Generating Global Behaviors Using Deep Knowledge of Local Dynamics*

Kenneth Man-kam Yip
MIT Artificial Intelligence Laboratory
NE 43 - 438
545 Technology Square,
Cambridge, MA 02139.

Abstract

Even with powerful numerical computers, exploring complex dynamical systems requires significant human effort and judgment to prepare simulations and to interpret numerical results. This paper describes one aspect of a computer program, KAM, that can autonomously prepare numerical simulations, and can automatically generate high-level, qualitative interpretations of the quantitative results. Given a dynamical system, KAM searches in the phase space for regions where the system exhibits qualitatively distinct behaviors: periodic, almost-periodic, and chaotic motion. KAM uses its knowledge of dynamics to constrain its searches. This knowledge is encoded by a grammar of dynamical behavior in which the primitives are geometric orbits, and in which the rules of combination are orbit adjacency constraints. A consistency analysis procedure analogous to Waltz's constraint satisfaction algorithm is implemented. The utility of the approach is demonstrated by exploring the dynamics of nonlinear conservative systems with two degrees of freedom.

1 Introduction

From the study of stars and galaxies formation, to aircraft and wing design, to blood flow in the heart, and to microelectronics device modeling, scientists and engineers are confronted with nonlinear equations that are inaccessible to analytical methods. Although powerful numerical computers can painlessly generate quantitative particular solutions to the equations, understanding the qualitative content of these equations requires substantial human effort and judgment to prepare numerical simulations, and interpret numerical results in qualitative terms.

This paper demonstrates that by combining ideas from symbolic computation with methods incorporating deep knowledge of the underlying dynamics of the physical domain, one can build an effective program that autonomously explores the behaviors of a specified dynamical system, and generates high-level, qualitative interpretation of the quantitative data.

*This report 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 N00014-86-K-0180.

Exploring a dynamical system involves two tasks: (1) generate all typical responses of the system by sampling a sufficient number of starting conditions, and (2) describe how these typical responses change their characters as the system parameters are varied.

Studying the swing of a simple pendulum provides a good illustration of these tasks. An experimenter observes how the pendulum starting at various initial states – angular position and velocity of the blob – swings. He may repeat the experiments by controlling the surrounding conditions such as air resistance and gravity. From the experimental results, he wishes to answer qualitative questions like the following: What are the steady states? Are they stable? Under what starting conditions will the pendulum oscillate? Will it rotate? Will it swing chaotically? In short, one wants to know the typical responses of the system without actually solving the equations governing the pendulum motion.

Blind exhaustive testing of every point of the phase space – the space of all possible starting conditions – to find out the system responses is out of the question because the number of possible starts is overwhelming. Moreover, interesting qualitative changes in behavior often occur in so small a region in the phase space that unguided experiments will likely miss them.

One way to meet these difficulties is to exploit knowledge of the natural constraints in the physical problem. Fluid flow provides a useful illustration. Fig. 1 depicts a flow pattern in some small region; it shows four flow lines. The flow pattern, as it stands, is not complete: some important feature is missing. Let us see why. Since the horizontal flow lines are going in opposite direction, there must be a line of fluid particles whose velocity vectors have zero horizontal components. A similar argument about the vertical flow lines shows there must be a line of fluid particles whose velocity vectors have zero vertical components. In general, these two lines of fluid particles will intersect transversely. The point of intersection, which has a zero velocity vector, is known as a **stagnant point.** [1] With the stagnant point added, the local flow pattern becomes consistent.

Now, we want to turn this physical insight around, and use knowledge about possible local flow patterns to derive an understanding of the global behaviors of a physical system.

KAM [2] is an implemented program that embodies this type of knowledge. KAM describes dynamical behaviors

[1] This argument can be made rigorous using the Index Theorem on singular points of vector fields [Arnold, 1973].

[2] The initials stand for three mathematicians: Kolmogorov, Arnold, and Moser.

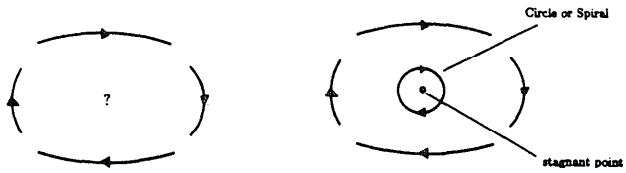


Figure 1: *Left: Inconsistent Flow Lines. Right: With additional flow lines and a stagnant point, flow becomes consistent.*

with a geometric language. [3] An **orbit** in the geometry world represents a particular response in the physical world. The totality of the system behavior is represented in KAM by an **Orbit Adjacency Graph**. A node of the Orbit Adjacency Graph represents an orbit; an arc linking two nodes stands for an adjacency relation between two orbits.

KAM exploits two types of domain knowledge. [4] First, KAM classifies all possible orbits into a few geometric categories such as a periodic orbit, a closed orbit, an island chain, and a chaotic region. Second, KAM has a list of *orbit consistency rules*; these rules impose adjacency constraints on neighboring orbits.

One way to look at KAM is to contrast its consistency analysis with Waltz's constraint analysis in line drawings [Waltz, 1975]. In Waltz's problem, the junction labels are ambiguous, but the connections among neighboring junctions are known. In KAM's problem, the orbit interpretations are unambiguous, but the connections among neighboring orbits are not known. Whereas Waltz's constraint analysis seeks to eliminate impossible line interpretations, KAM's consistency procedure looks for new orbits to eliminate impossible orbit connections. Waltz's procedure starts with a complete network of connections. KAM, on the other hand, builds the connection network incrementally. A common theme in both works is the emphasis on finding the correct categorization of primitive objects that interact in a well-defined and tightly constrained way.

KAM has three parts: (1) an **Orbit Recognition Module** to classify dot patterns, (2) a **Phase Space Searching Module** to explore the space of initial states, and (3) a **Parameter Space Searching Module** to explore the space of parameter values. This paper describes only one aspect of KAM, namely, how it explores the phase space for a fixed parameter value. A previous paper [Yip, 1987] shows how orbit recognition is done by using minimal spanning trees. The problem of searching in the parameter space, and its solution will be described in a forthcoming dissertation [Yip, 1988]. KAM is implemented in Zetalisp; all the examples in the rest of this paper are actual output of the KAM program.

The current work complements recent research in qualitative physics [Bobrow, 1985] in two directions. First, it studies a new domain, a nontrivial class of dynamical systems having essential nonlinearity, which brings up hard issues of representing complicated qualitative behaviors: multiple steady states, periodic, almost-periodic, and chaotic motion.

Second, the complexity of behavior necessitates a style of qualitative reasoning that emphasizes the overall *shape* of the motion rather than the details of an individual quantity's variation with time. So, while almost all works in qualitative physics are based on some form of qualitative calculus and time-series representation of time-varying behaviors, this research employs a geometric representation of behavior that is based on the Dynamical Systems Theory [Smale, 1967]. An orbit representing a particular system behavior is a static, unchanging object which can be easily visualized, and be compared and contrasted with other orbits.

The paper is organized as follows. The next section defines some terminologies. Section 3 describes the target domain; it introduces a nonlinear discrete mapping which serves as an example throughout the paper. Section 4 examines the constraints imposed by the problem domain. Section 5 presents KAM's consistency analysis algorithm. Section 6 shows KAM's performance for a particular parameter value of the mapping. Finally, section 7 explains the performance and discusses the limitations of the current implementation.

2 Ontology

To begin with, we must understand something of dynamical systems.

The **state** of a system at any time t_0 is a minimum set of values of variables $\{x_1, \ldots, x_n\}$ which, along with the input to the system for $t \geq t_0$, is sufficient to determine the behavior of the system for all time $t \geq t_0$. The variables which define the state are called **state variables**. The set of all possible states of a system is called its **phase space**.

A dynamical system consists of two parts: (1) the system state, and (2) the evolution law. As the system evolves with time, the state traces out a path in the phase space; the path is called an **orbit** or a **trajectory**. A **phase portrait** is a partition of the phase space into orbits.

The **evolution law** determines how the state evolves with time. In a discrete time system, the evolution law is given by a system of difference equations, or, more abstractly, a function $f : X \rightarrow X$ where X is the phase space of the discrete system. The function f which defines a discrete dynamical system is called a **mapping**, or a **map**, for short. The **multipliers** of the map f are the eigenvalues of the Jacobian of f. An **area-preserving map** is a map whose Jacobian has a unit determinant. In the evolution law, variables that are not state variables are called **parameters**. A **one-parameter family of maps** $\{ f_\alpha \}$ is a class of maps indexed by the same parameter α.

The set of iterates of f, $\{ f^n(x) \mid n \in Z \}$, is called the **orbit of x relative to f**. The point x is a **fixed point** of f if $f(x) = x$. A fixed point x is **stable**, or **elliptic**, if all the multipliers of f at x lie on the unit circle; it is **unstable**, or **hyperbolic**, otherwise. The point x is a **periodic point** of period n if $f^n(x) = x$. The least positive n for which

[3] Describing dynamical phenomena geometrically is a great insight due to Henri Poincaré.

[4] The domain knowledge is based on deep mathematical facts; it does not come from *ad hoc* stipulations.

$f^n(x) = x$ is the period of x. The set of all iterates of a periodic point forms a periodic orbit.

Although differential equations are the most common tool for modeling continuous phenomena, I choose to study difference equations for two reasons. First, in practice, iterating maps is far easier than integrating differential equations; one can generate data a thousand times faster. Second, mathematically difference equations exhibit the same class of behavior as differential equations [Smale, 1967]; so there is no loss of generality. The price, however, is that it is sometimes not obvious what physical situation a given set of difference equations corresponds to.

3 The Task

KAM explores dynamical systems characterized by a one-parameter family of nonlinear area-preserving maps. This class of problems is important because many problems in physics – the restricted 3-body problem, orbits of particles in accelerators, and two coupled nonlinear oscillators, for instance – can be reduced to the study of area-preserving maps.

KAM takes three inputs: (1) a one-parameter family of area-preserving maps, (2) the ranges of the state variables, and (3) the range of the parameter. The output is a family of phase portraits. Each phase portrait is partitioned into regions belonging to one of the three types: first, a regular region containing periodic or almost-periodic orbits [5]; second, a chaotic region containing chaotic or unbounded orbit; third, an intermediate region between the first two consisting of large island chains. The family of phase portraits records the history of qualitative changes in the phase space structure as the parameter is varied.

Let us focus on describing how one phase portrait is produced for a specified parameter value. Consider a typical area-preserving map studied by Henon [Henon, 1969]:

$$\begin{aligned} x_{n+1} &= x_n \cos\alpha - (y_n - x_n^2)\sin\alpha \\ y_{n+1} &= x_n \sin\alpha + (y_n - x_n^2)\cos\alpha \end{aligned}$$

where $x_n \in (-1\ 1), y_n \in (-1\ 1)$ and $\alpha \in (0\ 2.2)$.

The Henon map is historically important because it shows how an addition of a simple quadratic nonlinearity (via x_n^2) to a linear rotation can lead to dramatic changes in the behavior of system. Fig. 2, taken directly from Henon's paper, displays the phase portrait of the Henon Map for a particular parameter value $\alpha = 1.3284305$ ($\cos\alpha = 0.24$).

Fig. 2 depicts nine representative orbits of the phase portrait. Near the elliptic fixed point at origin is a regular region consisting of three closed orbits. Just inside the outermost closed orbit lies two chains of five smaller closed curves. Islands of the chains are separated by a separatrix, the closed curve with five loops. Finally, outside the regular region is a chaotic orbit whose iterates no longer lie on a curve, but seem rather to fill a two-dimensional region in a chaotic manner.

KAM's task is to search the phase space for such representative orbits. Note in particular the 5-loop separatrix enclosing the island chain. Because the separatrix is confined to such a small region of the phase space, the chance of finding this orbit by random selection of initial states is almost zero. Later in the paper, we will see how KAM can discover the separatrix in a few trials.

[5] See section 4.1 for the relevant definitions.

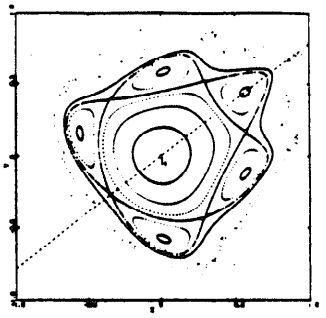


Figure 2: *A Typical Phase Portrait of the Henon Map ($\cos\alpha = 0.24$). Prominent features: an elliptic fixed point at origin, two large 5-island chains, a separatrix, and chaotic region.*

4 Characteristics of Task Domain

4.1 Primitive Orbits

A fundamental property of conservative (Hamiltonian) systems with two degrees of freedom is the area-preserving property: a bundle of initial points covering a small region is mapped onto another region with the same area. The shape of the region may change, but the area is invariant under the map. This property severely limits the possible long-time behaviors of orbits.

In the following, I enumerate the six possible ways in which the iterates of an area-preserving map can appear in the phase space [6] (see Fig. 3):

1. **Periodic Orbit.** A finite number of N iterates are encountered repeatedly.
2. **Almost-periodic Orbit.** The iterates fill densely a smooth closed curve surrounding a elliptic fixed point, but they never repeat themselves.
3. **Island Chain.** The iterates form almost-periodic orbits around a stable periodic orbit. The number of islands (or loops) in the chain is equal to the period of the periodic orbit enclosed by the chain.
4. **Separatrix.** A separatrix is an orbit joining hyperbolic periodic points. The number of loops of a separatrix is equal to the period of its associated periodic orbit.
5. **Chaotic Orbit.** The iterates form a random splatter of points that fills up some area of the phase space.
6. **Escape Orbit.** The iterates approach arbitrarily large values in the phase space; they form an unbounded orbit.

[6] The categorization depends on two mathematical results: (1) orbits in the phase space do not cross, and (2) the phase space of a Hamiltonian system with two degrees of freedom is diffeomorphic to a solid 2-torus. See [Arnold, 1978].

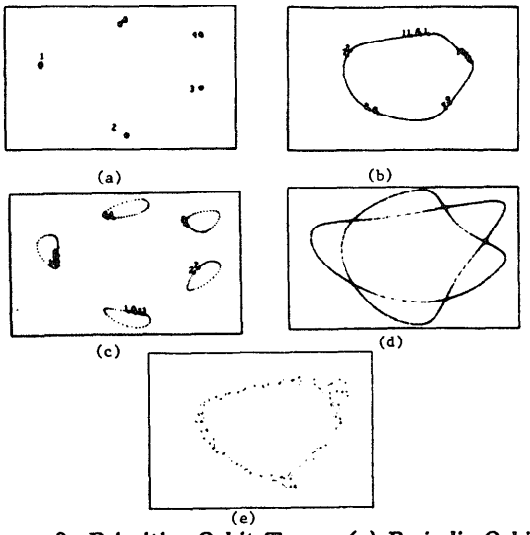


Figure 3: *Primitive Orbit Types: (a) Periodic Orbit (period 5 is shown), (b) Almost-periodic orbit, (c) Island chain (a 5-island chain is shown), (d) Separatrix (a 5-separatrix is shown), (e) Chaotic Orbit (An escape orbit looks the same except eventually becomes unbounded.)*

4.2 Continuity of Rotation Number

As suggested in the introduction, two neighboring flow lines must satisfy a continuity requirement: a continuous flow cannot have a sudden change in flow direction. There is a direct analog for discrete maps. An orbit of a discrete map is no longer a smooth line, but consists of points jumping from one place to another. The crucial quantity in the discrete case turns out to be the **rotation number** of an almost-periodic orbit.

The rotation number measures the asymptotic average of the angular distances between any two successive iterates in units of 2π-radian [MacKay, 1982]. A periodic orbit can be thought of as a degenerate almost-periodic orbit whose rotation number is rational. For example, the iterates of an orbit with rotation number $= \frac{1}{5}$ repeat themselves after every five times. An almost-periodic orbit always has an irrational rotation number.

The concept of rotation number is important because it can be shown [MacKay, 1982] that *the rotation numbers of nearby almost-periodic orbits change continuously.* We can exploit the continuity of rotation number to locate periodic orbits and island chains as follows. Consider two nearby almost-periodic orbits having rotation numbers ρ_1 and ρ_2 respectively. Suppose ρ_1 is slightly smaller than $\frac{1}{5}$, and ρ_2 slightly larger. By continuity, there must exist a third, nearby orbit with rotation number exactly equal to $\frac{1}{5}$. In other words, a periodic orbit of period 5 must exist between the two almost-periodic orbits. [7]

[7] Since there are infinitely many rationals between any two irrationals, infinitely many periodic orbits exist. However, the size of the island chain enclosing a periodic orbit decreases exponentially as the period of the periodic orbit increases [Arnold, 1978]. So, in practice, periodic orbits with large periods will be too tiny to be seen.

The rotation number of an almost-periodic orbit is represented by an open interval with rational endpoints. For example, $\rho \in (p\ q)$ means the rotation number ρ lies somewhere between the reduced rationals p and q. [8] Two rotation numbers are said to be **compatible** if their associated open intervals are not disjoint. An integer k is compatible with a rotation number if k is equal to the denominator of one of the endpoints of the associated open interval.

5 Implementation

5.1 Pairwise Orbit Consistency

The key strategy underlying KAM's search for orbits in the phase space is to focus the search in regions where neighboring orbits are inconsistent. This section describes KAM's rules for determining pairwise orbit consistency.

As explained before, KAM recognizes six primitive orbit types. Considering consistency, we can make two simplifications. First, a periodic orbit is thought of as a degenerate almost-periodic orbit with rational rotation number. Second, neither a chaotic orbit nor an escape orbit constrains its neighboring orbits. With three orbit types – almost-periodic, island chain, and separatrix – the number of possible pairwise combinations is 6. The inconsistent combinations are described by the rules below [9] :

- RULE 1: **Missing Islands.** Two almost-periodic orbits with incompatible rotation numbers are not consistent.
- RULE 2: **Missing Separatrix.** An island chain and an almost-periodic orbit are not consistent: a separatrix is missing.
- RULE 3: **Missing Almost-periodic.** Two island chains with different number of islands are not consistent.
- RULE 4: **Missing Orbit-1.** A separatrix with k loops is inconsistent with an almost-periodic orbit having rotation number ρ unless k is compatible with ρ.
- RULE 5: **Missing Orbit-2.** A separatrix with k loops is inconsistent with an island chain with m islands unless k = m.
- RULE 6: **Missing Orbit-3.** Two separatrices having different number of loops are not consistent.

5.2 Consistency Complaints

When two neighboring orbits are inconsistent, a complaint will be made. A **consistency complaint** is a data structure describing the nature of the complaint. Specifically, it records the type of complaint, and the identity of the orbits involved.

5.3 The Basic Algorithm

The basic data structure is an Orbit Adjacency Graph. The graph has a single type of node, and a single type of

[8] A reduced rational is a rational whose numerator and denominator have no prime factor in common.

[9] These rules have not been proven valid, but they are consistent with what is rigorously known about Hamiltonian dynamics.

link. Each node in the graph represents an orbit. A link between two nodes is **valid** if the orbits in question are adjacent in the phase portrait. A link is **inconsistent** if the adjacency cannot be part of a legal flow pattern allowed by local dynamics.

Consistency analysis is the process of updating adjacency links: create new links, remove invalid links, and identify inconsistent links. The process has two purposes: (1) maintain correct adjacency relations between orbits as new orbits are added, and (2) create a complaint for each inconsistent link. The complaints are stored in a stack, the **complaint-stack**. A complaint is removed from the stack if either the adjacency link causing the complaint is no longer valid, or the complaint does not lead to useful new orbits. KAM continually searches for new orbits until the complaint-stack is emptied.

The principal steps of consistency analysis are:

1. Initially, pick some random initial states. Create orbits corresponding to these states.
2. Add the newly created orbits to the Orbit Adjacency Graph.
3. Update adjacency links. Produce a list of invalid adjacency links to be removed, and a list of new adjacency links to be added.
4. For each new adjacency link to be added, run inconsistency rules against it. If the link is inconsistent, a new complaint is made. Put the complaint on top of the complaint-stack. Add the new link to the graph.
5. For each invalid adjacency link to be removed, delete, if any, its associated complaint in the complaint-stack. Remove the link from the graph.
6. Handle the complaint on top of the complaint-stack. Examine the type of complaint, and propose a list of new initial states to try.
7. Create a new goal with the suggested initial states. Make new orbits starting from these initial states.
8. Repeat the process (steps 2 to 7) until the complaint-stack is empty.

5.4 Suggestion Rules

After a complaint is made, KAM searches for the missing orbits to restore consistency. The purpose of the suggestion rules is to propose new initial states. Currently, KAM has six suggestion rules corresponding to the six inconsistency situations. The basic idea here is to do a "bisection search" in the region delimited by the two offending orbits. For example, if two almost-periodic orbits are inconsistent, the midpoints between the two set of iterates are promising initial states to try.

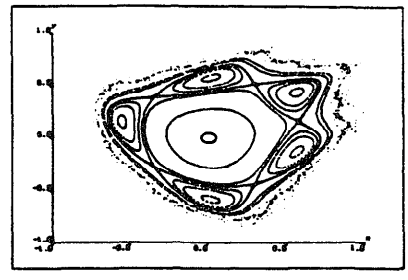
5.5 Single Orbit Consistency

Besides the pairwise orbit consistency rules, KAM has rules that act on single orbits. For example, the **Boundary Circle Rule** extends a regular region as far as possible until a boundary circle – an almost-periodic orbit that is isolated on at least one side from other almost-periodic orbits – is encountered. A second example is the **Empty Circle Rule** which says: An almost-periodic orbit that encloses a large region, and does not encloses any other orbit is not consistent.

6 Experiments

6.1 Main Illustration

To illustrate the algorithm, I give an example of how KAM explores the map with a specific parameter value $\cos\alpha = 0.24$. The total running time for this experiment is 3.5 hours. [10] KAM explores 15 initial states. Ten of these orbits are useful: 5 almost-periodic orbits, 3 island-chains, 1 separatrix, and 1 escape orbit. Fig. 4 displays these orbits; the figure is remarkably similar to Fig. 2 taken from Henon's paper. The remaining five initial states result in orbits that KAM fails to recognize. Fig. 5 shows how KAM finds the first eight orbits. Note that KAM successfully finds the separatrix at the eighth trial (Fig. 5i).



The portrait has an elliptic fixed point at (0.0 0.0).
Surrounding the fixed point is a regular region bounded by a KAM curve with rotation number between 1/5 and 1/4.
Outside the regular region lies a chain of 3 islands.
The island chain is surrounded by a KAM curve with rotation number between 4/21 and 3/26.
The outermost region is occupied by a chaotic orbit which eventually escapes.

Figure 4: *Upper: KAM finds 10 orbits in the finished phase portrait; the portrait resembles that shown in Fig. 2. Lower: KAM generates a summary report of its findings.*

6.2 Other experiments

I have tested the consistency analysis algorithm with more than 30 different values of the parameter $\alpha \in (0\ 2.2)$ of the Henon Map. KAM is able to reproduce all phase portraits like those found in Henon's paper. The algorithm has also been successfully tested on the Standard Map [MacKay, 1982] whose phase space is a torus instead of the usual Euclidean plane.

7 Evaluating the performance

7.1 What Works Well

KAM matches a domain expert in its ability to select "promising" initial states. This success can in part be attributed to the geometric representation of the orbits. When two neighboring orbits are inconsistent, the geometry of these orbits delimits a bounded region in the phase

[10] On the average, KAM takes about 10 times longer to produce one phase portrait than a domain expert does. Approximately 80 % of the running time, however, is spent on orbit recognition.

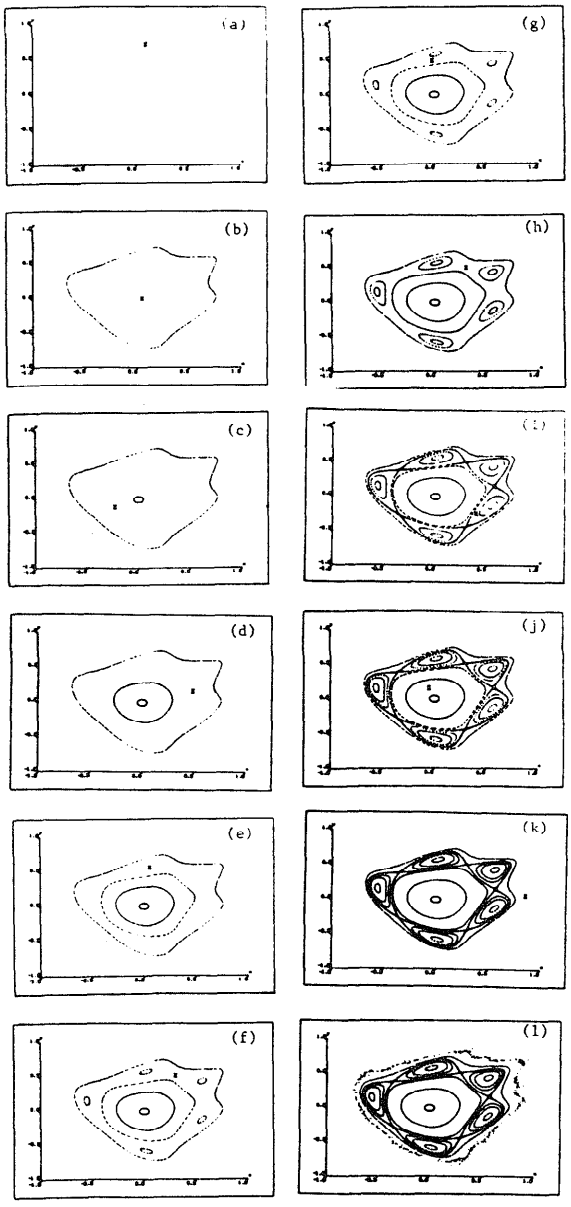


Figure 5: *How KAM picks initial conditions: (a) Randomly pick an initial state. (b) Find an almost-periodic orbit; try centroid of the orbit. (c) Find a second almost-periodic orbit; try a midpoint. (d) Find a third almost-periodic orbit; try a midpoint. (e) Find a fourth almost-periodic orbit; try a midpoint. (f) Find a 5-island chain; try a midpoint between islands. (g) No useful orbit; try a midpoint. (h) Find a second 5-island chain; try a midpoint between islands. (i) Find a 5-separatrix; try a midpoint. (j) Find a third 5-island chain; try a point in interior. (k) No useful orbit; extend boundary circle. (l) Find an escape orbit.*

space for future search. The boundedness of the region allows the "bisection search" strategy to rapidly zero in the desired orbits.

7.2 What Does Not Work As Well

Island chains that are embedded inside a chaotic (or escape) region are difficult to find. This is expected because a chaotic (or escape) orbit imposes no constraint on its neighboring orbits. A domain expert has less trouble with this problem largely because he can recognize the internal shape of a chaotic region. KAM, in contrast, cannot describe the number of holes and connected components that a given two-dimensional dot pattern may have.

8 Conclusion

This paper shows how a computer program, KAM, that exploits specific domain knowledge based on rigorous mathematical results is able to automatically generate phase portraits that are essentially the same as those appearing in published papers in the experimental dynamics literature. Knowledge of the six primitive types of long-time behavior, and the modes of transition between primitive types allows a simple consistency analysis to draw global conclusions about the phase space dynamics without requiring sophisticated problem-solving mechanisms. Describing behavior geometrically makes high-level, qualitative interpretations of numerical data possible. I expect this method of analysis – defining the primitive categories, and cataloging possible transitions – pioneered by Waltz in the early seventies, and the emphasis on geometric structures to be a powerful methodology for attacking hard computational problems in science and engineering.

References

[Arnold, 1973] V.I. Arnold. *Ordinary Differential Equations.* MIT Press, 1973.

[Arnold, 1978] V.I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, 1978.

[Bobrow, 1985] Daniel G. Bobrow, editor. *Qualitative Reasoning about Physical Systems.* M. I. T. Press, 1985.

[Henon, 1969] M. Henon. Numerical study of quadratic area-preserving mappings. *Quarterly of Applied Mathematics*, 27, 1969.

[MacKay, 1982] Robert S MacKay. *Renormalization in Area Preserving Maps.* PhD thesis, Princeton University, 1982.

[Smale, 1967] Stephen Smale. Differentiable dynamical systems. *Bulletin American Mathematical Society*, 73, 1967.

[Waltz, 1975] David Waltz. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision.* McGraw-Hill, 1975.

[Yip, 1987] Kenneth Yip. Extracting qualitative dynamics from numerical experiments. In *Proceedings AAAI-87*, 1987.

[Yip, 1988] Kenneth Yip. *KAM: A Computer Program that Autonomously Explores Dynamical Systems.* PhD thesis, MIT, 1988. Forthcoming.