Computers Math. Applic. Vol. 20, No. 8, pp. 13–35, 1990
Printed in Great Britain.

0097-4943/90 $3.00 + 0.00

THE BIFURCATION INTERPRETER: A STEP TOWARDS THE AUTOMATIC ANALYSIS OF DYNAMICAL SYSTEMS

H. Abelson

Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 62139, U.S.A.

(*Received* 9 *October* 1989)

Abstract—The Bifurcation Interpreter is a computer program that autonomously explores the steady-state orbits of one-parameter families of periodically-driven oscillators. To report its findings, the Interpreter generates schematic diagrams and English text descriptions similar to those appearing in the science and engineering research literature. Given a system of equations as input, the Interpreter uses symbolic algebra to automatically generate numerical procedures that simulate the system. The Interpreter incorporates knowledge about dynamical systems theory, which it uses to guide the simulations, to interpret the results and to minimize the effects of numerical error.

1. INTRODUCTION

Most of the effort in scientific computing concerns programs that produce numerical output, although there is also much interest in developing graphical rendering techniques that help people to visualize this output [1]. But whether a numerical experiment generates numbers or pictures, there remains the task of interpreting the results—to distill the numerical output into high-level, often qualitative descriptions that can be summarized, reasoned about and used to guide new experiments. This paper illustrates how such interpretations can be created automatically, with appropriate combinations of numerical and symbolic processing.

The Bifurcation Interpreter is a computer program that autonomously explores the steady-state orbits of one-parameter families of periodically-driven oscillators. To report its findings, the Interpreter generates schematic diagrams and English text descriptions similar to those appearing in the science and engineering research literature.

Section 2 of this paper illustrates reports generated by the Interpreter and compares these with similar reports published by dynamicists to describe their own investigations of nonlinear systems. Section 3 summarizes the range of phenomena that the Bifurcation Interpreter can observe, and describes the internal data structures it maintains to keep track of its observations. Section 4 is a detailed description of the methods used by the program. The paper ends by discussing the limitations of the present implementation and describes directions for improvement.

2. QUALITATIVE BEHAVIOR OF DYNAMICAL SYSTEMS

A periodically-driven dissipative nonlinear oscillator typically admits a finite number of periodic orbits, where the period of each orbit is an integer multiple of the drive period.† This multiple is called the *order* of the orbit. In a parametrized family of oscillators, smooth variations in the parameters give rise to families of orbits that also vary smoothly, except at certain critical values of the parameters at which there occur *bifurcations*, or discontinuous changes in orbital behavior. Depending on the type of the bifurcation, a family of orbits may vanish, change order or merge with other families, or new families of orbits may be born. Families that meet at a bifurcation are said to be in the same *class*.

The geometric theory of dynamical systems focuses on the evolution of steady-state orbits through bifurcations as a framework for capturing the qualitative behavior of dynamical systems.

†In addition, nonlinear oscillators may have nonperiodic steady-state orbits. These include *quasi-periodic orbits*, which have discrete-frequency spectra, but not at rational multiples of the drive frequency; and *chaotic* orbits, which, loosely speaking, are steady-state orbits that are neither periodic nor quasi-periodic.

A major achievement of the theory has been to show that for one-parameter families, at least, the bifurcations typically encountered can be classified into a small number of recognized types; the type of the bifurcation determines, up to homeomorphism, the behavior of the family near the bifurcation point.†

2.1. A sample report by the Interpreter

Given a one-parameter family of periodically-driven oscillators, the Bifurcation Interpreter identifies families of stable periodic orbits and classifies the bifurcations through which they evolve. The Interpreter's report catalogues the classes of orbits and the types of the bifurcations.

As an example, the following form of Duffing's equation,

$$\ddot{x} + k\dot{x} + x^3 = p \cos t,$$

describes a one-parameter family of nonlinear oscillators for each fixed value of k. This family, for $k = 0.1$, was presented to the Bifurcation Interpreter for analysis by specifying the equation (as a system of two first-order equations), a varying parameter and parameter interval, and a domain in state-space outside of which orbits should be considered to be unbounded:

- system: $\dot{x}_1 = x_2, \quad \dot{x}_2 = p \cos t - kx_2 - x_1^3$ (1)
- fixed parameters: $k = 0.1$
- varying parameter: p, 1 to 25
- bounds on state variables: x_1, -5 to 5; x_2, -10 to 10.

The Bifurcation Interpreter's output is an automatically-generated textual report that presents the results of the analysis using the technical terms of the dynamics literature. (The meanings of these terms are explained in Section 3 below.) Here is an excerpt:

> The system was explored for values of p between 1 and 25, and 10 classes of stable periodic orbits were identified.
>
> Class A is already present at the start of the parameter range $p = 1$ with a family of order-1 orbits A_0. Near $p = 2.287$, there is a supercritical-pitchfork bifurcation, and A_0 splits into symmetric families $A_{1,0}$ and $A_{1,1}$, each of order 1. $A_{1,0}$ vanishes at a fold bifurcation near $p = 3.567$. $A_{1,1}$ vanishes similarly.
>
> Class B appears around $p = 3.085$ with a family of order-1 orbits B_0 arising from a fold bifurcation. As the parameter p increases, B_0 undergoes a period-doubling cascade, reaching order 2 near $p = 4.876$, and order 4 near $p = 5.441$. Although the cascade was not traced past the order 4 orbit, there is apparently another period-doubling near $p = 5.52$, and a chaotic orbit was observed at $p = 5.688$.
>
> ⋮
>
> Class D appears around $p = 5.642$ with a family of order-3 orbits D_0 arising from a fold bifurcation. Near $p = 6.691$, there is a supercritical-pitchfork bifurcation, and D_0 splits into symmetric families $D_{1,0}$ and $D_{1,1}$, each of order 3. Near $p = 7.992$, there is a supercritical-pitchfork bifurcation where $D_{1,0}$ and $D_{1,1}$ merge to form a family D_2 of order 3. Near $p = 9.677$, there is a transcritical bifurcation where D_2 vanishes by exchanging stability with family D_3 of order 3. D_3 vanishes at a bifurcation of undetermined type near $p = 9.858$.
>
> ⋮
>
> Class J appears around $p = 23.96$ as a family of order-5 orbits J_0 arising from a fold bifurcation. J_0 is present at the end of the parameter range at $p = 25$.

In addition to the text report, the Interpreter can generate diagrams, such as Fig. 1, that present its findings in schematic form.‡ As shown in Fig. 1, the horizontal axis represents the parameter range, and ticmarks indicate the values where bifurcations occur. The classes are labeled and arrayed vertically to show the parameter extent over which orbits in each class occur. For example, for values of p between $p_3 \approx 3.1$ and $p_4 \approx 3.6$, the system has four coexisting steady-state orbits (two in class A, one in B, and one in C); each system trajectory will evolve to one of these four orbits, depending upon the initial conditions. Each bifurcation type is indicated by a standard

†Guckenheimer and Holmes [2] present the mathematical foundations of bifurcation theory. Thompson and Stewart [3] provide an introduction to dynamical systems theory for scientists and engineers.

‡The graphical-output routines used in the Interpreter program were implemented by Ognen Nastov, under the auspices of the MIT Undergraduate Research Opportunities Program.

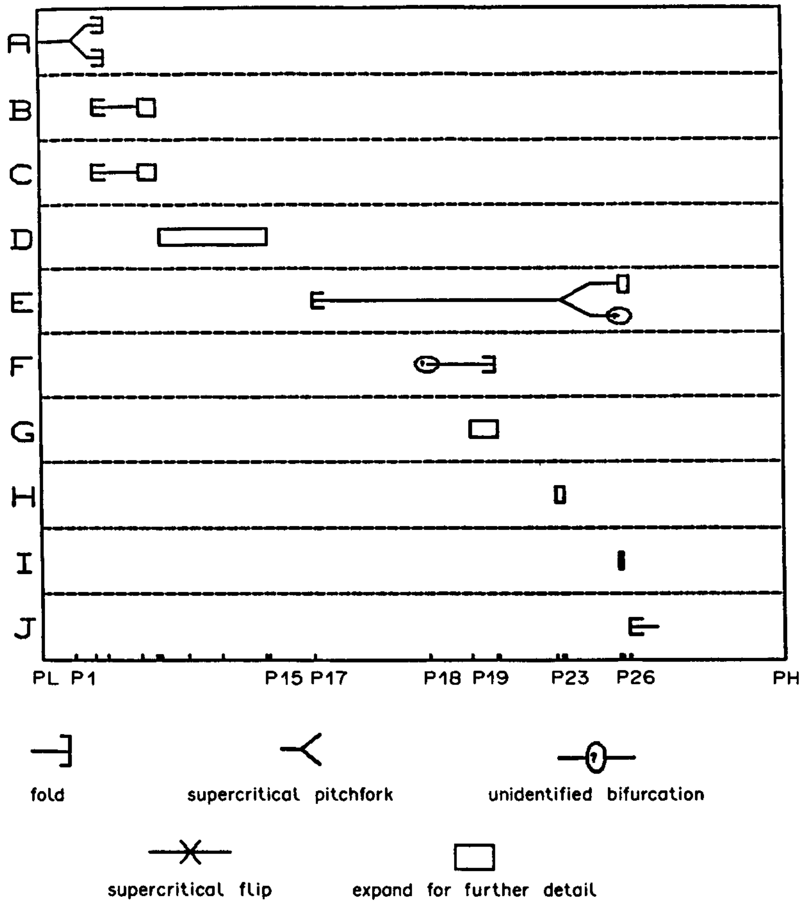

Fig. 1. This diagram, generated automatically by the Bifurcation Interpreter, shows the analysis of the system described by equation (1) for $k = 0.1$. The Interpreter has traced the evolution of 10 classes of families of periodic orbits and their bifurcations. Each bifurcation type is identified by a standard symbol—the key indicates the Interpreter's symbols for fold bifurcations, supercritical-pitchfork bifurcations and for bifurcations that the Interpreter has failed to identify. The ▭ indicate parameter intervals over which the figure should be expanded to reveal more detail, as shown in Figs. 2 and 3. The p values along the horizontal axis indicate the parameter values at which the bifurcations occur.

symbol: —] indicates a fold, the "Y" a pitchfork, and —(?)— a bifurcation that the Interpreter was unable to identify. The ▭ denote parameter intervals over which the diagram should be expanded in order to reveal further details. Figure 2 shows this expansion for the indicated regions in classes B and C, and Fig. 3 shows details of class D.

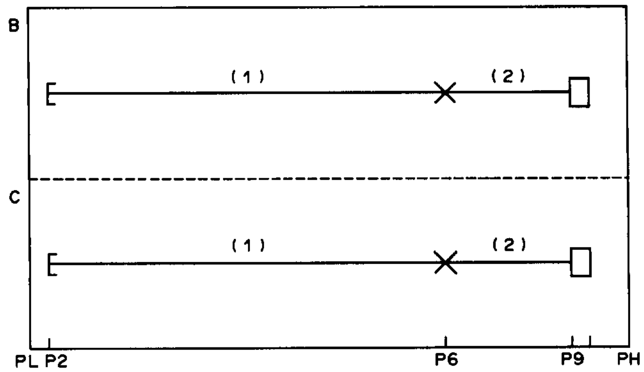

Fig. 2. Expanding the interval from $p = 3$ to $p = 5.7$ shows details of classes B and C from Fig. 1. Each class begins as order 1, then doubles to order 2 via supercritical-flip bifurcations. Expanding the interval indicated by the rectangles would show further doubling. The Interpreter recognizes this progression as a period-doubling cascade, and reports it as such.

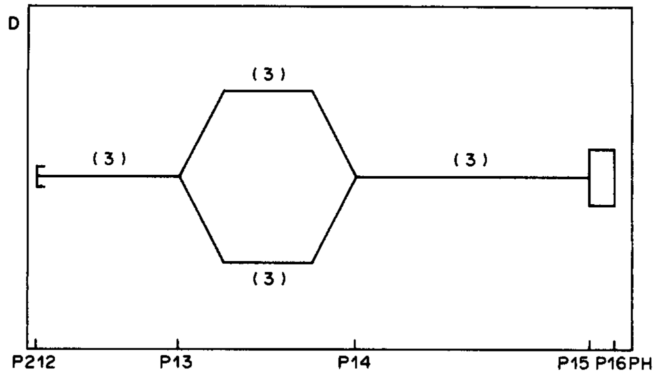

Fig. 3. Expanding the interval from $p = 5.6$ to $p = 10$ shows details of the class D of order-3 orbits. Starting as a single family, the class splits into a pair near $p = 6.7$ and these merge again near $p = 8$.

Observe that the Interpreter's textual report includes information that is not reflected in the diagrams. For instance, the Interpreter has recognized the succession of bifurcations in class B as a period-doubling cascade, and noted that the existence of the cascade is consistent with its observation of a chaotic orbit at a higher value of the parameter.

2.2. *Comparison with manually-produced reports*

The diagrams and the text generated by the Bifurcation Interpreter are similar to accounts that scientists and engineers have published to document their investigations of nonlinear systems. One example, shown in Fig. 4, is a diagram presented by Franceschini [4] to report the results of a series of simulation studies of fluid flow, modeled using a one-parameter family of nonlinear equations parameterized by the Reynolds number. Franceschini describes his diagram as a "graphical summary of the phenomenology" of the model, and compares diagrams for different models to determine whether they exhibit the same behavior.† Although the detailed layout of the diagram is different from those generated by the Bifurcation Interpreter, the information presented is much

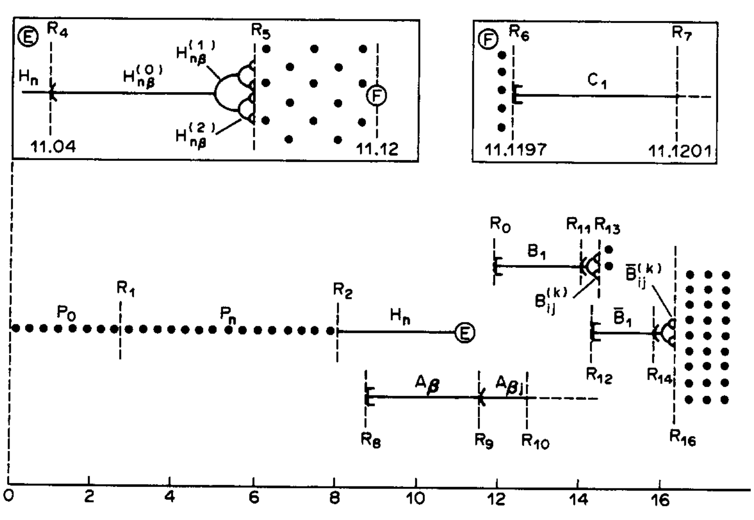

Fig. 4. This diagram, reproduced from Ref. [4], is a manually-produced report similar to the diagrams that the Bifurcation Interpreter generates automatically.

†The Navier–Stokes equations for fluid flow are a system of partial differential equations. Expanding the solution as a Fourier series and truncating the expansion at a fixed order reduces this to a system of ordinary differential equations. One approach to studying the Navier–Stokes equations is to study these finite-mode approximations in the hope that their solutions will have the same qualitative properties as those of the full equations. Franceschini's experiments [4] test the validity of this approach by comparing the qualitative properties of the 8-mode and 9-mode models. He finds that the two models have almost completely different phenomenologies, although both can exhibit turbulence at low Reynolds numbers.

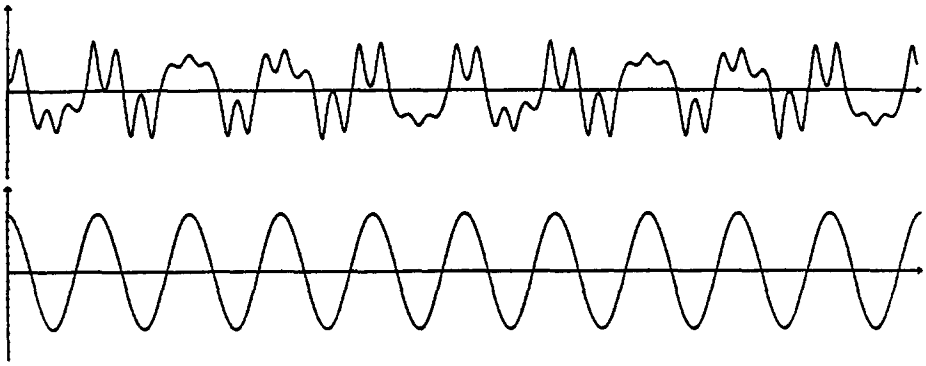

Fig. 5. Shown here and in Fig. 6 are classes of steady-state orbits automatically reported by the Bifurcation Interpreter for the system in equation (1) that are missing from a well-known published catalogue of orbits for this system [5]. Plotted here are $x(t)$ and the drive for an order-5 orbit from class J of Fig. 1 ($k = 0.1, p = 24.25$).

the same.† Incidentally, since the Interpreter generates its diagrams from a symbolic data structure that it maintains internally (described in Section 3.2 below), it should not be difficult to extend the Interpreter to automatically contrast the qualitative behaviors of two different systems, as Franceschini does in his paper.

The Interpreter's reports are based upon numerical experiments conducted at finite resultion, so it is possible that the program may fail to observe orbits that exist only over small parameter intervals or small regions of state space. On the other hand, experiments conducted by people suffer the same criticism. Indeed, one can check the quality of the Interpreter's observations of the system in equation (1) by comparing them with an often-cited study published by Ueda [5], who has catalogued the steady-state solutions of this system over the range $0 < k < 0.8, 0 < p < 25$ by means of extensive simulations with analog and digital computers. In a series of tests with various values of k, the Interpreter identified all the orbits in Ueda's catalogue and sometimes observed orbits that are missing from the catalogue. For instance, the order-5 family J shown above in the Interpreter's report for $k = 0.1$ is absent from Ueda's catalogue. Also, although Ueda describes the order-3 orbit in class D, he does not note the split into distinct families between $p = 6.7$ and $p = 8$ that is indicated in Fig. 3. Figures 5 and 6 illustrate these missing orbits.

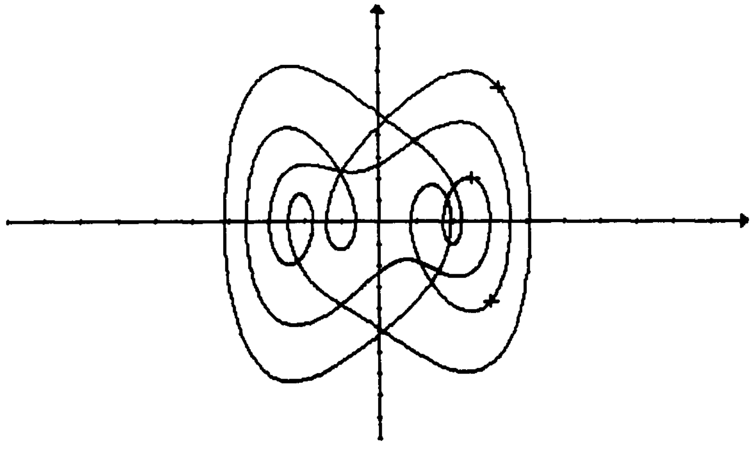

Fig. 6. This is the phase portrait for one of two distinct order-3 orbits in class D (Fig. 1) that coexist at the same parameter values ($k = 0.1\ p = 7.45$), a phenomenon reported by the Bifurcation Interpreter, but not noted in the catalogue published in Ref. [5]. The phase portrait of the second orbit is the reflection of this one under the symmetry $(x, \dot{x}) \mapsto (-x, -\dot{x})$. The crosses drawn on the orbit indicate the *Poincaré points*, at which t is a multiple of the period.

†The most significant difference between the two types of diagrams is that Fig. 4 indicates unstable periodic orbits (the dotted line beginning at the lower left of the figure) and shows regions of chaotic behavior (at the right-hand sides of the main figure and of in Ⓔ). The Bifurcation Interpreter does not currently record such information, but could be extended to do so.

3. WHAT THE INTERPRETER CAN RECOGNIZE

The Interpreter analyzes one-parameter families of dynamical systems. These can be discrete dynamical systems, formed by iterating diffeomorphisms f_p (p is the family parameter); or they can be systems of ordinary differential equations $\dot{x} = H_p(x, t)$, where each H_p is periodic in t with period t_H.† The periodic continuous case reduces to the discrete case by the standard trick of considering f_p to be the *period map*, which maps each state x to the point at $t = t_H$ on the trajectory of H_p that starts from x at $t = 0$. Since H_p is periodic, the successive iterates $f_p^{(k)}(x)$ are the points on the trajectory at times that are multiples of t_H. Thus, a trajectory that is periodic with period nt_H corresponds to the periodic point of order-n point of the period map, i.e. a point x such that $f_p^{(n)}(x) = x$.

Given a system to analyze, a parameter interval $p_l \leqslant p \leqslant p_h$ and a bounded domain in state space, the Interpreter attempts to determine, for each p in the parameter interval, the periodic points of f_p that lie within the specified domain. It does this by locating periodic points for a few values of p, and tracing the family of periodic points that evolves as p varies. A family may persist until the end of the parameter interval; or a family may vanish because the periodic point goes out of bounds (leaves the designated state-space domain); or the family may vanish when the periodic point undergoes a bifurcation.

3.1. Types of bifurcations

A bifurcation of a periodic point x of order n can be recognized computationally by the fact that, as p varies, an eigenvalue of $Df_p^{(n)}(x)$, the Jacobian derivative of f_p at x, crosses the unit circle in the complex plane.‡ When the Interpreter encounters a bifurcation, it attempts to classify the bifurcation, based upon the eigenvalues of $Df_p^{(n)}(x)$ and the pattern of stable and unstable periodic points meeting at x.

Here are the types of bifurcations that the Interpreter recognizes:§

- *Fold bifurcation*

  For values of p to one side of the bifurcation value a stable and an unstable periodic point (of the same order) coexist. These two periodic points meet as p approaches the bifurcation value; on the other side of the bifurcation, both periodic points have vanished.¶ An eigenvalue of $Df_p^{(n)}(x)$ crosses the unit circle at 1.
- *Flip bifurcations*

  —*Supercritical flip*. A stable periodic point of order n splits to form an unstable periodic point of order n and a stable periodic point of order $2n$.

  —*Subcritical flip*. This is equivalent to a supercritical flip of f^{-1}, the inverse of f: an unstable periodic point of order $2n$ merges with a stable periodic point of order n to form an unstable periodic point of order n.

  An eigenvalue of $Df_p^{(n)}(x)$ crosses the unit circle at -1.
- *Niemark bifurcations*

  —*Supercritical Niemark*. A stable periodic point vanishes, forming an unstable periodic point and a stable limit cycle.

  —*Subcritical Niemark*. A stable periodic point meets an unstable limit cycle, resulting in an unstable periodic point.

  An eigenvalue of $Df_p^{(n)}(x)$ crosses the unit circle at a value other than 1 or -1.
- *Transcritical bifurcation*

  A stable periodic point and an unstable periodic point meet and "exchange stabilities": on the other side of the bifurcation, the (extrapolated) stable point is now

---

†At present, the Interpreter has been fully implemented only for systems with a two-dimensional state space $x = (x_1, x_2)$, although most of the underlying algorithms will work in any number of dimensions. See the discussion in Section 5.

‡So long as no eigenvalue has modulus 1, the periodic point x is *hyperbolic* and thus will not change type with small changes in p. See Ref. [2] for details.

§In referring to the bifurcation types, we have adopted the terminology of Ref. [3]. Various authors use different, and even incompatible terminology. For example, the flip is also called a cusp or a pitchfork.

¶Depending on the direction from which the fold is encountered, it will be observed either as a stable periodic point and an unstable periodic point meeting and annihilating each other, or alternatively as the birth of a stable–unstable pair of periodic points. Since the Interpreter ordinarily encounters a bifurcation by tracking a stable point into it, we describe the bifurcations from that orientation.

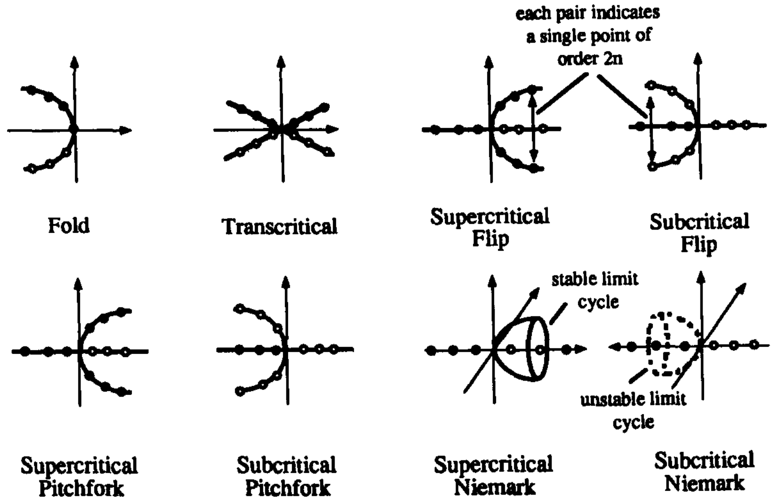

Fig. 7. This is the set of bifurcation types recognized by the Bifurcation Interpreter. The figure shows commonly used "bifurcation diagrams" that describe the pattern of stable and unstable periodic points near each bifurcation, in which the horizontal axis represents the parameter direction and the vertical axis (vertical plane in the case of the Niemark bifurcations) represents the state space. The curves indicate the loci of the periodic points as the parameter varies. Open circles indicate the loci of unstable points and solid circles indicate stable points.

unstable and the unstable point is stable. An eigenvalue of $Df_p^{(n)}(x)$ crosses the unit circle at 1.

- *Pitchfork bifurcations*
  - —*Supercritical pitchfork.* A stable periodic point splits to form two stable periodic points and an unstable periodic point, all of the same order.
  - —*Subcritical pitchfork.* A stable periodic point merges with two unstable periodic points, resulting in an unstable periodic point.

  An eigenvalue of $Df_p^{(n)}(x)$ crosses the unit circle at 1.

These bifurcations are summarized in Fig. 7 by means of bifurcation diagrams, of the kind commonly employed in the dynamics literature. These diagrams show, near each bifurcation, the paths of the periodic points as p varies, projected onto a one-dimensional subspace of phase space (two-dimensional subspace for the Niemark bifurcations). The parabolic paths drawn for most of the bifurcations are more than simply schematic. It can be shown [3] that, for values of p very near the bifurcation value, the periodic points approach the bifurcations along such parabolic paths. The methods by which the Interpreter recognizes bifurcations make use of this fact (Section 4.4).

The bifurcations listed here are the typical ones encountered in one-parameter families of maps, in the following sense. It can be shown that the only bifurcations that appear in generic one-parameter families are the fold, the flip, and the Niemark bifurcations. In addition, the transcritical bifurcation is generic for one-parameter families of maps that are constrained to have a common fixed point for all f_p; and the pitchfork bifurcations are generic in families of maps that are invariant under a symmetry. See Refs [2, 3] for more details.

In addition to recognizing individual bifurcations, the Interpreter also notices typical patterns of bifurcations. One such pattern is the *period-doubling cascade* [6], in which an infinite sequence of supercritical-flip bifurcations occurring at a converging sequence of values of p produces periodic points of orders n, $2n$, $4n, \ldots$, followed by chaotic orbits at values of p beyond the accumulation point. A second pattern is the *symmetric pair*, produced when an orbit that is invariant under a symmetry of the system loses its symmetry and splits, at a supercritical-pitchfork bifurcation, into two orbits, each of which is transformed into the other by the symmetry.† Continued evolution of these orbits produces symmetric pairs of families and bifurcations.

†For example, the Duffing system described in Section 2.1 has the form $\dot{x} = H(x, t)$, where $H(-x, t + \pi) = -H(x, t)$. One can show that this entails a symmetry: for every closed orbit S, the set $-S = \{-x | x \in S\}$ is also a closed orbit. Thus, either S itself is invariant under the symmetry, or is one of a symmetric pair of closed orbits, as in Fig. 6.

3.2. The family report

Although the text and diagrams illustrated in Section 2.1 are the most conspicuous part of the Bifurcation Interpreter's output, the bulk of the program is concerned with conducting numerical experiments and distilling their results into a data structure called a *family report*. The family report is the basis for generating the text and diagrams. It can also be examined by other programs to answer questions about dynamical systems, e.g. to indicate the ranges over which multiple families exist or to list the types of bifurcations encountered.

The family report lists the classes of families in order of appearance with increasing p. For instance, the analysis of the Duffing system shown in Fig. 1 has 10 classes. The first of these, class A, contains three families, two of which form a symmetric pair:

```
(class:A
 (members
  ( #[family:A.0] (symmetric-pair  #[family:A.1.0] #[family:A.1.1]))))
```

Each family is represented by a data structure that contains all the information collected about the family, such as the name and order of the family, the bifurcations at which the family appears and vanishes and the instances of the periodic point that was tracked to form the family. Here is the entry for $A_{1,0}$, the second family in class A. This entry includes the information that $A_{1,0}$ is part of a symmetric pair, paired with family $A_{1,1}$:

```
(family:A.1.0
 (order 1)
 (appears #[bifurcation:bif.1])
 (vanishes #[bifurcation:bif.5])
 (paired-with (#[family:A.1.1]))
 (instances 59 #[list of instances ...]))
```

The list of instances describes the individual periodic points. Here is the first point in the list for $A_{1,0}$:

```
(instance
 (parameter-value 2.2867)
 (type periodic)
 (orbit #(1.8851 .39982))
 (jacobian #(#(-.29979 -9.5576e-2) #(11.327 1.8317)))
 (eigenvalues .99665 .53528)
 (sensitivity  #(-7.9580 111.496)))
```

This is a periodic point at $p = 2.2867$ and $x = (1.8851, 0.3998)$. Also recorded are the Jacobian and the eigenvalues of the period map at this point, and the sensitivity derivative (i.e. the derivative with respect to p of the (vector-valued) function $p \mapsto f(p, x)$), which is used in tracking periodic points as explained in Section 4.3.

In addition to the list of classes, the Interpreter generates a list of the bifurcations encountered. Here is the bifurcation at which $A_{1,0}$ appears:

```
(bifurcation:bif.1
 (type supercritical-pitchfork)
 (parameter-range (2.2864 2.2867))
 (families-vanishing (#[family:A.0]))
 (families-appearing  (#[family:A.1.0] #[family:A.1.1]))
 (symmetric-pair (split to form symmetric pair))
 (direction parameter-increasing)
 (comments ((inferred from merging families))))
```

This is a supercritical-pitchfork bifurcation located between $p = 2.2864$ and $p = 2.2867$, where A_0 vanishes and $A_{1,0}$ and $A_{1,1}$ appear. The vanishing family splits to form a symmetric pair of families with a common head (instance at lowest value of p). The bifurcation is oriented in the

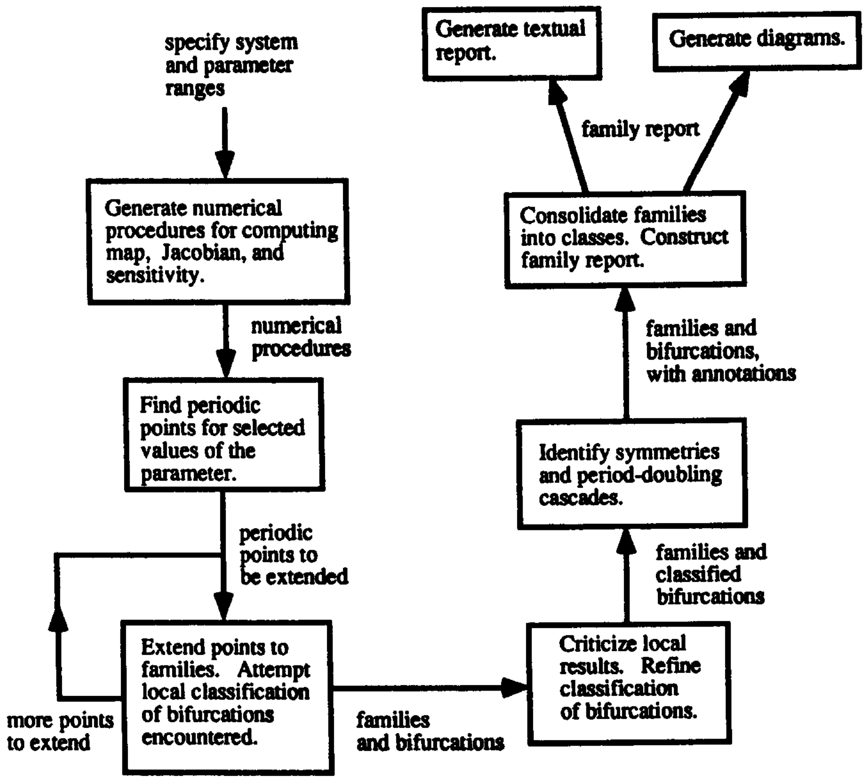

Fig. 8. The block diagram shows the major components of the Bifurcation Interpreter, beginning with a system specified by algebraic equations and ending with a family report, from which text and diagrams are generated. The arrows between blocks indicate the principal output of each module, which is the input to the next module.

"parameter increasing" direction (i.e. the split happens as p increases). Finally, a comment records that, in this case, the Interpreter discovered the bifurcation by noticing that two families merged [see Fig. 10(b)].

Finally, the Interpreter maintains a list of the parameter values at which it found orbits that it was not able to classify. Depending on the nearby bifurcations, these might be interpreted as chaotic orbits. Given the information in the family report, it should be clear how both the text and the diagrams in Section 2.1 can be generated.

4. HOW THE INTERPRETER WORKS

Figure 8 shows the major modules in the Bifurcation Interpreter and the information that flows from one module to the next. Beginning with equations that describe a dynamical system, and a designated parameter range to explore, the Interpreter generates numerical procedures that compute the transformation from state to state, the Jacobian of the transformation and the sensitivity (the derivative of the transformation with respect to the varying parameter). With the aid of these procedures, the Interpreter locates stable periodic points at various parameter values. Each periodic point is tracked as the parameter varies, producing a family that generally ends in a bifurcation. The Interpreter attempts to classify the bifurcation based upon information obtained in the tracking process. Typically, classifying the bifurcation will produce new stable periodic points that must themselves be tracked.

When all known periodic points have been extended to complete families, the Interpreter reexamines the bifurcations in the light of more global information, and corrects its previous classifications if necessary. Next, bifurcations and families are examined for symmetric pairs and period-doubling cascades. The Interpreter consolidates families into classes, assigns names to the families and bifurcations and organizes the data into a family report. The family report is the basis for generating text and diagrams as exhibited in Section 2.1.

We now consider each of these steps in detail.

4.1. Compiling procedures for Newton's method

Most of the Interpreter's algorithms rely upon the ability to locate fixed points. This is accomplished with the aid of Newton's method. Applying Newton's method to find fixed points of a map f requires evaluating both f and the Jacobian derivative Df at various points x in the state space. For a discrete dynamical system, where f is specified explicitly by algebraic equations, the Interpreter differentiates the equations symbolically to produce expressions for the components of the Jacobian matrix $[Df(x)]_{ij} = \partial f^i/\partial x_j$. These expressions are compiled into a procedure that evaluates the Jacobian, given a numerical value for x.

For a system of ordinary differential equations $\dot{x} = H(x, t)$ with $H(x, t)$ periodic in t, f is not given by an explicit algebraic equation. Rather, f is the period map, evaluated by integrating H along the trajectory starting from x. The components of the (transpose of the) Jacobian matrix, $\delta_{ij} = [Df(x)]^T_{ij} = [Df(x)]_{ji}$ are computed by integrating, along the same trajectory, the associated *variational system*

$$\frac{d\delta_{ij}}{dt} = \sum_k \frac{\partial H^j}{\partial x_k} \delta_{ik},$$

with initial conditions $\delta_{ij}(0) = 0$ for $i \neq j$ and $\delta_{ij}(0) = 1$ for $i = j$. The Interpreter symbolically differentiates the expressions for $H(x, t)$ to obtain expressions for the $\partial H^i/\partial x_j$ and compiles these into a procedure for computing the Jacobian by numerical integration.

For example, Fig. 9 shows the Duffing system of equation (1) as it is actually input to the Interpreter, with a specification of the state variables, the parameter and equations for the derivatives of the state variables.† The Interpreter produces from this a *system-derivative generator.* This is a procedure which, given values for the parameters k and p, returns a *system derivative*: a procedure which, given a state vector (t, x_1, x_2), returns a vector whose components are the derivatives with respect to t of the state-vector components. Figure 9 also shows the procedure generated for evolving the associated variational system. Given values for k and p, this produces a variational-system derivative, which operates on a variational state vector with components t, x_1, x_2 and the δ_{ij}, returning the vector whose components are the time derivatives of t, x_1, x_2, together with the components of the Jacobian, computed as

$$\frac{d\delta_{11}}{dt} = \delta_{12},$$

$$\frac{d\delta_{12}}{dt} = -3x_1^2\delta_{11} - k\delta_{12},$$

$$\frac{d\delta_{21}}{dt} = \delta_{22}$$

and

$$\frac{d\delta_{22}}{dt} = -3x_1^2\delta_{21} - k\delta_{22}.$$

The Interpreter generated the expressions for these derivatives by symbolic differentiation.

The system-derivative procedures generated by the Interpreter can be combined with any numerical integrator to evaluate the period map and the Jacobian. Computations for the Duffing example described in this paper were carried out using a Bulirsch–Stoer integrator adapted from Ref. [8].

Having developed procedures for computing maps and Jacobians, the Interpreter can use Newton–Raphson iteration to search for periodic points of a map f. Namely, given an approximation x to a periodic point of order n for f, attempt to correct x by subtracting from it a small

†The input language used here, phrased in terms of constraints, is more general than is required for the purposes of the Interpreter. The same language, and the techniques for generating system derivatives and variational equations, were used by Abelson and Sussman to support an electrical-circuit analysis program, as described in Ref. [7].

```
(define-network duffing
  ((x2 state-variable)
   (x1 state-variable)
   (p parameter)
   (k parameter))
  (constraint '(= ,(rate x1) ,x2))
  (constraint '(= ,(rate x2)
                  (- (* ,p (cos t)) (+ (* ,k ,x2) (* ,x1 ,x1 ,x1))))))

(lambda (k p)
  (lambda (*state*)
    (let ((t (vector-ref *state* 0))
          (x1 (vector-ref *state* 1))
          (x2 (vector-ref *state* 2)))
      (vector 1
              x2
              (+ (* -1 k x2) (* -1 x1 x1 x1) (* p (cos t)))))))

(lambda (k p)
  (let ((g1 (* -1 k)))
    (lambda (*varstate*)
      (let ((t (vector-ref *varstate* 0))
            (x1 (vector-ref *varstate* 1))
            (x2 (vector-ref *varstate* 2))
            (del.x1.x1 (vector-ref *varstate* 3))
            (del.x1.x2 (vector-ref *varstate* 4))
            (del.x2.x1 (vector-ref *varstate* 5))
            (del.x2.x2 (vector-ref *varstate* 6)))
        (let ((g2 (* x1 x1)))
          (let ((g3 (* -3 g2)))
            (vector 1
                    x2
                    (+ (* -1 g2 x1) (* g1 x2) (* p (cos t)))
                    del.x1.x2
                    (+ (* g1 del.x1.x2) (* g3 del.x1.x1))
                    del.x2.x2
                    (+ (* g1 del.x2.x2) (* g3 del.x2.x1))))))))))
```

Fig. 9. The Interpreter manipulates the symbolic expressions that describe the system, in order to generate numerical procedures for locating periodic points. Shown here is the actual input that describes the Duffing system of equation (1). Below this is a Lisp procedure that evolves the system state, which the Interpreter has generated and compiled; and below this is the procedure compiled to evolve the associated variational system. Notice that the compiler has performed some common-subexpression removal here ($g1$, $g2$ and $g3$) to make the computation more efficient.

vector δx such that $x - \delta x$ will be equal to $f^{(n)}(x - \delta x)$. Linearizing this equation near x gives an approximation for δx in terms of the Jacobian $Df^{(n)}(x)$:

$$f^{(n)}(x - \delta(x)) = x - \delta x,$$
$$f^{(n)}(x) - Df^{(n)}(x)\delta x \approx x - \delta x$$

and so

$$\delta x \approx (I - Df^{(n)}(x))^{-1}(x - f^{(n)}(x)).$$

In terms of procedures that compute f and Df, this computation can be accomplished by computing the orbit of x:

$$x_0 = x, \quad x_i = f(x_{i-1}) \qquad i = 1 \text{ to } n; \tag{2}$$

and using the recurrence

$$Df^{(i)}(x) = Df(x_{i-1})Df^{(i-1)}(x) \tag{3}$$

to compute $Df^{(n)}(x)$.

Given an initial approximation x to a periodic point† one repeatedly updates x to $x - \delta x$, looking for convergence. Since Newton's method normally converges rapidly if it converges at all, and the update procedure is expensive (computing Jacobians can require integration), the Interpreter is quick to abandon an attempt to find a periodic point if the sequence does not appear to be converging. In particular, if the second correction $\delta x^{(2)}$ is larger than the first correction δx, the Interpreter will conclude that the initial guess for the periodic point was bad, and stop the search.‡ Newton's method will also fail if $I - Df^{(n)}(x)$ becomes singular. In addition, the Interpreter's Newton's-method procedure can be called with an arbitrary predicate, which successive guesses must satisfy in order for the search to continue. For example, guesses must normally remain inside the bounded region in state space that the Interpreter is exploring. Also, some of the Interpreter's bifurcation identification procedures require finding periodic points that lie within certain specified neighborhoods; these procedures call Newton's method with appropriate predicates to enforce these constraints.

Finally, if the sequence of approximations converges to a periodic point x of the desired order n, the Interpreter checks the orbit of x to see whether x in fact has order smaller than n.

4.1.1. Sensitivity derivatives. In addition to approximating periodic points in a parametrized family of maps, the Interpreter also tracks the loci of the periodic points as the parameter varies. This requires computing the infinitesimal change in position of a periodic point x of f with respect to p. To see how to compute this, consider first the case where x is a fixed point of the map $x \mapsto f(p, x)$. For an incremental change δp, compute the corresponding incremental change δx such that $x + \delta x$ approximates $f(p + \delta p, x + \delta x)$ to first order:

$$\begin{aligned} x + \delta x &= f(p + \delta p, x + \delta x) \\ &\cong f(p, x) + D_x f(p, x) + D_p f(p, x)\delta p \\ &= x + Df_p(x)\delta x + D_p f_p(x)\delta p, \end{aligned}$$

where $Df_p(x)$ is Jacobian at x of the map $f_p\colon x \mapsto f(p, x)$ and $D_p f_p(x)$ is the derivative with respect to p of the (vector-valued) function $p \mapsto f(p, x)$. Subtracting x from both sides of this equation and solving for δx gives

$$\delta x \approx (I - Df_p(x))^{-1} D_p f_p(x)\delta p. \tag{4}$$

If f_p is given by explicit equations (the case of a discrete dynamical system), the Bifurcation Interpreter can symbolically differentiate the expressions for f_p with respect to p and construct a numerical procedure that computes $D_p f_p(x)$. For a system of ordinary differential equations $\dot{x} = H_p(x, t)$, where f_p is the period map, $D_p f_p(x)$ can be computed by integrating the components of $\mathrm{d}H/\mathrm{d}p$ over the trajectory starting from x. In the case of the Duffing system (1) for example, the components of $\mathrm{d}H/\mathrm{d}p$ to be integrated are obtained by differentiating system (1) with respect to p to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}x_1}{\mathrm{d}p}\right) = \frac{\mathrm{d}x_2}{\mathrm{d}p}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}x_2}{\mathrm{d}p}\right) = \cos t - k\frac{\mathrm{d}x_2}{\mathrm{d}p} - 3x_1^2\frac{\mathrm{d}x_1}{\mathrm{d}p}.$$

Just as in computing Jacobians, the Interpreter uses symbolic algebra to perform this differentiation. It automatically compiles a *sensitivity derivative* which can be integrated to compute $D_p f_p(x)$.§

†Section 4.2 explains how the Interpreter selects initial approximations.
‡This technique was suggested to me by Mathew Halfant.
§This technique for computing sensitivities of fixed points was also used in Ref. [7] for analyzing electrical networks.

Tracking an order-n periodic point of f can be treated as tracking a fixed point of $f^{(n)}$. However, as with locating fixed points, it is convenient to use the recurrence scheme (2, 3) and to compute $D_p f_p^{(n)}(x)$ recursively in terms of $D_p f_p(x)$ by means of the relation

$$D_p f_p^{(i)}(x) = D_p f_p(x_{i-1}) + Df(x_{i-1}) D_p f_p^{(i-1)}(x).$$

This equation follows from the chain rule for partial derivatives;

$$D_p(h \circ g)(x) = D_p h(g(x)) + Dh(g(x)) D_p g(x),$$

taking h as f_p and g as $f_p^{(i-1)}$.

In summary, whenever the Interpreter successfully applies Newton's method to produce a periodic point x of order n, it saves x, together with its orbit, the value of p, the Jacobian $Df_p^{(n)}(x)$ and the sensitivity derivative $D_p f_p^{(n)}(x)$.

4.2. Choosing an initial set of periodic points

After compiling the numerical procedures as described above, the Interpreter computes an initial set of periodic points that it will later extend to families. The parameter interval to be explored is divided into equal-length subintervals at points p_i. For each p_i, the Interpreter attempts to find the stable periodic points of f_{p_i}. The algorithm for finding these periodic points produces, at each p_i, a collection of periodic points and possibly some "unknown orbits", which arise from trajectories that do not lead to periodic points.† The Interpreter next compares the collection of points discovered at adjacent values p_i, p_{i+1}. If these are not equivalent— i.e. if the two collections do not contain the same number of periodic points with corresponding orders, and the same number of unknown orbits—then the interval from p_i to p_{i+1} is bisected, and periodic points are computed at the midpoint. Bisection continues until adjacent values of p either have equivalent periodic-point sets, or else are closer than some prespecified minimum distance.

To compute periodic points at a parameter value p, the Interpreter first guesses approximations for the periodic points by means of the "unravelling algorithm" of Hsu and Guttalu [9, 10]. This begins by imposing a grid on the region in state space to be examined. Define a cell-to-cell transformation by mapping each grid cell C to the cell containing the point obtained by applying f_p to the midpoint of C, mapping C to a distinguished value "infinity" if the image of the midpoint under f_p lies outside the grid. Regarding the cells as the nodes of a graph, where each cell is linked to its image under the cellular map, produces a directed graph in which each node has at most one successor. Use a marking algorithm to traverse the graph and partition the cells into connected components of the graph. During the marking, whenever you encounter a cycle of order n starting at a cell C, choose the midpoint of C as a guess for an order-n periodic point of f_p.‡

The Interpreter now tries each guess as the starting point for a Newton–Raphson iteration. If the iteration succeeds, the periodic point and its orbit are recorded. If Newton's method fails (e.g. the successive iterates may become unbounded, or the resulting limit point may be unstable, or the Jacobian may become singular) the Interpreter uses the guess as a starting point for a "trajectory search", which proceeds as follows:

Given an initial point $x = x_0$, generate the trajectory $x_1 = f_p(x_0)$, $x_2 = f_p(x_1), \ldots$. As each new point x_a is generated, check to see if it is close to a point in a previously-generated trajectory, in which case abandon the search starting at x. Alternatively, if x_a is close to a point in the current trajectory, scan backward along the trajectory to find the smallest n such that x_{a-n} is close to x_a.§ Now generate a few more points starting from x_a, and check whether the points $x_{a+1}, x_{a+2}, \ldots,$ remain close, respectively, to $x_{a+1-n}, x_{a+2-n}, \ldots$ If so, use x_a as the starting point for a

†These unknown orbits may corespond, in actuality, to orbits that are indeed nonperiodic, or to orbits that have extremely large period, or possibly to bad numerical properties of f_p. For example, iterating f_p for p extremely close to a bifurcation value may lead to unknown orbits.

‡Note that any path through the graph will end in a cycle, a fixed point (cycle of order 1) or else infinity. The connected components of the graph correspond to the "basins of attraction" of these cycles. Regarding the corresponding sets of cells as approximations to the actual basins of attraction of f leads to interesting computational methods for exploring dynamical systems. These are described in Ref. [9], which also presents more sophisticated variations on the unravelling algoritm described above.

§Trajectory points are stored in a quad tree [11] and also linked in a doubly-linked list to facilitate finding close pairs of points and scanning forwards and backwards through the trajectory.

Newton–Raphson search for a stable periodic point of order n. If the iteration does not succeed, or if x_a is not close to any previous trajectory point, then continue evolving the trajectory. If the number of points in a trajectory grows to exceed some maximum, declare the trajectory to be of type "unknown" and abandon it.

As an example, in producing the report on the Duffing system in Section 2.1, the Interpreter's parameters were set so that it initially divided the parameter interval from 1 to 25 into subintervals of size 0.5. The minimum size for an interval (at which bisection ceased) was 0.1. The state-space region $-5 < x < 5$, $-10 < y < 10$ was partitioned by a grid of size 0.5 on a side. With these settings, the Interpreter ended up scanning for periodic points at 93 values of p. In all, the Hsu–Guttalu algorithm produced 307 guesses for periodic points. Of these, 114 were successfully refined to periodic points by Newton–Raphson; the other 193 became starting points for trajectory searches. At many values of p several initial guesses were refined to the same periodic point, and after removing duplicates, the Interpreter had in all identified 117 distinct periodic points and 15 unknown orbits.†

4.3. Extending periodic points to families

Each stable periodic point now becomes the "seed point" for a family of periodic points whose position varies with p. The Interpreter tracks the point, extending each family as far as possible in the directions of increasing and decreasing p. During the tracking process, families may merge, or they may terminate in bifurcations, which the Interpreter attempts to classify.

To track a stable periodic point x from p_s to p_t, the Interpreter chooses a small increment δp (respectively, a small decrement δp if $p_s > p_t$) and computes the corresponding first-order change δx using equation (4), together with the values of the Jacobian and the sensitivity derivative that were stored along with x, as explained in Section 4.1. The Interpreter then uses Newton–Raphson iteration to attempt to correct $x + \delta x$ to a stable periodic point of f at $p_s + \delta_p$.

In tracking a periodic point at discrete steps like this, one must be careful not to mistakenly "jump" to the path of a nearby periodic point. The Interpreter attempts to guard against this by adaptively adjusting the stepsize δp, always choosing it small enough so that: (1) the magnitude of the computed δx is small; (2) the actual periodic point discovered by Newton's method is close to the first-order prediction $x + \delta x$; and (3) the change in magnitude of the eigenvalues of the Jacobian from x to the new periodic point is small.

At each step, Newton's method may fail to produce a stable periodic point. The iteration may not converge, or the matrix $I - Dpf^{(n)}(x)$ may become singular, so that no periodic point will be produced. Alternatively, the periodic point found may be unstable (which can be determined by examining the eigenvalues of the Jacobian). Or, the point may be stable, but with an order less than the order of x. Upon such failures, the Interpreter decreases the size of δp and tries again. When δp becomes less than some prespecified lower bound, and Newton's method still fails, the Interpreter regards the failure as evidence of a bifurcation at a parameter value betwen p_s and $p_s + \delta p$. It records the nature of the failure and uses this information in an attempt to classify the bifurcation, as described below in Section 4.4. If Newton–Raphson succeeds, the Interpreter increases the step size δp and continues tracking toward p_t.

The Interpreter employs this tracking method to extend each family as far as possible—until the family terminates in a bifurcation, or the periodic point leaves the bounded region in state space that the Interpreter was directed to explore, or the tracker reaches an endpoint of the parameter interval. As each family is generated, it is represented as a list of periodic points in order of increasing p. Let the *current starting value* of a family be the smallest value of p to which the family has so far been extended, and the *first known instance* is the periodic point at that value of p. Similarly, the *current ending value* and the *last known instance* give the parameter and the periodic point at the largest value of p to which the family has so far been extended. A family is said to

†Note that the grid size used here was rather large, so one should expect the Hsu–Guttalu algorithm to produce many bad guesses for periodic points. Choosing a finer grid, however, would be computationally more expensive. More work needs to be done to investigate the tradeoffs here, and to find good ways to pick the initial grid size. Similarly, it is not obvious how to choose good sizes for the parameter intervals. Choosing intervals that are large risks overlooking significant families of periodic points and bifurcations.

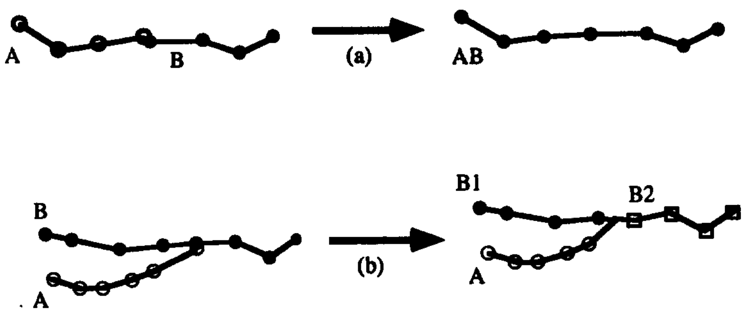

Fig. 10. Merging families. (a) If family A extends to meet the first known instance of B, the two families are merged to form a single family AB. (b) If A coincides with a point in the middle of B (at a synchronization value of the parameter), B is split into two families B_1 and B_2. A and B_1 vanish together at a bifurcation somewhere before the synchronization point, and B_2 appears there. Note that, in this figure, the horizontal direction represents the p-axis, while the vertical direction represents the location of the points in state space (projected onto a one-dimensional subset).

have its *vanishing known* (respectively, its *appearance known*) if it has been extended as far as possible in the direction of increasing p (respectively, decreasing p).

While tracking, the Interpreter must also check for merges among families. This requires some care, because individual families are tracked separately, and the points of different families will in general be computed at different parameter values. Specifically, the Interpreter proceeds as follows.†

The parameter interval is partitioned at a number of equal-spaced *synchronization values*. The Interpreter chooses a family, say family A, whose vanishing is not yet known. Let x be the first known instance of A, at parameter value p_s. Choose p_t to be the minimum of:

(i) the right-most endpoint of the parameter interval;
(ii) the smallest value after p_s that is the current starting value of a family whose appearance is not yet known;
(iii) the smallest synchronization value larger than p_s;
(iv) the smallest value larger than p_s at which there is a bifurcation.

If the tracker fails to extend A as far as p_t, then the type of failure (point going out of bounds or reaching a bifurcation) shows how A vanishes, and the Interpreter accordingly marks the As vanishing as known. If the extension to p_t is successful, then the Interpreter checks for merges of A with other families. There are several cases to consider, depending on how p_t was chosen:

(1) If p_t is the endpoint of the parameter interval, then the Interpreter marks A as persisting to the end of the interval, and the extension of A is complete.
(2) If p_t is the current starting value of a family B, the Interpreter compares the currently tracked point with the first known instance of B. If the points are the same, then A and B are merged as shown in Fig. 10(a). If not, tracking of A continues from p_t.
(3) If p_t is a synchronization value, the currently tracked point is compared with the points of other families p_t. If it matches a point of some other family B, then A and B must merge at some value between p_s and p_t. This merge corresponds to a bifurcation; and B must actually be two families: B_1 (the part of B before the bifurcation); and B_2 (the part of B after the bifurcation). At the bifurcation, A and B_1 vanish and B_2 appears. The Interpreter searches the interval between p_s and p_t to locate the merge point and replaces B by B_1 and B_2 [see Fig. 10(b)].

Locating the precise merge point here is slightly tricky. Families A and B coincide at p_t and are distinct at p_s, so the merge point can be found by binary search. However, the periodic points on A and B have not been computed at the same parameter values between p_s and p_t and so cannot be compared directly.

†For simplicity, we describe the tracking process in the direction of increasing p. Tracking in the direction of decreasing p proceeds analogously.

Comparing the families at intermediate parameter values thus requires computing additional points of A or B (or both). This is accomplished by tracking from existing points at nearby parameter values.†

(4) Finally, if p_t is a bifurcation value, the Interpreter checks to see if A vanishes at the bifurcation. If not, tracking of A continues at p_t. Otherwise, A is compared with the other families currently known to vanish at the bifurcation. (This comparison may require generating additional periodic points, as in the previous paragraph.) If A does coincide with some family B that has already been tracked to the bifurcation at p_t, then A must in fact merge with B at another bifurcation before p_t, as in case (3). Alternatively, if A is a new family vanishing at the bifurcation, then the Interpreter marks it as such.

In this way, tracking proceeds until all families have been extended so that their appearance and vanishing is known.

4.4. *Identifying bifurcations*

When the periodic-point tracking algorithm fails, the Interpreter assumes that it has encountered a bifurcation, and it attempts to classify the bifurcation. There are three ways in which tracking a stable periodic point might fail: (1) Newton–Raphson iteration fails to converge, so that no new periodic point is found at the next parameter value; (2) the iteration converges, but to an unstable periodic point; and (3) the iteration converges to a stable periodic point, but of lower order than the point being tracked.

Comparing these failure types against the descriptions of the generic bifurcations given in Section 3.1 and Fig. 7 indicates how to begin matching tracking failures against bifurcation types. In case (3), for example, the only bifurcation in which a stable periodic point transforms into a stable periodic point of a lower order is the supercritical flip, where a point of order $2n$ changes to a point of order n.‡ Accordingly, if the Interpreter encounters a failure of type (3), it expects that the order of the new point will be half the order of the old point, and that an eigenvalue of map is crossing the unit circle at -1. If both these conditions hold, the Interpreter will attempt to identify the transition as a supercritical flip. Otherwise, it will report a bifurcation of unknown type.

For the Interpreter, "identifying a bifurcation" means finding the complete complement of stable and unstable periodic points that should coalesce at the bifurcation. Thus, at a supercritical flip, there is a stable point of order n to one side of the bifurcation, and, to the other side, a stable point of order $2n$ and an unstable point of order n. If the Interpreter has found the two stable points, it attempts to locate the expected unstable point, using methods that are described below. If it succeeds, it reports the bifurcation as a supercritical flip. Otherwise it reports the bifurcation as unknown.

In general, the Interpreter begins its attempt to classify a bifurcation by examining the eigenvalues of the point near the tracking failure. If an eigenvalue crosses the unit circle near -1, the bifurcation (if it is one of the recognized types) must be a supercritical or a subcritical flip; crossing the unit circle near 1 leads to a fold, a transcritical bifurcation, a supercritical pitchfork or a subcritical pitchfork; crossing the unit circle off the real axis leads to a supercritical or subcritical-Niemark bifurcation. In each case, the Interpreter searches near the bifurcation for additional stable and unstable periodic points, until it finds enough points to match the bifurcation against one of the known types.§

If a recognized local portrait is found—i.e. if the number of stable and unstable periodic points at the bifurcation matches the pattern for one of the known bifurcation types—the Interpreter

---

†There is also the possibility that this secondary tracking may fail, since, after all, one is close to a bifurcation. This case the Interpreter assumes that a tracking failure here is due to the bifurcation it is searching for.

‡Notice, with reference to Fig. 7, that this transition corresponds to crossing through the supercritical-flip bifurcation from right to left. The Interpreter must recognize bifurcations encountered from either direction.

§In the case of a Niemark bifurcation, there is a stable fixed point to one side of the bifurcation, and, to the other side, an unstable fixed point and a limit cycle—a stable limit cycle for a supercritical Niemark and an unstable limit cycle for a subcritical Niemark. Currently, the Interpreter looks only for fixed points, not limit cycle. Thus, it cannot distinguish a supercritical Niemark from a subcritical Niewmark, and reports any such bifurcation simply as a "Niemark bifurcation".

records a successful local identification. Otherwise, the Interpreter records the bifurcation type as unknown. In either case, the result of the classification, together with the set of periodic points found by the search is passed to the next phase of the program, which criticizes these local identifications in the light of more global information, as discussed in Section 4.5.

Finally, any new stable periodic points located during this process become seed points for new families, which must be tracked in the direction away from the bifurcation. These are added to the collection of families to be extended using the method of Section 4.3, and the process continues.

4.4.1. Searching for periodic points near bifurcations. When the Interpreter searches for a new periodic point near a bifurcation, it guesses a location for the point, and uses this guess as the starting point for a Newton–Raphson iteration, attempting to find the actual new point.

The guess for the new periodic point depends upon the configuration of periodic points already observed near the bifurcation. As Fig. 7 indicates, when one is very close to the bifurcation, the periodic points tend to approach it along either parabolic or straight-line paths.† The Interpreter uses this local geometry to extrapolate locations for the unknown periodic points.

For example, suppose that the Interpreter is investigating a bifurcation at which an eigenvalue is approaching 1, and that there are two stable periodic points (of the same order) to one side of the bifurcation and one stable periodic point to the other side. Given the possibilities in Fig. 7, this ought to be a supercritical-pitchfork bifurcation, and there ought to be another unstable periodic point on the same side as the two stable points. Moreover, the local geometry near a supercritical-pitchfork indicates that the two stable points are located approximately symmetrically with respect to the unstable point. The Interpreter therefore searches for an unstable periodic point at the same parameter value as the two stable points, and at their average position in state space. This particular search method is called *averaging*. The same method is used when there are two unstable points to one side of the bifurcation and an stable point to the other side to search for a new stable point at the average position of the unstable points.

In general, each of the Interpreter's search methods is triggered by a particular pattern of stable and unstable periodic points. To investigate a bifurcation, the Interpreter tries each applicable search method, looking for new periodic points. As new periodic points are discovered, different search methods become applicable. If all applicable search methods have been tried, and the pattern of fixed points is not recognized, the Interpreter reports the bifurcation as unknown.

Here is the rest of the search methods used at bifurcations where an eigenvalue approaches 1. For these bifurcations the periodic points all have the same order. These methods are also illustrated schematically in Fig. 11.

- *Reflect*
  Pattern—a stable and an unstable periodic point to one side of the bifurcation. Search for a second stable point at the same parameter value by reflecting the stable point in the unstable point. (Analogously, search for a second unstable point by reflecting the unstable point in the stable point.)
- *Extrapolate across*
  Pattern—a stable point and an unstable point to one side of the bifurcation, say, at parameter value p_a, and an unstable point to the other side of p_b. Search for a stable point at p_b as follows. At p_a find the vector from the unstable point to the stable point. At p_b subtract this vector from the unstable point, and search for a stable point here. Analogously, given instead a stable point at p_b, search for an unstable point by extrapolating from the stable and unstable points at p_a.
- *Extrapolate through*
  Pattern—two stable points and one unstable point to one side of the bifurcation. Search for a stable point on the other side, at the same position in state space as the unstable point.

†These local geometries can be verified by truncating the power-series expansion for $f(p, x)$ near the bifurcation and solving for the paths of the periodic points. This is in fact the basic technique for cataloguing the types of generic bifurcations. See Refs [2, 3] for details.

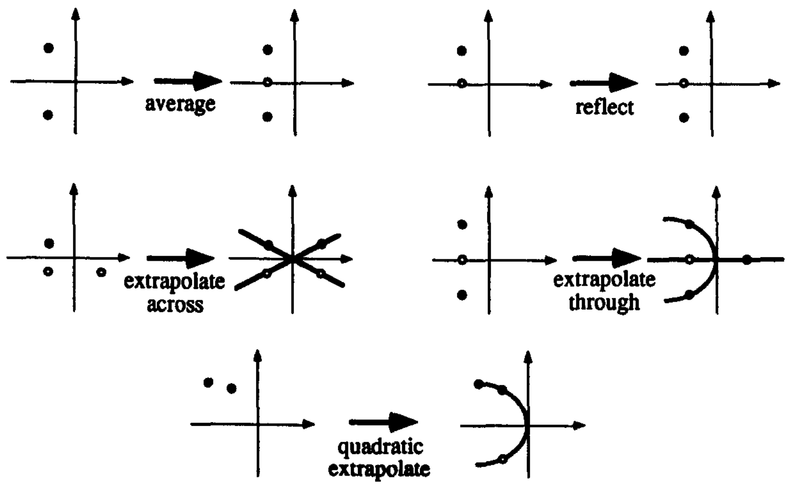

Fig. 11. These figures indicate the methods used by the Interpreter to search for new stable and unstable periodic points near a bifurcation where an eigenvalue approaches 1. Solid circles indicate stable points and open circles indicate unstable points. The coordinate system drawn here has the bifurcation at the origin. The horizontal axis gives the parameter direction p. The vertical axis represents the two-dimensional state space. The curves shown for the extrapolation methods indicate the assumed linear or parabolic paths of the periodic points near the bifurcation, which form the basis for the extrapolation. Compare this figure with the bifurcation diagrams in Fig. 7 to see the full local geometries that these search methods are attempting to complete.

- *Quadratic extrapolate*

  Pattern—a single stable point to one side of the bifurcation at p_b. Search for an unstable point also at p_b, using the fact that, for folds or pitchforks, points very close to the bifurcation approach it along parabolic paths. The method requires knowing not only the location of the point at p_b, but also the periodic point found by the tracker at the previous step to p_b, say at parameter value p_c. Let p_a be the parameter value where the tracking failed. Fit a branch of a parabola through the points (x_{p_c}, p_c), (x_{pb}, p_b) and $(0, p_a)$. Take, as a guess for the unstable point, the point on the opposite branch of the parabola at p_b.

Here is an example, taken from a test of the Interpreter, which illustrates these search methods in action. The test map here is

$$f_p\colon (x_1, x_2) \mapsto \left(\frac{x_2}{2}, x_1 - \frac{x_2^2}{4} + \frac{px_2}{2}\right).$$

This map has a transcritical bifurcation at $p = 1$, $(x_1, x_2) = (0, 0)$, which the Interpreter successfully identified. In the test, the Interpreter first encountered the bifurcation while tracking a fixed point for decreasing p. The tracker found a stable fixed point before the bifurcation at $p = 1.0004$ and an unstable fixed point after the bifurcation at $p = 0.999$. Over the transition, an eigenvalue crossed the unit circle at 1. The Interpreter first used quadratic extrapolation of the stable point. This successfully located an unstable point before the bifurcation. The Interpreter next attempted to find a second stable point before the bifurcation by reflecting the stable point in the unstable point. This attempt failed. The Interpreter then extrapolated the unstable point across the bifurcation to produce a stable point after the bifurcation. This resulted in stable–unstable pairs both before and after the bifurcation, and the Interpreter announced this as a transcritical bifurcation.

4.4.2. Index scan. The search methods described above are computationally inexpensive, in that they propose a definite guess for a period-point location and apply Newton–Raphson; no search of state space is required. However, these methods are not sufficient. In the case of approaching a supercritical-flip bifurcation, for instance, the Interpreter is tracking a stable periodic point of order n that suddenly becomes unstable when an eigenvalue crosses the unit circle at -1. There should be a stable periodic point of order $2n$ on the other side of the bifurcation, but there is no obvious guess for the location of this point. The best one can do is to search a neighborhood of the bifurcation in the two-dimensional state space.

Fortunately, the two-dimensional search can be replaced by two one-dimensional searches (which is computationally much cheaper) using a method described by Hsu [12], based upon the *Poincaré index*. If g is a map from the plane to the plane and C is a closed curve, define $I(g, C)$—the index of g over C—to be the number of rotations over the path of the varying direction from x to $g(x)$:

$$I(g, C) = \frac{1}{2\pi} \oint_{c} \mathrm{d} \angle v(x), \tag{5}$$

where $v(x)$ is the vector pointing from x to $g(x)$. If C is a simple closed curve, then $I(g, C)$ is equal to the sum, over the fixed points x of g enclosed by C, of the *local indices* $I\ (g, x)$, where $I(g, x) = I(g, C_x)$ for C_x a small path enclosing the fixed point x (and no other fixed points of g). Moreover, the local index of a nondegenerate fixed point x is determined by the Jacobian of g at x:†

$$I(g, x) = \begin{cases} -1 & \text{if } \det(Dg(x) - I) < 0 \\ 1 & \text{if } \det(Dg(x) - I) > 0. \end{cases} \tag{6}$$

The formulas lead to an index scan search method for finding periodic points, essentially as described in Ref. [12]. For example, to find the stable periodic point of order $2n$ on the other side of a supercritical-flip bifurcation of a map f, begin with the known unstable order-n periodic point x_u of f and apply equation (6) with $g = f^{(2n)}$ to compute the local index of $f^{(2n)}$ at x_u. Using binary search, attempt to find circles C_r of radius r and $C_{r+\delta r}$ of radius $r + \delta r$ such that the index around C_r, as computed by equation (5), is equal to the local index of g at x_u, and the index around $C_{r+\delta r}$ is not.‡ There must be another fixed point of $f^{(2n)}$ in the annulus between C_r and $C_{r+\delta r}$. Now search for the fixed point along the annulus, choosing each successive point as the start for a Newton–Raphson iteration.

This search method is much more computationally expensive than the methods described in the previous section. The Interpreter resorts to index scanning only when none of the other methods apply, or when they have already been tried and yet the bifurcation has still not been identified.

Note. In applying equation (5) the Interpreter does not integrate angles directly, but rather uses an algorithm due to Leland [13], which computes winding numbers of counting the varying direction's signed crossings with the positive x-axis.

4.5. Criticizing local results

At this point in the process, the Interpreter has isolated a collection of families of periodic points. Each family has been fully extended, and the bifurcations at which the families terminate have been tentatively classified. These classifications are based upon purely local information—analysis of individual bifurcations, triggered by the tracking failure of individual periodic points. The local methods work well when the map f_p can be computed very accurately. In dozens of tests cases, where f_p was given by simple arithmetic formulas, local searches successfully classified all bifurcations. However, when f_p cannot be so accurately computed—as in the case of period map that must be numerically integrated—numerical errors can confuse the local methods. Therefore, after all bifurcations and families have been found, the Interpreter reexamines its classifications and attempts to identify and correct errors.

The Interpreter begins by removing families that could not be tracked, i.e. that contain only a single periodic point that could not be extended. Presumably these are flukes due to numerical error or to extremely bad local behavior. Any bifurcation involving such a family has its type reset to unknown. Two bifurcations that are connected by such a family are merged, the type of the merged bifurcation is set to unknown.

The Interpreter next tries to determine whether two bifurcations that appeared to be distinct are in reality the same bifurcation. Figure 12 shows an example of this kind of error. Bifurcations B_1

†These relations are proved by Hsu [12]. Note that the index here is not the same as the *instability index* (number of eigenvalues with magnitude > 1), which also plays a role in dynamical systems theory.

‡Occasionally, numerical error will make it impossible to start the search by locating a small circle for which the index is the same as the local index, in which case the Interpreter will abandon the index scan for this point. Also, the Interpreter will not begin to investigate a suspected period doubling if the order of the periodic point exceeds some predefined maximum (order 32 for the examples in this paper).

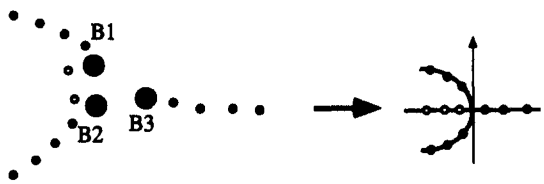

Fig. 12. Numerical error can confuse the Interpreter's local identification. Here, three "different" bifurcations B_1, B_2 and B_3, were discovered by tracking three different families of stable periodic points to slightly different locations. In fact, this is probably a single supercritical pitchfork bifurcation, and the two unstable periodic points at B_1 and B_2 belong to the same family. The Interpreter reexamines its local classifications and attempts to recognize such errors.

and B_2 were encountered by tracking stable points in the direction of increasing p, and each tracking failed when an eigenvalue approached 1. At each bifurcation, the Interpreter found an unstable periodic point before the bifurcation, and so classified each bifurcation as a fold. Also, there is a nearby bifurcation B_3 that was found by tracking a stable periodic point in the direction of decreasing p until an eigenvalue approached 1. At B_3 none of the search methods produced a new periodic point, and the Interpreter classified B_3 as unknown. In fact, however, these three bifurcations are the same one. It is a supercritical pitchfork, and the unstable points at B_1 and B_2 belong to the same family of unstable points. Due to numerical error, the bifurcations were found at slightly different locations—locations sufficiently different to be outside the predefined tolerance that the Interpreter uses to check whether a point being tracked has merged into a previously-discovered bifurcation.

One might argue that such errors could be avoided simply by choosing a larger tolerance during tracking. But that would lead, in other situations, to merging nearby bifurcations that are in fact distinct. Instead, the Interpreter waits until all bifurcations have been located and then checks for appropriate merges. Suppose that B was identified as a known type, with all required stable and unstable periodic points discovered. Examining the list of known bifurcations in Fig. 7 shows that B cannot possibly merge with another bifurcation to form another one of known type—unless B was identified as a fold, in which case B may actually be a pitchfork or a transcritical bifurcation at which the Interpreter's search methods failed to discover all the local families.

Thus, the Interpreter reexamines only the folds and unknown bifurcations to find pairs that are suspiciously close. During the tracking phase, each bifurcation's parameter value p was localized to an interval of size δp. The Interpreter uses equation (4) to approximate the change δx in periodic-point location corresponding to a parameter change of size δp. Two bifurcations are merged if their parameter intervals overlap and their periodic points are within a distance of the magnitude of δx.

Finally, the Interpreter reexamines all bifurcations that are still classified as unknown, and it opportunistically tries to identify them based on very liberal criteria. For example, a stable family of order n, tracked to a bifurcation with eigenvalue approaching -1, is probably a flip, even if the Interpreter was unable to find the point of order $2n$. The Interpreter checks the bifurcation at the other end of the order-n family, and if this is a supercritical flip, it marks the unknown bifurcation as probable supercritical flip, on the grounds that chains of supercritical flips (period-doubling cascades) are fairly common. These opportunistic identifications are flagged as having been performed at this stage rather than during tracking, since they are presumably less reliable than identifications that have been confirmed by finding all the expected local periodic points.

4.6. Generating the family report

Once bifurcations and families have been isolated and classified, it is a simple matter to produce a family report. The Interpreter first scans the bifurcations, and annotates the ones involving symmetric pairs or period-doubling cascades. Any chain of two or more supercritical flips is marked as a period-doubling cascade. Symmetric pairs are recognized as evolving from supercritical pitchforks whose two branches lead to sequences of bifurcations B_i^0 and B_i^1, where, for each i, B_i^0 and B_i^1 have the same type and are located at the same value of p.

The annotated bifurcations and families can be viewed as a graph, where the nodes are the bifurcations and the arcs are the families. The Interpreter separates these into connected components of the graph, which form the classes of families. Then it assigns names to the classes and to the families and bifurcations within each class, numbered in the direction of increasing p. The final family report is a list, for each class, of the families or groups of families in the class. The list for class A of the Duffing system analysis, for instance, was shown in Section 3.2.

The bifurcation data structures also retain information that was generated during the tracking and recognition phases. Here, for example, is the first bifurcation in class E of the Duffing system (Fig. 1). This is a supercritical pitchfork at which family E_0 splits into $E_{1,0}$ and $E_{1,1}$:

```
(bifurcation:bif.1
 (parameter-range (21.3493 21.3503))
 (families-vanishing (#[family:E.0]))
 (families-appearing (#[family:E.1.0] #[family:E.1.1]))
 (id-info
  ((bif-id-info (#[id-info] (cannot identify saddle bifurcation)))
   (bif-id-info (#[id-info] (cannot identify saddle bifurcation)))))
 (comments ((merged-to-incorporate #[bifurcation: bif.9])))
 (classified-by-critic true)
 (type supercritical-pitchfork)
 (symmetric-pair (split to form symmetric pair))
 (direction parameter-increasing))
```

The **bif-id-info** slot here contains information generated by the local classification algorithm (Section 4.4), including pointers to the stable and unstable periodic points discovered by the local search, together with a comment that the local recognition algorithm failed—the Interpreter found that this was a saddle bifurcation (eigenvalue approaching 1) but was unable to identify it. There are two **bif-id** entries here because the bifurcation was formed by merging two bifurcations, each with a local identification that failed, as described in Section 4.5. As noted, the final classification of this bifurcations as a supercritical flip was accomplished in the critical phase, after the merge.

Finally, the lists of families and bifurcations are traversed by a simple text generator, or a graphics generator, to produce the kind of text and graphical output illustrated in Section 2.1. Not all the information in the lists is reflected in this output, but the information remains as part of the family report, available for further processing.

5. DISCUSSION

The ability to progress beyond raw numerical data to uncover underlying qualitative phenomena is sometimes called insight. It is intriguing that a degree of this "insight" can be achieved automatically, with only a few simple methods. Although the Bifurcation Interpreter combines techniques from numerical computing, symbolic algebra, and knowledge encoding, it draws upon each of these areas to only a very limited degree. Conversely, considering each of these areas in turn immediately suggests extensions and improvements to the present implementation.

The most obvious extension is to higher-dimensional systems. Note that the index scan method for searching for periodic points near bifurcations (Section 4.4) is the only algorithm in the current implementation that relies on the fact that the state space is two-dimensional. Additional computational methods for "extending a periodic point through a bifurcation," which may be successful in higher dimensions, are discussed by Parker and Chua [14].† Exploring systems where the parameter space has more than one dimension is more challenging; for one thing, the generic bifurcations of such systems have not been mathematically classified. But it should be possible to extend the Interpreter to map out these higher-dimensional families as well, and to identify the bifurcations encountered, at least along codimension-1 subspaces of the parameter space.

†Parker and Chua's INSITE system [14, 15] includes a wide assortment of numerical methods that are useful in the study of nonlinear dynamics, including routines that locate individual periodic points and track them to bifurcations, in much the same way as the tracking phase of the Interpreter (Section 4.3).

In the area of improving the numerical methods, one straightforward extension is to have the Interpreter track and report on families of unstable periodic points, in addition to the stable ones. This would produce a more complete picture of the dynamical system. A second improvement, in the case of ordinary differential equations, is to have the program use information based upon the complete trajectories of points rather than just the period map. For instance, in testing for a symmetric pair of families, the Interpreter should verify the symmetry of the trajectories. In addition, much work needs to be done in choosing the tolerances of the numerical routines—such as the minimum distance at which two bifurcations are considered to be distinct. One possible approach here is to run the program repeatedly with different tolerances until the high-level qualitative report stabilizes.

The Interpreter's symbolic methods are likewise very restricted. It currently uses symbolic algebra only to derive formulas for Jacobians and sensitivities. Another obvious application for symbolic processing is to handle symmetries. For instance, one should expect to find pitchfork bifurcations only if the system is invariant under a symmetry (as mentioned in Section 3.1). Many such symmetries can be identified by examining the defining equations, and, having found a symmetry, the Interpreter could use this to avoid redundant computations. More significantly, in the case where f is computed by algebraic formulas rather than by numerical integration, one can attempt to recognize and classify bifurcations purely symbolically, by deriving the normal forms of the maps. Such computations, using the Macsyma symbolic algebra system, have been demonstrated by Rand and Keith [16].

It is in the area of incorporating more explicit knowledge about dynamics that the most important work remains to be done. For instance, beyond identifying collections of fixed points, the present Interpreter pays no attention to the geometry of the phase space. Thus, it can hope to recognize only those bifurcations that occur at individual fixed points. Dealing with more global reorganizations of phase space, such as saddle connections [2], will require a more explicit representation of phase-space geometry. Additionally, representing geometric relations between nearby trajectories can enable a program to apply powerful consistency constraints in guiding its exploration of the phase space. Programs that automatically investigate dynamical systems by "looking at" phase-space geometry have been demonstrated, for Hamiltonian systems with 3 degrees-of-freedom, by Yip [17–19], and for planar vector fields, by Sacks [20].

A final important area for further work is to apply the Interpreter to systems that are derived from models of physical situations, rather than presented *a priori* as equations, and to have the program formulate its interpretations in terms of underlying physical phenomena rather than only in terms of the bare mathematics. For example, if a nonlinear oscillator describes the rolling motion of a ship, then a transition to chaos can be interpreted as the possibility of capsizing [21]. A program that describes chemical oscillations by making qualitative investigations of the corresponding dynamical system is being developed by Eisenberg [22].

The Bifurcation Interpreter, even with its present limitations, lets us imagine a new category of programs that formulate numerical results in qualitative terms, thereby enabling their users to control computational experiments in terms of high-level behavioral descriptions (see Ref. [23] for further examples). Over the past 40 years, increasingly sophisticated numerical tools have radically transformed the role of mathematical modeling in science and engineering. Looking forward to combining these numerical techniques with symbolic and knowledge-based methods, shows that this tranformation has only just begun.

Acknowledgements—This research was carried out as part of the MIT Project on Mathematics and Computation, and I am indebted to many members of the project for helpful discussions and suggestions about dynamics, and for maintaining the MIT Scheme computing system, with which the Interpreter was developed. Most especially, I would like to thank Gerald Sussman, who collaborated on the symbolic algebra system and the fixed-point tracking algorithms, and Ognen Nastov, who designed and implemented the graphics routines that generate pictures from the Interpreter's family reports.

REFERENCES

1. B. McCormick, T. Desanti and M. Brown (Eds), Visualization in Scientific Computing (Special Issue). *Computer Graph.* **21**(6), (1987).
2. J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer, New York (1983).

3. J. M. T. Thompson and H. B. Stewart, *Nonlinear Dynamics and Chaos*. Wiley, New York (1986).
4. V. Franceschini, Two models of truncated Navier–Stokes equations on a two-dimensional torus. *Phys. Fluids* **26**(2), 433–447 (1983).
5. Y. Ueda, Steady motions exhibited by Duffing's equation: a picture book of regular and chaotic motions. In *New Approaches to Nonlinear Problems in Dynamics* (Ed. P. J. Holmes), pp. 311–322. SIAM, Philadelphia, Pa (1980).
6. M. J. Feigenbaum, Universal behavior in nonlinear systems. *Los Alamos Sci.* **1**, 4–27 (1980).
7. H. Abelson and G. J. Sussman, The dynamicist's workbench I: automatic preparation of numerical experiments. MIT Artificial Intelligence Lab. Memo No. 955 (May 1987).
8. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes: the Art of Scientific Computing*. Cambridge Univ. Press, Cambs. (1985).
9. C. S. Hsu, *Cell-to-cell Mapping: a Method of Global Analysis for Nonlinear Systems*. Springer, New York (1987).
10. C. S. Hsu and R. S. Guttalu, An unravelling algorithm for global analysis of dynamical systems: an application of cell-to-cell mappings. *Trans. ASME* **47**, 940–948 (1980).
11. F. Preparata and M. Shamos, *Computational Geometry*. Springer, New York (1985).
12. C. S. Hsu, A theory of index for point mapping dynamical systems. *J. appl. Mech.* **47**, 185–190 (1980).
13. K. O. Leland, An algorithm for winding numbers of closed polygonal paths. *Maths Comput.* **29**, 554–558 (1975).
14. T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*. Springer, New York (1990).
15. T. S. Parker and L. O. Chua, INSITE—a software toolkit for the analysis of nonlinear dynamical systems. *Proc. IEEE* **75**(8), 1081–1089 (1987).
16. R. H. Rand and W. L. Keith, Normal form and center manifold calculations on Macsyma. In *Applications of Computer Algebra* (Ed. R. Pavelle), pp. 309–328. Kluwer, Boston, Mass. (1985).
17. K. Yip, Extracting qualitative dynamics from numerical experiments. AAAI Report (1987).
18. K. Yip, Generating global behaviors using deep knowledge of local dynamics. AAAI Report (1988).
19. K. Yip, KAM: automatic planning and interpretation of numerical experiments using geometrical methods. Ph.D. Dissertation, Dept of Electrical Engineering and Computer Science, MIT, Cambridge, Mass. (Aug. 1989).
20. E. Sacks, Qualitative analysis of continuous dynamic systems by intelligent numeric experimentation. To appear.
21. A. H. Nayfeh and N. E. Sanchez, Chaos and dynamic instability in the rolling motion of ships. Presented at the *17th ONR Symp. on Naval Hydrodynamics* (1988).
22. M. Eisenberg, Descriptive simulation: combining symbolic and numerical methods in the analysis of chemical reaction mechanisms. MIT Artificial Intelligence Lab. Memo No. 1171 (Sept. 1989).
23. H. Abelson, M. Eisenberg, M. Halfant, J. Katzenelson, E. Sacks, G. J. Sussman, J. Wisdom and K. Yip, Intelligence in scientific computing. *Communs ACM* **32**(5), 546–562 (1989).