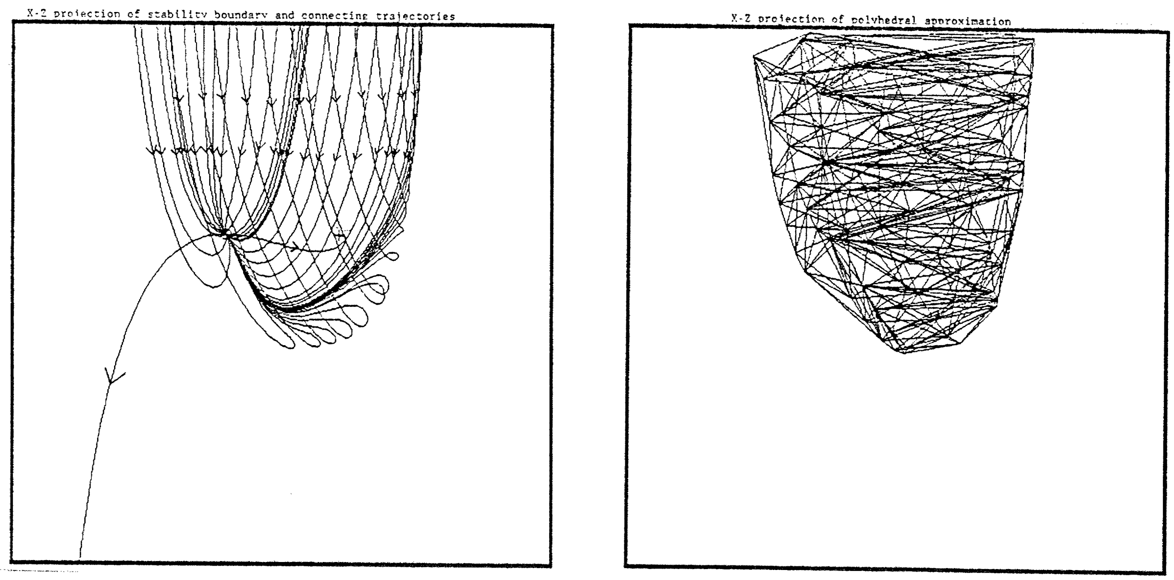
Figure 4: The analysis of a buckling column: (a) stability boundary and connecting trajectories; (b) polyhedral approximation.

Our algorithm works for higher dimensional systems as well. Consider the following 3rd order nonlinear system

$$\begin{cases} x' = y \\ y' = z - x^2 - y \\ z' = 1.0 - y^2 - 0.8z^2 \end{cases}$$

The program locates an attractor at $(1.06, 0.0, 1.12)$ and a saddle at $(-1.06, 0.0, 1.12)$ within the region $-5.0 \leq x \leq 5.0$, $-5.0 \leq y \leq 5.0$, and $-5.0 \leq z \leq 5.0$, and determines that the stable trajectories of the saddle form the stability boundary for the attractor. The stability boundary is a two dimensional surface and is approximated by a set of relatively evenly spaced trajectories. The program then tessellates the phase space with tetrahedra and extracts a polyhedral approximation to the stability region of the attractor (see Figure 5). In general, the program uses n dimensional geometric primitives — n-simplices — to approximate stability regions in dimension n.

We have also run the program on a system that is not generic and falls outside the domain of our method (see discussion in Section 4.1). The system is the earlier 2nd order example (1) in Section 2.4 with parameter value $u = 0.0$. The program



(a) (b)

Figure 5: The analysis of a 3rd order nonlinear system: (a) projection of stability boundary and connecting trajectories in x-z plane; (b) projection of polyhedral approximation in x-z plane.

proceeds to locate the equilibrium points within the region $-1.0 \leq x \leq 4.0$ and $-1.0 \leq y \leq 4.0$ and to compute the stable and unstable trajectories of saddles. It terminates with the following partial description

```
<equilibrium-points:
 1. saddle at:(3. 0.)
 2. attractor at:(2.1 1.9799999999999995)
 3. saddle at:(1. 0.)
 4. attractor at:(-1.252352859568025 7e-14 -4.157455987748001e-17)>

<saddle-connection:
    trajectory: (from saddle #(1. 0.) to saddle #(3. 0.))>
```

The program discovers a saddle connection in the course of determining the relational graph of the phase space structure: the trajectory that is both an unstable trajectory of the saddle at $(1.0, 0.0)$ and a stable trajectory of the saddle at $(3.0, 0.0)$. It concludes that the system is not generic, which also implies structural unstability for the planar system here, and abandons any further efforts to characterize the phase space structure. Since saddle connections are often precursors for chaos or structural unstability, we believe that they are important in partially characterizing the phase space structures of chaotic or structurally unstable systems.

3 Hierarchical Extraction and Representation of Phase Space Information

We have described our algorithm for analyzing a dynamical system through successive computations on the system, starting from its system equation representation. The analysis program generates a high level description of the dynamical system at the end of the analysis. To bridge the large semantic gap between the deduced symbolic description and the system equation representation of the input, we have employed multiple intermediate representations for the dynamical system, shown in Figure 6.

The program extracts the information incrementally, applying a set of operations to each intermediate representation. At each level of the representation, implicit properties (such as spatial relations) of the system at different scales are made explicit and thus can be accessed and manipulated by the operators at the

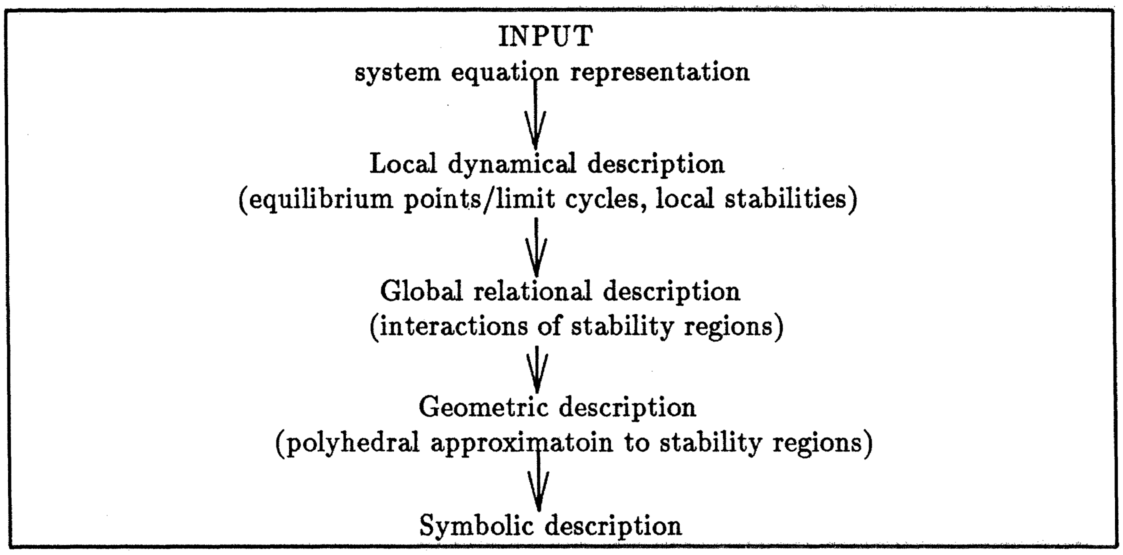


Figure 6: A multi-layered representation for a dynamical system.

next level of the representation. In the order of analysis, the program generates a local description of equilibrium points, limit cycles, and their eigenstructures, a relational description of equilibrium points, limit cycles, and their interactions, polyhedral approximations to stability regions, and a high level description of the phase space structure.

The *internal representation* of the phase space description as a relational graph captures the qualitative aspects of the phase space structure. In the relational graph, nodes are attractors and arcs denote the relations between their stability regions. Each node has information about the attractor it represents, the associated stability region and its polyhedral approximation, and the boundary trajectories and boundary equilibrium points and limit cycles. Each equilibrium point has information about its position, stability type, eigenvalues and eigenvectors, and stable and unstable trajectories.

4 Discussion

The program analyzes qualitatively different regions of nonlinear dynamical systems using knowledge about stability boundaries from dynamical system theory. It extracts the geometric information from the numerical results and represents it with a collection of geometric pieces. In this section, we discuss the class of dynamical systems to which our method applies, the further extensions to the program, and the comparison with other related work.

4.1 Scope of the method

We have stated that the theoretical basis of our algorithm for locating stability boundaries is the result of Chiang et al. [5]. Chiang et al. give a topological and dynamical characterization of the stability boundaries for a class of autonomous dynamical systems and show that the stability boundary of an attractor consists of the stable trajectories of equilibrium points and limit cycles whose unstable trajectories approach the attractor. This class of dynamical systems satisfies three conditions:

1. Hyperbolicity: All equilibrium points on stability boundaries are hyperbolic[1].

2. Transversality: The stable and unstable trajectories of equilibrium points on stability boundaries are transversal to each other.

3. Finite limits of boundary trajectories: Every trajectory on the stability boundaries approaches an equilibrium point or a limit cycle as $t \rightarrow \infty$.

Conditions (1) and (2) are generic properties that almost all dynamical systems satisfy. The example in Section 2.5 that has a saddle connection violates condition (2). Condition (3) excludes some structurally stable systems (see [10] for a definition of generic properties and [6] for discussions on structural stability of dynamical systems). Therefore, our method applies to a fairly large class of autonomous dynamical systems. We note that a periodic non-autonomous dynamical systems can be converted into an autonomous system [6] and can also be analyzed by our program.

4.2 Extensions

The theoretical basis of our approach holds regardless of dimensions, so does the geometric extraction method of our algorithm. The polyhedral approximation seems

[1]Consider the Jacobian of the vector field at an equilibrium point. If all the eigenvalues of the Jacobian have non-zero real parts, then the equilibrium point is *hyperbolic*.

a natural and economic representation for the boundary surface. We note, however, that the complexity grows exponentially with the number of dimensions. In dimensions three or higher, the stability boundary surfaces are only approximated by a set of trajectories. Obtaining a set of evenly spaced trajectories on a stability boundary surface is challenging. We are currently exploring various methods for different kinds of problems in order to best approximate curved hypersurfaces and reflect the underlying dynamics.

We have demonstrated our program on an example with interleaved phase space stability regions. Additional information about the trajectories in the stability region is used in extracting the geometric structures. Other complicated stability regions such as regions containing holes can also be tackled with this approach. More work needs to be done to catalogue region shapes with various kinds of heuristics.

We have shown that our program detected a saddle connection in a structurally unstable system. Considerable amount of work needs to be directed toward exploring robust methods for detecting saddle connections and extending the program to recognize chaotic attractors. Another possible extension is to augment the current program with a bifurcation analysis, similar to Abelson's Bifurcation Interpreter [3].

4.3 Related work

Yip has constructed a program, KAM, for automatically analyzing Hamiltonian systems with two degrees of freedom in planar phase sections [12]. The program uses techniques from computer vision to cluster point sets in phase sections and classifies phase portraits into meaningful categories. Our method applies to a large class of dissipative dynamical systems of any order in continuous phase spaces, in contrast to Hamiltonian maps on planar phase sections in KAM. Since we are interested in using our program to autonomously synthesize controllers in phase spaces, our program also extracts and represents stability regions with geometric pieces, as opposed to a point set representation in KAM.

Sacks' Poincare program analyzes planar systems through a partition algorithm on phase spaces and a bifurcation analysis on one parameter [9]. The partition algorithm is based on the properties of two dimensional flows in planar phase spaces that do not generalize to higher dimensions. Ours differs from Sacks' partition algorithm in that our method is able to analyze phase spaces of any dimensions, based on a general theoretical result on dynamical systems. Our program generates a relational graph for a dynamical system characterizing the spatial arrangement of phase space structures and containing geometric information about stability regions.

Hsu [7] developed the concept of cell space to approximate state spaces. The continuous state space along with the associated map of a system, when discretized into regular cells, becomes a cell space with a cell-to-cell map, which maps one cell to another cell. The cell-to-cell mapping method approximates the stability region of an attracting cell with a collection of cells that eventually map to that cell. The method can also be applied to continuous phase spaces. Much work remains to be done in this area. Because of the hierarchical (top-down) nature of Hsu's method, it can be integrated with our method to successively refine an approximation for a stability region.

5 Conclusions

We have developed a qualitative method for automatically analyzing phase space structures of nonlinear dynamical systems and have constructed a program to demonstrate the method. The program looks at the phase spaces, finds qualitatively different regions—the stability regions, and extracts and represents the qualitative features. It employs deep domain knowledge about dynamical system theory to recognize the qualitative structures of phase spaces. It computes a high level description about a dynamical system through a combination of numerical, combinatorial, and geometric computations and represents the phase space structure with a relational graph.

We are currently using our method to develop a novel control synthesis strategy for nonlinear control systems, with which a controller for a nonlinear system can be automatically synthesized in phase spaces. The strategy relies on the knowledge of phase spaces obtained from the analysis program, made possible by the internal representation of the phase space structures. It generates control laws by synthesizing shapes of dynamical systems and planning and navigating system trajectories in the phase spaces. More specifically, the control strategy consists of a global control path planner in phase spaces and a local trajectory generator. The global path planner finds an optimal path from an initial state to the goal state in the phase space, consisting of a sequence of path segments connected at intermediate points where control parameter changes. The high-level description of the phase space will be used to focus and prune the search for the global path. A brute-force search in high dimensional phase spaces would be prohibitively expensive. The local trajectory generator uses the flow information about the phase space regions to produce smoothed trajectories. The control synthesis program employs techniques such as phase space "surgery" operation, programmed damping and switching, and library-based approach [13]. Because of the human accessible aspect of the high level description, the qualitative analysis techniques presented in this paper can also assist engineers in designing controllers for complex systems.

6 Acknowledgments

The author would like to thank Harold Abelson and Gerald Sussman for support and many helpful discussions, Franklyn Turbak and Elisha Sacks for discussions and comments on the paper, Weiping Li for answering questions about nonlinear control engineering, John Guckenheimer for arranging for the author to visit Cornell University and to study dynamics with him during the summer of 1990, and members of MIT Project on Mathematics and Computation for developing the MIT Scheme programming language and utilities. Elisha Sacks has kindly provided us with his code for locating invariant curves of 2D maps. Elmer Hung allowed us to use his implementation of QR algorithm for computing eigenvalues.

References

[1] H. Abelson and G. J. Sussman, "The Dynamicist's Workbench I: Automatic Preparation of Numerical Experiments." In: *Symbolic Computation: Applications to Scientific Computing.* R. Grossman (ed.), SIAM, Philadelphia, PA, 1989.

[2] H. Abelson, et al., "Intelligence in Scientific Computing." *CACM*, 32(5), May 1989.

[3] H. Abelson, "Bifurcation Interpreter: A step towards the automatic analysis of dynamical systems." *Int. J. of Computers and Mathematics with Applications*, 20(8), June 1990.

[4] Jean-Daniel Boissonnat, "Geometric Structures for Three-Dimensional Shape Representation." *ACM Trans. on Graphics*, 3(4), October 1984.

[5] H. Chiang, M. W. Hirsh, and F. F. Wu, "Stability Regions of Nonlinear Autonomous Dynamical Systems." *IEEE Trans. Automatic Control*, 33(1), Jan 1988.

[6] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* Springer-Verlag, New York, 1983.

[7] C. S. Hsu, *Cell-to-Cell Mapping.* Springer-Verlag, New York, 1987.

[8] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems.* Springer-Verlag, New York, 1989.

[9] Elisha Sacks, "Automatic Analysis of One-Parameter Planar Ordinary Differential Equations by Intelligent Numeric Simulation." *CS-TR-244-90*, Princeton University, Jan 1990.

[10] S. Smale, "Differentiable Dynamical Systems." *Bulletin of the AMS*, 73, 1967.

[11] J. J. Stoker, *Nonlinear Vibrations.* Interscience, New York, 1950.

[12] K. M. Yip, "KAM: Automatic Planning and Interpretation of Numerical Experiments Using Geometrical Methods." *MIT AI-TR*-1163, August 1989.

[13] Feng Zhao, "Automatic Analysis and Synthesis of Controllers for Dynamical Systems Based on Phase Space Knowledge." *PhD Thesis Proposal*, Dept. of EECS, MIT, May 1, 1990.