Physica 16D (1985) 285 - 317
North-Holland, Amsterdam

DETERMINING LYAPUNOV EXPONENTS FROM A TIME SERIES

Alan WOLF†, Jack B. SWIFT, Harry L. SWINNEY and John A. VASTANO
Department of Physics, University of Texas, Austin, Texas 78712, USA

Received 18 October 1984

We present the first algorithms that allow the estimation of non-negative Lyapunov exponents from an *experimental* time series. Lyapunov exponents, which provide a qualitative and quantitative characterization of dynamical behavior, are related to the exponentially fast divergence or convergence of nearby orbits in phase space. A system with one or more positive Lyapunov exponents is defined to be chaotic. Our method is rooted conceptually in a previously developed technique that could only be applied to analytically defined model systems: we monitor the *long-term* growth rate of *small* volume elements in an attractor. The method is tested on model systems with known Lyapunov spectra, and applied to data for the Belousov–Zhabotinskii reaction and Couette–Taylor flow.

Contents

1. Introduction

Convincing evidence for deterministic chaos has come from a variety of recent experiments [1–6] on dissipative nonlinear systems; therefore, the question of detecting and quantifying chaos has become an important one. Here we consider the spectrum of Lyapunov exponents [7–10], which has proven to be the most useful dynamical diagnostic for chaotic systems. Lyapunov exponents are the average exponential rates of divergence or convergence of nearby orbits in phase space. Since nearby orbits correspond to nearly identical states, exponential orbital divergence means that systems whose initial differences we may not be able to resolve will soon behave quite differently – predictive ability is rapidly lost. Any system containing at least one positive Lyapunov exponent is defined to be chaotic, with the magnitude of the exponent reflecting the time scale on which system dynamics become unpredictable [10].

For systems whose equations of motion are explicitly known there is a straightforward technique [8, 9] for computing a complete Lyapunov spectrum. This method cannot be applied directly to experimental data for reasons that will be discussed later. We will describe a technique which for the first time yields estimates of the non-negative Lyapunov exponents from finite amounts of experimental data.

A less general procedure [6, 11–14] for estimating only the dominant Lyapunov exponent in experimental systems has been used for some time. This technique is limited to systems where a well-defined one-dimensional (1-D) map can be recovered. The technique is numerically unstable and the literature contains several examples of its improper application to experimental data. A discussion of the 1-D map calculation may be found

†Present address: The Cooper Union, School of Engineering, N.Y., NY 10003, USA.

*The reader may wish to skip the starred sections at a first reading.

0167-2789/85/$03.30

in ref. 13. In ref. 2 we presented an unusually robust 1-D map exponent calculation for experimental data obtained from a chemical reaction.

Experimental data inevitably contain external noise due to environmental fluctuations and limited experimental resolution. In the limit of an infinite amount of noise-free data our approach would yield Lyapunov exponents by definition. Our ability to obtain good spectral estimates from experimental data depends on the quantity and quality of the data as well as on the complexity of the dynamical system. We have tested our method on model dynamical systems with known spectra and applied it to experimental data for chemical [2, 13] and hydrodynamic [3] strange attractors.

Although the work of characterizing chaotic data is still in its infancy, there have been many approaches to quantifying chaos, e.g., fractal power spectra [15], entropy [16–18, 3], and fractal dimension [proposed in ref. 19, used in ref. 3–5, 20, 21]. We have tested many of these algorithms on both model and experimental data, and despite the claims of their proponents we have found that these approaches often fail to characterize chaotic data. In particular, parameter independence, the amount of data required, and the stability of results with respect to external noise have rarely been examined thoroughly.

The spectrum of Lyapunov exponents will be defined and discussed in section 2. This section includes table I which summarizes the model systems that are used in this paper. Section 3 is a review of the calculation of the complete spectrum of exponents for systems in which the defining differential equations are known. Appendix A contains Fortran code for this calculation, which to our knowledge has not been published elsewhere. In section 4, an outline of our approach to estimating the non-negative portion of the Lyapunov exponent spectrum is presented. In section 5 we describe the algorithms for estimating the two largest exponents. A Fortran program for determining the largest exponent is contained in appendix B. Our algorithm requires input parameters whose selection is discussed in section 6. Section 7 concerns sources of error in the calculations and the quality and quantity of data required for accurate exponent estimation. Our method is applied to model systems and experimental data in section 8, and the conclusions are given in section 9.

2. The Lyapunov spectrum defined

We now define [8, 9] the spectrum of Lyapunov exponents in the manner most relevant to spectral calculations. Given a continuous dynamical system in an n-dimensional phase space, we monitor the long-term evolution of an *infinitesimal* n-sphere of initial conditions; the sphere will become an n-ellipsoid due to the locally deforming nature of the flow. The ith one-dimensional Lyapunov exponent is then defined in terms of the length of the ellipsoidal principal axis $p_i(t)$:

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \log_2 \frac{p_i(t)}{p_i(0)}, \tag{1}$$

where the λ_i are ordered from largest to smallest†. Thus the Lyapunov exponents are related to the expanding or contracting nature of different directions in phase space. Since the orientation of the ellipsoid changes continuously as it evolves, the directions associated with a given exponent vary in a complicated way through the attractor. One cannot, therefore, speak of a well-defined direction associated with a given exponent.

Notice that the linear extent of the ellipsoid grows as $2^{\lambda_1 t}$, the area defined by the first two principal axes grows as $2^{(\lambda_1+\lambda_2)t}$, the volume defined by the first three principal axes grows as $2^{(\lambda_1+\lambda_2+\lambda_3)t}$, and so on. This property yields another definition of the spectrum of exponents:

†While the existence of this limit has been questioned [8, 9, 22], the fact is that the orbital divergence of *any* data set may be quantified. Even if the limit does not exist for the underlying system, or cannot be approached due to having finite amounts of noisy data, Lyapunov exponent estimates could still provide a useful characterization of a given data set. (See section 7.1.)

the sum of the first j exponents is defined by the long term exponential growth rate of a j-volume element. This alternate definition will provide the basis of our spectral technique for experimental data.

Any continuous time-dependent dynamical system without a fixed point will have at least one zero exponent [22], corresponding to the slowly changing magnitude of a principal axis tangent to the flow. Axes that are on the average expanding (contracting) correspond to positive (negative) exponents. The sum of the Lyapunov exponents is the time-averaged divergence of the phase space velocity; hence any dissipative dynamical system will have at least one negative exponent, the sum of all of the exponents is negative, and the post-transient motion of trajectories will occur on a zero volume limit set, an attractor.

The exponential expansion indicated by a positive Lyapunov exponent is incompatible with motion on a bounded attractor unless some sort of *folding* process merges widely separated trajectories. Each positive exponent reflects a "direction" in which the system experiences the repeated stretching and folding that decorrelates nearby states on the attractor. Therefore, the long-term behavior of an initial condition that is specified with *any* uncertainty cannot be predicted; this is chaos. An attractor for a dissipative system with one or more positive Lyapunov exponents is said to be "strange" or "chaotic".

The signs of the Lyapunov exponents provide a qualitative picture of a system's dynamics. One-dimensional maps are characterized by a single Lyapunov exponent which is positive for chaos, zero for a marginally stable orbit, and negative for a periodic orbit. In a three-dimensional continuous dissipative dynamical system the only possible spectra, and the attractors they describe, are as follows: $(+,0,-)$, a strange attractor; $(0,0,-)$, a two-torus; $(0,-,-)$, a limit cycle; and $(-,-,-)$, a fixed point. Fig. 1 illustrates the expanding, "slower than exponential," and contracting character of the flow for a three-dimensional system, the Lorenz model [23]. (All of the model systems that we will discuss are defined in table I.) Since Lyapunov exponents involve long-time averaged behavior, the short segments of the trajectories shown in the figure cannot be expected to accurately characterize the positive, zero, and negative exponents; nevertheless, the three distinct types of behavior are clear. In a continuous four-dimensional dissipative system there are three possible types of strange attractors: their Lyapunov spectra are $(+,+,0,-)$, $(+,0,0,-)$, and $(+,0,-,-)$. An example of the first type is Rossler's hyperchaos attractor [24] (see table I). For a given system a change in parameters will generally change the Lyapunov spectrum and may also change both the type of spectrum and type of attractor.

The magnitudes of the Lyapunov exponents *quantify* an attractor's dynamics in information theoretic terms. The exponents measure the rate at which system processes create or destroy information [10]; thus the exponents are expressed in bits of information/s or bits/orbit for a continuous system and bits/iteration for a discrete system. For example, in the Lorenz attractor the positive exponent has a magnitude of 2.16 bits/s (for the parameter values shown in table I). Hence if an initial point were specified with an accuracy of one part per million (20 bits), the future behavior could not be predicted after about 9 s [20 bits/(2.16 bits/s)], corresponding to about 20 orbits. After this time the small initial uncertainty will essentially cover the entire attractor, reflecting 20 bits of new information that can be gained from an additional measurement of the system. This new information arises from scales smaller than our initial uncertainty and results in an inability to specify the state of the system except to say that it is somewhere on the attractor. This process is sometimes called an information gain – reflecting new information from the heat bath, and sometimes is called an information loss – bits shifted out of a phase space variable "register" when bits from the heat bath are shifted in.

The average rate at which information contained in transients is lost can be determined from

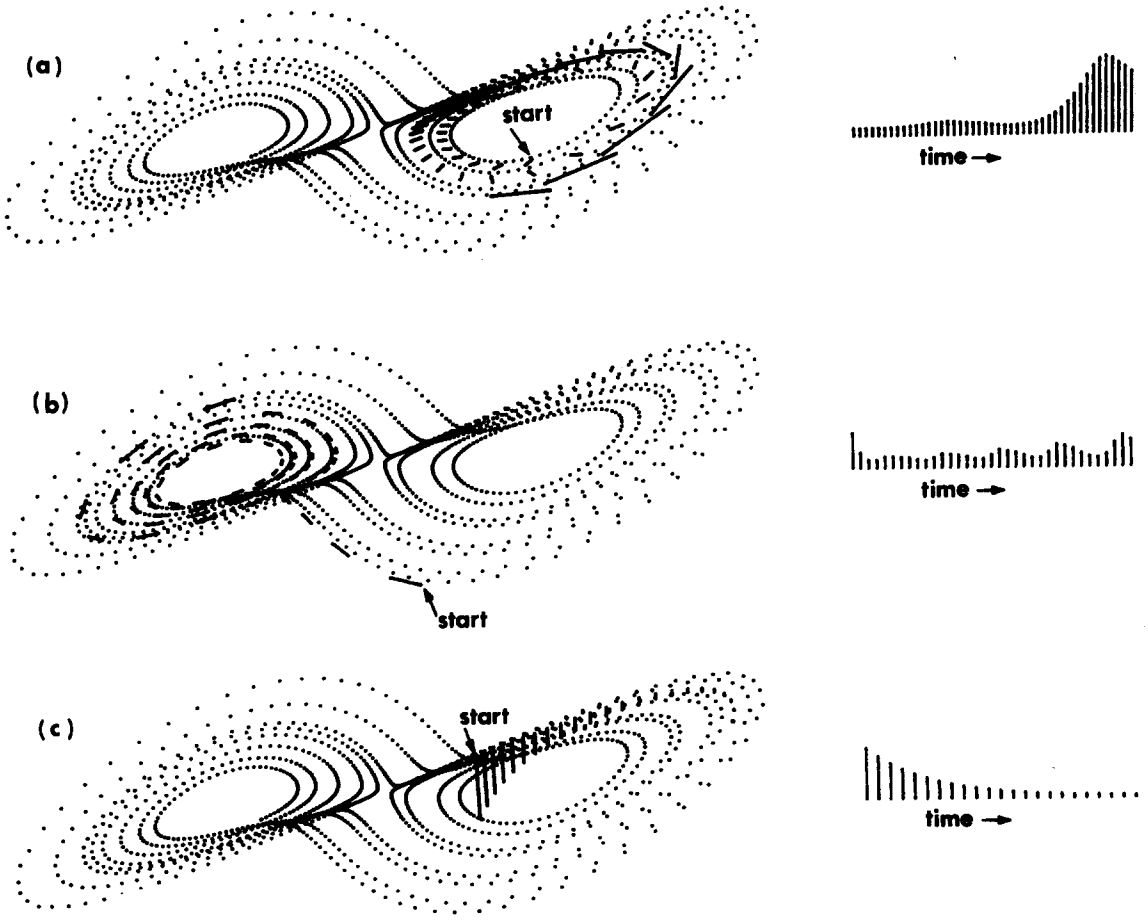

Fig. 1. The short term evolution of the separation vector between three carefully chosen pairs of nearby points is shown for the Lorenz attractor. a) An expanding direction ($\lambda_1 > 0$); b) a "slower than exponential" direction ($\lambda_2 = 0$); c) a contracting direction ($\lambda_3 < 0$).

the negative exponents. The asymptotic decay of a perturbation to the attractor is governed by the least negative exponent, which should therefore be the easiest of the negative exponents to estimate†.

For the Lorenz attractor the negative exponent is so large that a perturbed orbit typically becomes indistinguishable from the attractor, by "eye", in less than one mean orbital period (see fig. 1).

†We have been quite successful with an algorithm for determining the dominant (smallest magnitude) negative exponent from pseudo-experimental data (a single time series extracted from the solution of a model system and treated as an experimental observable) for systems that are nearly integer-dimensional. Unfortunately, our approach, which involves measuring the mean decay rate of many induced perturbations of the dynamical system, is unlikely to work on many experimental systems. There are several fundamental problems with the calculation of negative exponents from experimental data, but of greatest importance is that *post-transient* data may not contain resolvable negative exponent information and *perturbed* data must reflect properties of the unperturbed system, that is, perturbations must only change the state of the system (current values of the dynamical variables). The response of a physical system to a non-delta function perturbation is difficult to interpret, as an orbit separating from the attractor may reflect either a locally repelling region of the attractor (a positive contribution to the negative exponent) or the finite rise time of the perturbation.

Table I
The model systems considered in this paper and their Lyapunov spectra and dimensions as computed from the equations of motion

System	Parameter values	Lyapunov spectrum (bits/s)†	Lyapunov dimension‡
Hénon: [25]			
$X_{n+1} = 1 - aX_n^2 + Y_n$ $Y_{n+1} = bX_n$	$a = 1.4$ $b = 0.3$	$\lambda_1 = 0.603$ $\lambda_2 = -2.34$ (*bits/iter.*)	1.26
Rossler–chaos: [26]			
$\dot{X} = -(Y+Z)$ $\dot{Y} = X + aY$ $\dot{Z} = b + Z(X-c)$	$a = 0.15$ $b = 0.20$ $c = 10.0$	$\lambda_1 = 0.13$ $\lambda_2 = 0.00$ $\lambda_3 = -14.1$	2.01
Lorenz: [23]			
$\dot{X} = \sigma(Y-X)$ $\dot{Y} = X(R-Z) - Y$ $\dot{Z} = XY - bZ$	$\sigma = 16.0$ $R = 45.92$ $b = 4.0$	$\lambda_1 = 2.16$ $\lambda_2 = 0.00$ $\lambda_3 = -32.4$	2.07
Rossler–hyperchaos: [24]			
$\dot{X} = -(Y+Z)$ $\dot{Y} = X + aY + W$ $\dot{Z} = b + XZ$ $\dot{W} = cW - dZ$	$a = 0.25$ $b = 3.0$ $c = 0.05$ $d = 0.5$	$\lambda_1 = 0.16$ $\lambda_2 = 0.03$ $\lambda_3 = 0.00$ $\lambda_4 = -39.0$	3.005
Mackey–Glass: [27]			
$\dot{X} = \dfrac{aX(t+s)}{1+[X(t+s)]^c} - bX(t)$	$a = 0.2$ $b = 0.1$ $c = 10.0$ $s = 31.8$	$\lambda_1 = 6.30\text{E}{-3}$ $\lambda_2 = 2.62\text{E}{-3}$ $\lvert\lambda_3\rvert < 8.0\text{E}{-6}$ $\lambda_4 = -1.39\text{E}{-2}$	3.64

†A mean orbital period is well defined for Rossler chaos (6.07 seconds) and for hyperchaos (5.16 seconds) for the parameter values used here. For the Lorenz attractor a characteristic time (see footnote – section 3) is about 0.5 seconds. Spectra were computed for each system with the code in appendix A.
‡As defined in eq. (2).

The Lyapunov spectrum is closely related to the fractional dimension of the associated strange attractor. There are a number [19] of different fractional-dimension-like quantities, including the fractal dimension, information dimension, and the correlation exponent; the difference between them is often small. It has been conjectured by Kaplan and Yorke [28, 29] that the information dimension d_f is related to the Lyapunov spectrum by the equation

$$d_f = j + \frac{\sum_{i=1}^{j} \lambda_i}{|\lambda_{j+1}|}, \tag{2}$$

where j is defined by the condition that

$$\sum_{i=1}^{j} \lambda_i > 0 \quad \text{and} \quad \sum_{i=1}^{j+1} \lambda_i < 0. \tag{3}$$

The conjectured relation between d_f (a *static* property of an attracting set) and the Lyapunov

exponents appears to be satisfied for some model systems [30]. The calculation of dimension from this equation requires knowledge of all but the most negative Lyapunov exponents.

3. Calculation of Lyapunov spectra from differential equations

Our algorithms for computing a non-negative Lyapunov spectrum from experimental data are inspired by the technique developed independently by Bennetin et al. [8] and by Shimada and Nagashima [9] for determining a complete spectrum from a set of differential equations. Therefore, we describe their calculation (for brevity, the ODE approach) in some detail.

We recall that Lyapunov exponents are defined by the long-term evolution of the axes of an infinitesimal sphere of states. This procedure could be implemented by defining the principal axes with initial conditions whose separations are as small as computer limitations allow and evolving these with the nonlinear equations of motion. One problem with this approach is that in a chaotic system we cannot guarantee the condition of small separations for times on the order of hundreds of orbital periods†, needed for convergence of the spectrum.

This problem may be avoided with the use of a phase space plus tangent space approach. A "fiducial" trajectory (the center of the sphere) is defined by the action of the nonlinear equations of motion on some initial condition. Trajectories of points on the surface of the sphere are defined by the action of the linearized equations of motion on points infinitesimally separated from the fiducial trajectory. In particular, the principal axes are defined by the evolution via the linearized equations of an initially orthonormal vector frame anchored to the fiducial trajectory. By definition, *principal axes defined by the linear system are always infinitesimal relative to the attractor*. Even in the linear system, principal axis vectors diverge in magnitude, but this is a problem only because computers have a limited dynamic range for storing numbers. This divergence is easily circumvented. What has been avoided is the serious problem of principal axes finding the global "fold" when we really only want them to probe the local "stretch."

To implement this procedure the fiducial trajectory is created by integrating the nonlinear equations of motion for some post-transient initial condition. Simultaneously, the linearized equations of motion are integrated for n different initial conditions defining an arbitrarily oriented frame of n orthonormal vectors. We have already pointed out that each vector will diverge in magnitude, but there is an additional singularity – in a chaotic system, each vector tends to fall along the local direction of most rapid growth. Due to the finite precision of computer calculations, the collapse toward a common direction causes the tangent space orientation of all axis vectors to become indistinguishable. These two problems can be overcome by the repeated use of the Gram–Schmidt reorthonormalization (GSR) procedure on the vector frame:

Let the linearized equations of motion act on the initial frame of orthonormal vectors to give a set of vectors $\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n\}$. (The desire of each vector to align itself along the λ_1 direction, and the orientation-preserving properties of GSR mean that the initial labeling of the vectors may be done arbitrarily.) Then GSR provides the following orthonormal set $\{\boldsymbol{v}_1',\ldots,\boldsymbol{v}_n'\}$:

$$\begin{aligned}
\boldsymbol{v}_1' &= \frac{\boldsymbol{v}_1}{\|\boldsymbol{v}_1\|}, \\
\boldsymbol{v}_2' &= \frac{\boldsymbol{v}_2 - \langle \boldsymbol{v}_2, \boldsymbol{v}_1' \rangle \boldsymbol{v}_1'}{\|\boldsymbol{v}_2 - \langle \boldsymbol{v}_2, \boldsymbol{v}_1' \rangle \boldsymbol{v}_1'\|}, \\
&\vdots \\
\boldsymbol{v}_n' &= \frac{\boldsymbol{v}_n - \langle \boldsymbol{v}_n, \boldsymbol{v}_{n-1}' \rangle \boldsymbol{v}_{n-1}' - \cdots - \langle \boldsymbol{v}_n, \boldsymbol{v}_1' \rangle \boldsymbol{v}_1'}{\|\boldsymbol{v}_n - \langle \boldsymbol{v}_n, \boldsymbol{v}_{n-1}' \rangle \boldsymbol{v}_{n-1}' - \cdots - \langle \boldsymbol{v}_n, \boldsymbol{v}_1' \rangle \boldsymbol{v}_1'\|},
\end{aligned} \tag{4}$$

†Should the mean orbital period not be well-defined, a characteristic time can be either the mean time between intersections of a Poincaré section or the time corresponding to a dominant power spectral feature.

where $\langle\ ,\ \rangle$ signifies the inner product. The frequency of reorthonormalization is not critical, so long as neither the magnitude nor the orientation divergences have exceeded computer limitations. As a rule of thumb, GSR is performed on the order of once per orbital period.

We see that GSR never affects the direction of the first vector in a system, so this vector tends to seek out the direction in tangent space which is most rapidly growing (components along other directions are either growing less rapidly or are shrinking). The second vector has its component along the direction of the first vector removed, and is then normalized. Because we are changing its direction, vector v_2 is not free to seek out the most rapidly growing direction. Because of the manner in which we are changing it, it also is not free to seek out the second most rapidly growing direction†. Note however that the vectors v_1' and v_2' span the same two-dimensional subspace as the vectors v_1 and v_2. *In spite of repeated vector replacements, the space these vectors define continually seeks out the two-dimensional subspace that is most rapidly growing.* The area defined by these vectors is proportional to $2^{(\lambda_1+\lambda_2)t}$ [8]. The length of vector v_1 is proportional to $2^{\lambda_1 t}$ so that monitoring length and area growth allows us to determine both exponents. In practice, as v_1' and v_2' are orthogonal, we may determine λ_2 directly from the mean rate of growth of the projection of vector v_2 on vector v_2'. In general, the subspace spanned by the first k vectors is unaffected by GSR so that the long-term evolution of the k-volume defined by these vectors is proportional to 2^{μ} where $\mu = \sum_{i=1}^{k}\lambda_i t$. Projection of the evolved vectors onto the new orthonormal frame correctly updates the rates of growth of each of the first k-principal axes in turn, providing estimates of the k largest Lyapunov exponents. Thus GSR allows the integration of the vector frame for as long as is required for spectral convergence.

Fortran code for the ODE procedure appears in appendix A. We illustrate the use of this procedure for the Rossler attractor [26]. The spectral calculation requires the integration of the 3 equations of motion and 9 linearized equations for on the order of 100 orbits of model time (a few cpu minutes on a VAX 11/780) to obtain each exponent to within a few percent of its asymptotic value. In practice we consider the asymptotic value to be attained when the mandatory zero exponent(s) are a few orders of magnitude smaller than the smallest positive exponent. The convergence rate of zero and positive exponents is about the same, and is much slower than the convergence rate of negative exponents. Negative exponents arise from the nearly uniform attractiveness of the attractor which can often be well estimated from a few passes around an attractor, non-negative exponents arise from a once-per-orbit stretch and fold process that must be sampled on the order of hundreds of times (or more) for reasonable convergence.

The method we have described for finding Lyapunov exponents is perhaps more easily understood for a discrete dynamical system. Here we consider the Hénon map [25] (see table I). The linearization of this map is

$$\begin{pmatrix}\delta X_{n+1}\\ \delta Y_{n+1}\end{pmatrix} = J_n\begin{pmatrix}\delta X_n\\ \delta Y_n\end{pmatrix}, \tag{5}$$

where

$$J_n = \begin{bmatrix}-2.8X_n & 1\\ 0.3 & 0\end{bmatrix} \tag{6}$$

and X_n is the $(n-1)$st iterate of an arbitrary initial condition X_1.

An orthonormal frame of principal axis vectors such as $((0,1), (1,0))$ is evolved by applying the product Jacobian to each vector. For either vector

†This is clear when we consider that we may obtain different directions of vector v_2 at some specified time if we exercise our freedom to choose the intermediate times at which GSR is performed. That is, beginning with a specified v_1 and v_2 at time t_i, we may perform replacements at times t_{i+1} and t_{i+2}, obtaining the vectors v_1', v_2' and then v_1'', v_2'' or we may propagate directly to time t_{i+2}, obtaining v_1^*, v_2^*. v_2'' and v_2^* are *not* parallel; therefore, the details of propagation and replacement determine the orientation of v_2.

the operation may be written in two different ways. For example, for the vector (0, 1) we have

$$\begin{pmatrix} \delta X_n \\ \delta Y_n \end{pmatrix} = J_{n-1}\left(J_{n-2} \cdots J_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right), \tag{7}$$

or, by regrouping the terms,

$$\begin{pmatrix} \delta X_n \\ \delta Y_n \end{pmatrix} = [J_{n-1} J_{n-2} \cdots J_1] \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{8}$$

In eq. (7) the latest Jacobi matrix multiplies each *current* axis vector, which is the initial vector multiplied by all previous Jacobi matrices. The magnitude of each current axis vector diverges, and the angular separation between the two vectors goes to zero. Fig. 2 shows that divergent behavior is visible within a few iterations. GSR corresponds to the replacement of each current axis vector. Lyapunov exponents are computed from the growth rate of the length of the first vector and the growth rate of the area defined by both vectors.

In eq. (8) the *product* Jacobian acts on each of the *initial* axis vectors. The columns of the product matrix converge to large multiples of the eigenvector of the biggest eigenvalue, so that elements of the matrix diverge and the matrix becomes singular. Here GSR corresponds to factoring out a large scalar multiplier of the matrix to prevent the magnitude divergence, and doing row reduction with pivoting to retain the linear independence of the columns. Lyapunov exponents are computed from the eigenvalues of the long-time product matrix†.

We emphasize that Lyapunov exponents are not local quantities in either the spatial or temporal sense. Each exponent arises from the average, with respect to the dynamical motion, of the local deformation of various phase space directions. Each is determined by the *long-time* evolution of a *single* volume element. Attempts to estimate exponents by averaging local contraction and expansion rates of phase space are likely to fail at the point where these contributions to the exponents are combined. In fig. 3a we show vector v_1' at each renormalization step for the Lorenz attractor over the course of several hundred orbits [32]. The apparent multivaluedness of the most rapidly growing direction (in some regions of the attractor) shows that this direction is not simply a function of position on the attractor. While this direction is often nearly parallel to the flow on the Lorenz attractor (see fig. 3b) it is usually nearly transverse to the flow for the Rossler attractor. We conclude that exponent calculation by averaging *local* divergence estimates is a dangerous procedure.

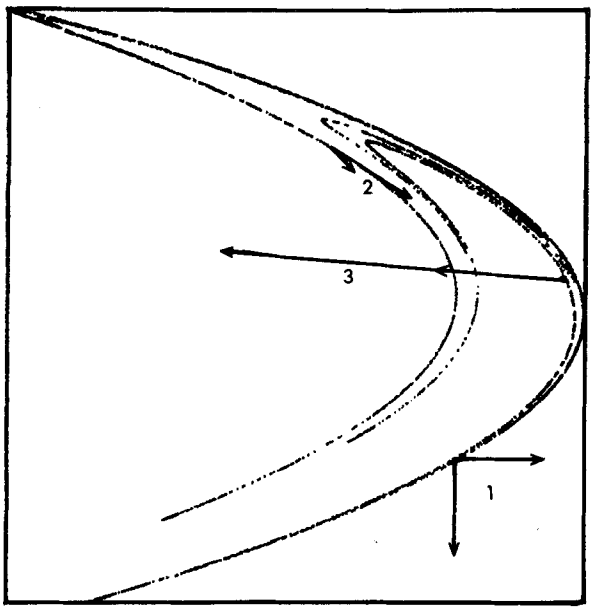

Fig. 2. The action of the product Jacobian on an initially orthonormal vector frame is illustrated for the Hénon map: (1) initial frame; (2) first iterate; and (3) second iterate. By the second iteration the divergence in vector magnitude and the angular collapse of the frame are quite apparent. Initial conditions were chosen so that the angular collapse of the vectors was uncommonly *slow*.

†We are aware of an attempt to estimate Lyapunov spectra from experimental data through direct estimation of local Jacobian matrices and formation of the long time product matrix [31]. This calculation is essentially the same as ours (we avoid matrix notation by diagonalizing the system at each step) and has the same problems of sensitivity to external noise, and to the amount and resolution of data required for accurate estimates.

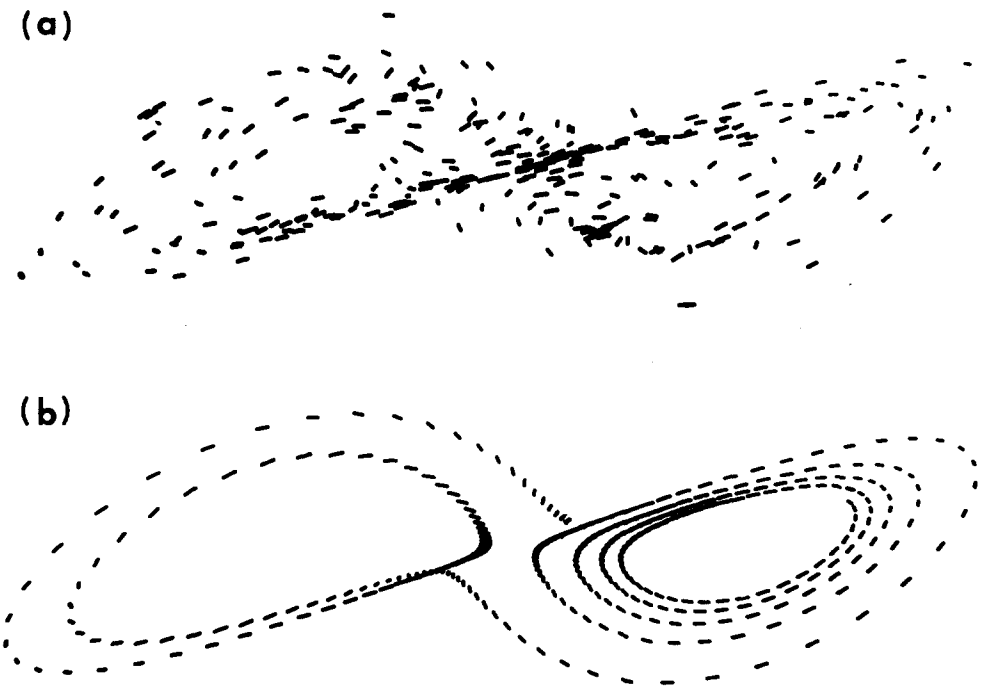

Fig. 3. A modification to the ODE spectral code (see appendix A) allows us to plot the running direction of greatest growth (vector v_1') in the Lorenz attractor. In (a), infrequent renormalizations confirm that this direction is not single-valued on the attractor. In (b), frequent renormalizations show us that this direction is usually nearly parallel to the flow. In the Rossler attractor, this direction is usually nearly orthogonal to the flow.

4. An approach to spectral estimation for experimental data

Experimental data typically consist of discrete measurements of a single observable. The well-known technique of phase space reconstruction with delay coordinates [2, 33, 34] makes it possible to obtain from such a time series an attractor whose Lyapunov spectrum is identical to that of the original attractor. We have designed a method, conceptually similar to the ODE approach, which can be used to estimate non-negative Lyapunov exponents from a reconstructed attractor. To understand our method it is useful to summarize what we have discussed thus far about exponent calculation.

Lyapunov exponents may be defined by the *phase space* evolution of a sphere of states. Attempts to apply this definition numerically to equations of motion fail since computer limitations do not allow the initial sphere to be constructed sufficiently small. In the ODE approach one avoids this problem by working in the *tangent space* of a fiducial trajectory so as to obtain always infinitesimal principal axis vectors. The remaining divergences are easily eliminated with Gram–Schmidt reorthonormalization.

The ODE approach is not directly applicable to experimental data as the linear system is not available. All is not lost provided that the linear approximation holds on the smallest length scales defined by our data. Our approach involves working in a reconstructed attractor, examining orbital divergence on length scales that are always as small as possible, using an approximate GSR procedure in the reconstructed *phase space* as

necessary. To simplify the ensuing discussion we will assume that the systems under consideration possess at least one positive exponent.

To estimate λ_1 we in effect monitor the long-term evolution of a single pair of nearby orbits. Our reconstructed attractor, though defined by a single trajectory, can provide points that may be considered to lie on different trajectories. We choose points whose temporal separation in the original time series is at least one mean orbital period, because a pair of points with a much smaller temporal separation is characterized by a zero Lyapunov exponent. Two data points may be considered to define the early state of the first principal axis so long as their spatial separation is small. When their separation becomes large we would like to perform GSR on the vector they define (simply normalization for this single vector), which would involve replacing the non-fiducial data point with a point closer to the fiducial point, in the same direction as the original vector. With finite amounts of data, we cannot hope to find a replacement point which falls exactly along a specified line segment in the reconstructed phase space, but we can look for a point that comes close. *In effect, through a simple replacement procedure that attempts to preserve orientation and minimize the size of replacement vectors, we have monitored the long-term behavior of a single principal axis vector.* Each replacement vector may be evolved until a problem arises, and so on. This leads us to an estimate of λ_1. (See fig. 4a.)

The use of a finite amount of experimental data does not allow us to probe the desired infinitesimal length scales of an attractor. These scales are also inaccessible due to the presence of noise on finite length scales and sometimes because the chaos-producing structure of the attractor is of negligible spatial extent. A discussion of these points is deferred until section 7.1.

An estimate of the sum of the two largest exponents $\lambda_1 + \lambda_2$ is similarly obtained. In the ODE procedure this involves the long-term evolution of a fiducial trajectory and a pair of tangent space vectors. In our procedure a triple of points is evolved in the reconstructed attractor. Before the area element defined by the triple becomes comparable to the extent of the attractor we mimic GSR by keeping the fiducial point, replacing the remainder of the triple with points that define a smaller area element and that best preserve the element's phase space orientation. Renormalizations are necessary solely because vectors grow too large, *not* because vectors will collapse to indistinguishable directions in phase space (this is unlikely with the limited amounts of data usually available in experiments). The exponential growth rate of area elements provides an estimate of $\lambda_1 + \lambda_2$. (See fig. 4b.)

Our approach can be extended to as many non-negative exponents as we care to estimate: $k+1$ points in the reconstructed attractor define a k-volume element whose long-term evolution is possible through a data replacement procedure that attempts to preserve phase space orientation and probe only the small scale structure of the attractor. The growth rate of a k-volume element provides an estimate of the sum of the first k Lyapunov exponents.

In principle we might attempt the estimation of negative exponents by going to higher-dimensional volume elements, but information about contracting phase space directions is often impossible to resolve. In a system where fractal structure can be resolved, there is the difficulty that the volume elements involving negative exponent directions collapse exponentially fast, and are therefore numerically unstable for experimental data (see section 7.1).

5. Spectral algorithm implementation

We have implemented several versions of our algorithms including simple "fixed evolution time" programs for λ_1 and $\lambda_1 + \lambda_2$, "variable evolution time" programs for $\lambda_1 + \lambda_2$, and "interactive" programs that are used on a graphics machine†.

†The interactive program avoids the profusion of input parameters required for our increasingly sophisticated expo-

In appendix B we include Fortran code and documentation for the λ_1 fixed evolution time program. This program is not sophisticated, but it is concise, easily understood, and useful for learning about our technique. We do not include the fixed evolution time code for $\lambda_1 + \lambda_2$ (though it is briefly discussed at the end of appendix B) or our other programs, but we will supply them to interested parties. We can also provide a highly efficient data base management algorithm that can be used in any of our programs to eliminate the expensive process of exhaustive search for nearest neighbors. We now discuss the fixed evolution time program for λ_1 and the variable evolution time program for $\lambda_1 + \lambda_2$ in some detail.

5.1. *Fixed evolution time program for* λ_1

Given the time series $x(t)$, an m-dimensional phase portrait is reconstructed with delay coordinates [2, 33, 34], i.e., a point on the attractor is given by $\{x(t), x(t+\tau), \ldots, x(t+[m-1]\tau)\}$ where τ is the almost arbitrarily chosen delay time. We locate the nearest neighbor (in the Euclidean sense) to the initial point $\{x(t_0), \ldots, x(t_0+[m-1]\tau)\}$ and denote the distance between these two points $L(t_0)$. At a later time t_1, the initial length will have evolved to length $L'(t_1)$. The length element is propagated through the attractor for a time short enough so that only small scale attractor structure is likely to be examined. If the evolution time is too large we may see L' shrink as the two trajectories which define it pass through a folding region of the attractor. This would lead to an underestimation of λ_1. We now look for a new data point that satisfies two criteria reasonably well: its separation, $L(t_1)$, from the evolved fiducial point is small, and the angular separation between the evolved and replacement elements is small (see fig. 4a). If an adequate replacement point cannot be found, we retain the points that were being used. This procedure is repeated until the fiducial trajectory has traversed the entire data file, at which point we estimate

$$\lambda_1 = \frac{1}{t_M - t_0} \sum_{k=1}^{M} \log_2 \frac{L'(t_k)}{L(t_{k-1})}, \tag{9}$$

where M is the total number of replacement steps. In the fixed evolution time program the time step $\Delta = t_{k+1} - t_k$ (EVOLV in the Fortran program) between replacements is held constant. In the limit of an infinite amount of noise-free data our procedure always provides replacement vectors of infinitesimal magnitude with no orientation error, and λ_1 is obtained by definition. In sections 6 and 7 we discuss the severity of errors of orientation and finite vector size for finite amounts of noisy experimental data.

5.2. *Variable evolution time program for* $\lambda_1 + \lambda_2$

The algorithm for estimating $\lambda_1 + \lambda_2$ is similar in spirit to the preceeding algorithm, but is more complicated in implementation. A trio of data points is chosen, consisting of the initial fiducial point and its two nearest neighbors. The area $A(t_0)$ defined by these points is monitored until a replacement step is both *desirable* and *possible* – the evolution time is variable. This mandates the use of several additional input parameters: a minimum number of evolution steps between replacements (JUMPMN), the number of steps to evolve backwards (HOPBAK) when a replacement site proves inadequate, and a maximum length or area before replacement is attempted.

nent programs. This program allows the operator to observe: the attractor, a length or area element evolving over a range of times, the best replacement points available over a range of times, and so forth. Each of these is seen in a two or three-dimensional projection (depending on the graphical output device) with terminal output providing supplementary information about vector magnitudes and angles in the dimension of the attractor reconstruction. Using this information the operator chooses appropriate evolution times and replacement points. The program is currently written for a Vector General 3405 but may easily be modified for use on other graphics machines. A 16mm movie summarizing our algorithm and showing the operation of the program on the Lorenz attractor has been made by one of the authors (A.W.).

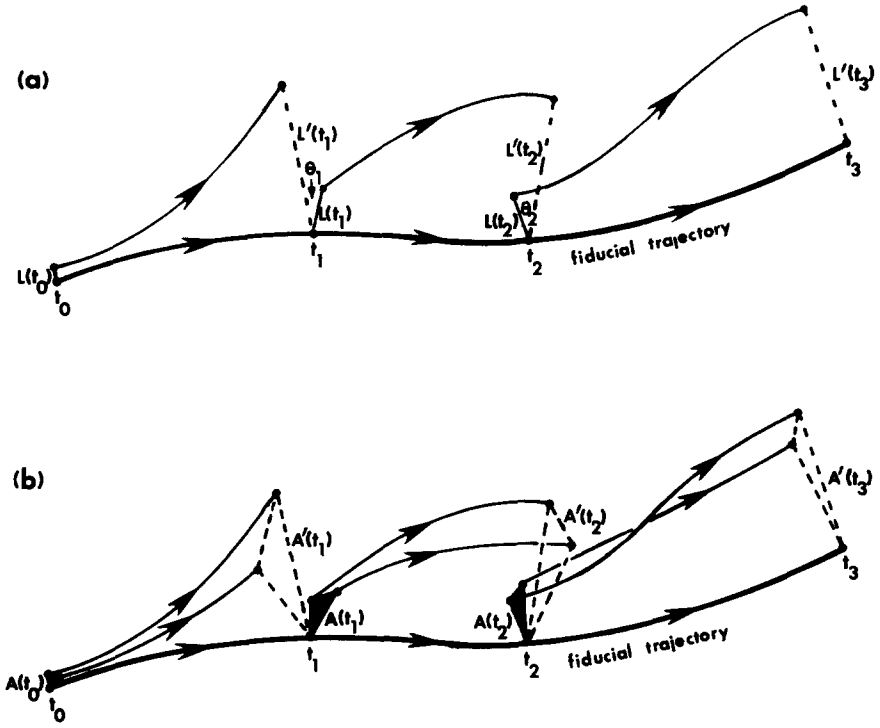

Fig. 4. A schematic representation of the evolution and replacement procedure used to estimate Lyapunov exponents from experimental data. a) The largest Lyapunov exponent is computed from the growth of length elements. When the length of the vector between two points becomes large, a new point is chosen near the reference trajectory, minimizing both the replacement length L and the orientation change ϑ. b) A similar procedure is followed to calculate the sum of the two largest Lyapunov exponents from the growth of area elements. When an area element becomes too large or too skewed, two new points are chosen near the reference trajectory, minimizing the replacement area A and the change in phase space orientation between the original and replacement area elements.

Evolution continues until a "problem" arises. In our implementation the problem list includes: a principal axis vector grows too large or too rapidly, the area grows too rapidly, and the skewness of the area element exceeds a threshold value. Whenever any of these criteria are met, the triple is evolved backwards HOPBAK steps and a replacement is attempted. If replacement fails, we will pull the triple back another HOPBAK steps, and try again. This process is repeated, if necessary, until the triple is getting uncomfortably close to the previous replacement site. At this point we take the best available replacement point, and jump forward at least JUMPMN steps to start the next evolution. At the first replacement time, t_1, the two points not on the fiducial trajectory are replaced with two new points to obtain a smaller area $A(t_1)$ whose orientation in phase space is most nearly the same as that of the evolved area $A'(t_1)$. Determining the set of replacement points that best preserves area orientation presents no fundamental difficulties.

Propagation and replacement steps are repeated (see fig. 4b) until the fiducial trajectory has traversed the entire data file at which point we estimate

$$\lambda_1 + \lambda_2 = \frac{1}{t_M - t_0} \sum_{k=1}^{M} \log_2 \frac{A'(t_k)}{A(t_{k-1})}, \tag{10}$$

where t_k is the time of the kth replacement step.

It is often possible to verify our results for λ_1 through the use of the $\lambda_1 + \lambda_2$ calculation. For

attractors that are very nearly two dimensional there is no need to worry about preserving orientation when we replace triples of points. These elements may rotate and deform within the plane of the attractor, but replacement triples always lie within this same plane. Since λ_2 for these attractors is zero, area evolution provides a direct estimate for λ_1. With experimental data that appear to define an approximately two-dimensional attractor, an independent calculation of d_f from its definition (feasible for attractors of dimension less than three [35]) may justify this approach to estimating λ_1.

6. Implementation details

6.1. *Selection of embedding dimension and delay time*

In principle, when using delay coordinates to reconstruct an attractor, an embedding [34] of the original attractor is obtained for any sufficiently large m and almost any choice of time delay τ, but in practice accurate exponent estimation requires some care in choosing these two parameters. We should obtain an embedding if m is chosen to be greater than twice the dimension of the underlying attractor [34]. However, we find that attractors reconstructed using smaller values of m often yield reliable Lyapunov exponents. For example, in reconstructing the Lorenz attractor from its x-coordinate time series an embedding dimension of 3 is adequate for accurate exponent estimation, well below the sufficient dimension of 7 given by ref. [34]†. When attractor reconstruction is performed in a space whose dimension is too low, "catastrophes" that interleave distinct parts of the attractor are likely to result. For example, points on separate lobes of the Lorenz attractor may be coincident in a *two*-dimensional reconstruction of the attractor. When this occurs, replacement elements may contain points whose separation in the original attractor is very large; such elements are liable to grow at a dramatic rate in our reconstructed attractor in the short term, providing an enormous contribution to the estimated exponent. As these elements tend to blow up almost immediately, they are also quite troublesome to replace‡.

If m is chosen too large we can expect, among other problems, that noise in the data will tend to decrease the density of points defining the attractor, making it harder to find replacement points. Noise is an infinite dimensional process that, unlike the deterministic component of the data, fills each available phase space dimension in a reconstructed attractor (see section 7.2). Increasing m past what is minimally required has the effect of unnecessarily increasing the level of contamination of the data.

Another problem is seen in a three-dimensional reconstruction of the Hénon attractor. The reconstructed attractor looks much like the original attractor sitting on a two-dimensional sheet, with this sheet showing a simple twist in three-space. We expect that this behavior is typical; when m is increased, surface curvature increases*. Increasing m therefore makes it increasingly difficult to satisfy orientation constraints at replacement time, as the attractor is not sufficiently flat on the smallest length scales filled out by the fixed quantity of data. It is advisable to check the stationarity of

†We have found that it is often possible to ignore several components of evolving vectors in computing their average exponential rate of growth: keeping two or more components of the vector often suffices for this purpose. As our discussion of "catastrophes" will soon make clear, the search for replacement points most often requires that all of the delay coordinates be used.

‡If two points lie at opposite ends of an attractor, it is possible that their separation vector lies entirely outside of the attractor so that no orientation preserving replacement can be found. If this goes undetected, the current pair of points is likely to be retained for an orbital period or longer, until these points are accidentally thrown close together.

*A simple study for the Hénon system showed that for reconstructions of increasing dimension the mean distance between the points defining the attractor rapidly converged to an attractor independent value. The fold put in each new phase space direction by the reconstruction process tended to make the concept of "nearby point in phase space" meaningless for this finite data set.

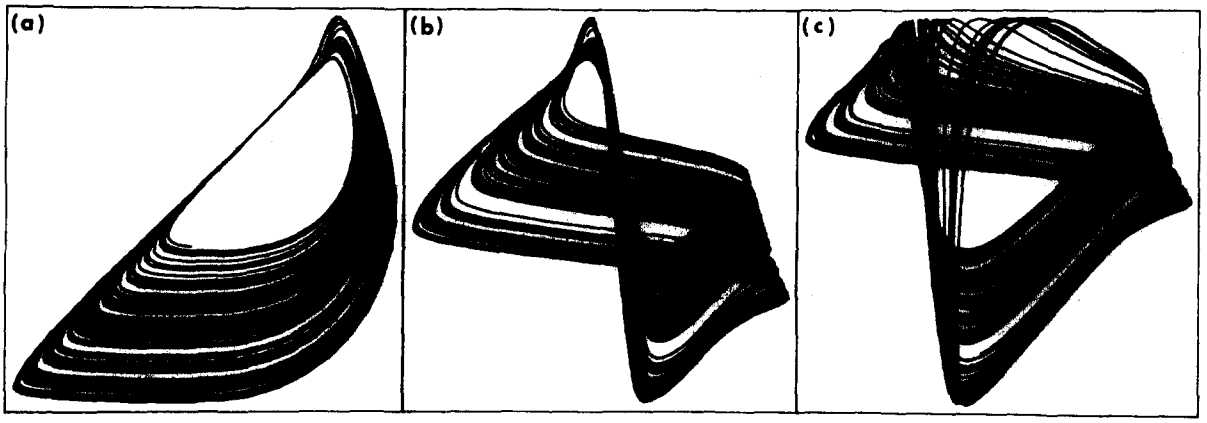

Fig. 5. The strange attractor in the Belousov–Zhabotinskii reaction is reconstructed by the use of delay coordinates from the bromide ion concentration time series [2]. The delays shown are a) $\frac{1}{12}$; b) $\frac{1}{2}$; and c) $\frac{3}{4}$ of a mean orbital period. Notice how the folding region of the attractor evolves from a featureless "pencil" to a large scale twist.

results with m to ensure robust exponent estimates.

Choice of delay time is also governed by the necessity of avoiding catastrophes. In our data [2] for the Belousov–Zhabotinskii chemical reaction (see fig. 5) we see a dramatic difference in the reconstructed attractors for the choices $\tau = 1/12$, $\tau = 1/2$ and $\tau = 3/4$ of the mean orbital period. In the first case we obtain a "pencil-like" region which obscures the folding region of the attractor. This structure opens up *and grows larger* relative to the total extent of the attractor for the larger values of τ, which is clearly desirable for our algorithms. We choose τ neither so small that the attractor stretches out along the line $x = y = z = \ldots$, nor so large that $m\tau$ is much larger than the orbital period. A check of the stationarity of exponent estimates with τ is again recommended.

6.2. *Evolution times between replacements*

Decisions about propagation times and replacement steps in these calculations depend on additional input parameters, or in the case of the interactive program, on the operator's judgement. (The stationarity of λ_1 values over ranges of all algorithm parameters is illustrated for the Rossler attractor in figs. 6a–6d.) Accurate exponent calculation therefore requires the consideration of the following interrelated points: the desirability of maximizing evolution times, the tradeoff between minimizing replacement vector size and minimizing the concomitant orientation error, and the manner in which orientation errors can be expected to accumulate. We now discuss these points in turn.

Maximizing the propagation time of volume elements is highly desirable as it both reduces the frequency with which orientation errors are made and reduces the cost of the calculation considerably (element propagation involves much less computation than element replacement). In our variable evolution time program this is not much of a problem, as replacements are performed only when deemed necessary (though the program has been made conservative in such judgments). In the interactive algorithm this is even less of a problem, as an experienced operator can often process a large file with a very small number of replacements. The problem is severe, however, in our fixed evolution time program, which is otherwise desirable for its extreme simplicity. In this program replacements are attempted at fixed time steps, independent of the behavior of the volume element.

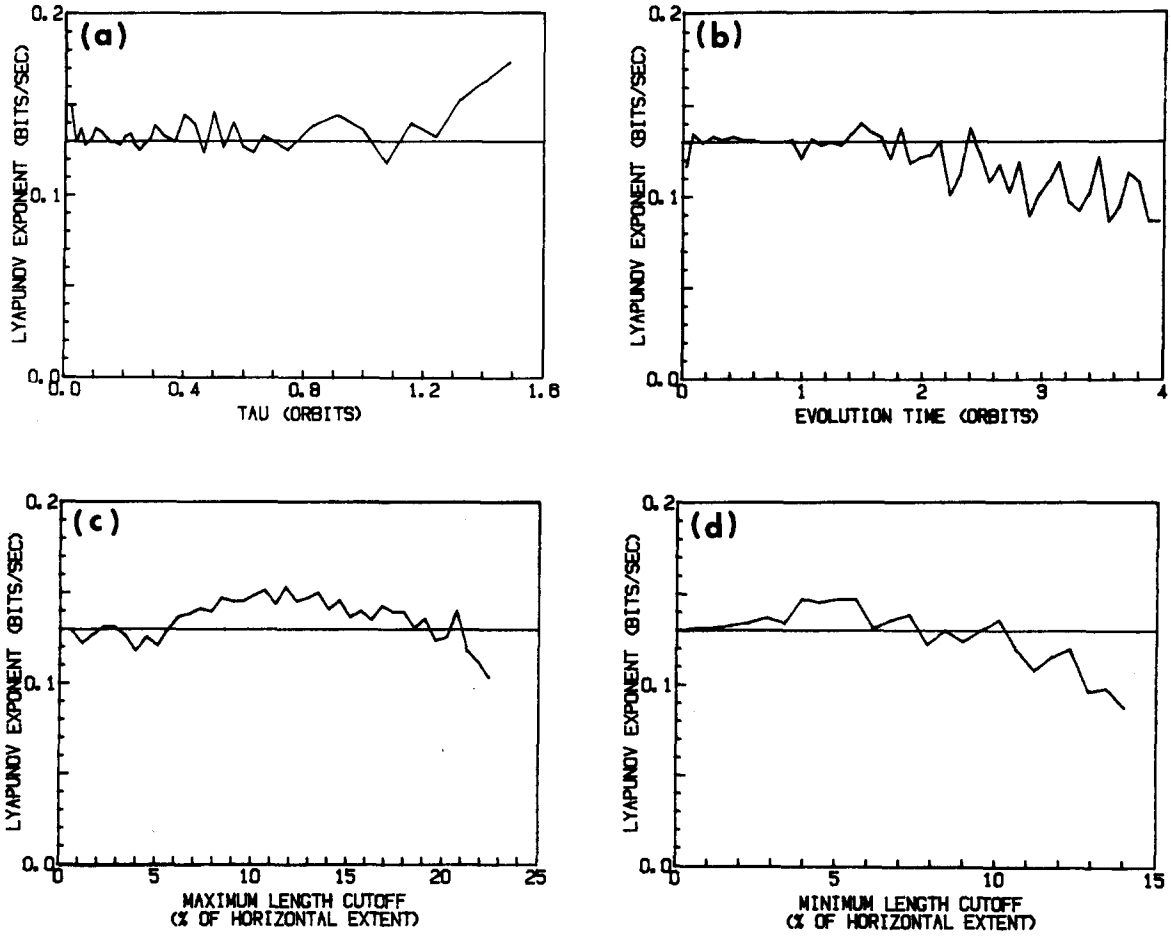

Fig. 6. Stationarity of λ_1 for Rossler attractor data (8192 points spanning 135 orbits) for the fixed evolution time program is shown for the input parameters: a) Tau (delay time); b) evolution time between replacement steps; c) maximum length of replacement vector length allowed; and d) minimum length of replacement vector allowed. The correct value of the positive exponent is 0.13 bits/s and is shown by the horizontal line in these figures.

Our numerical results on noise-free model systems have produced the expected results: too frequent replacements cause a dramatic loss of phase space orientation, and too infrequent replacements allow volume elements to grow overly large and exhibit folding. For the Rossler, Lorenz, and the Belousov–Zhabotinskii attractors, each of which has a once-per-orbit chaos generating mechanism, we find that varying the evolution time in the range $\frac{1}{2}$ to $1\frac{1}{2}$ orbits almost always provides stable exponent estimates. In systems where the mechanism for chaos is unknown, one must check for exponent stability over a wide range of evolution times. For such systems it is perhaps wise to employ only the variable evolution time program or the interactive program.

There are other criteria that may affect replacement times for variable evolution time programs such as avoiding regions of high phase space velocity, where the density of replacement points is likely to be small. Such features are easily integrated into our programs.

In the Lorenz attractor, the separatrix between the two lobes of the attractor is not a good place to find a replacement element. An element chosen here is likely to contain points that will almost

immediately fly to opposite lobes, providing an enormous contribution to an exponent estimate. This effect is certainly related to the chaotic nature of the attractor, but is not directly related to the values of the Lyapunov exponents. This has the same effect as the catastrophes that can arise from too low a value of embedding dimension as discussed in section 6.1. While we are not aware of any foolproof approach to detecting troublesome regions of attractors it may be possible for an exponent program to avoid catastrophic replacements. For example, we may monitor the *future* behavior of potential replacement points and reject those whose separation from the fiducial trajectory is atypical of their neighbors.

6.3. *Shorter lengths versus orientation errors*

With a given set of potential replacement points some compromise will be necessary between the goals of minimizing the length of replacement vectors and minimizing changes in phase space orientation. On the one hand, short vectors may in general be propagated further in time, resulting in less frequent orientation errors. On the other hand, we may wish to minimize orientation errors directly. We must also consider that short vectors are likely to contain relatively large amounts of noise.

In the fixed evolution time program the search for replacements involves looking at successively larger length scales for a minimal orientation change. In the variable evolution time program, points satisfying minimum length and orientation standards are assigned scores based on a linear weighting (with heuristically chosen weighting factors) of their lengths and orientation changes. We have also performed numerical studies by searching successively larger angular displacements while attempting to satisfy a minimum length criterion. Fortunately, we find that these different approaches perform about equally well. Attempts to solve the tradeoff problem analytically have suggested "optimal" choices of initial vector magnitude, but due to the system dependent nature of these calculations, we cannot be confident that our results are of general validity.

The problem of considering the magnitude of evolved or replacement vectors is complicated by the fact that at a given point in an attractor, the orientation of a vector can determine whether or not it is too large. If we consider a system with an underlying 1-D map such as the Rossler attractor, it is the magnitude of the vector's component transverse to the attractor that is relevant. In this case our algorithm is closely related to obtaining the Lyapunov exponent of the map through a determination of its local slope profile [13]. The transverse vector component plays the role of the chord whose image under the map provides a slope estimate. This chord should obviously be no longer than the smallest resolvable structure in the 1-D map, a highly system-dependent quantity. Since the underlying maps of commonly studied model and physical systems have not had much detailed structure on small length scales (consider the logistic equation, cusp maps, and the Belousov–Zhabotinskii map [2]) we have somewhat arbitrarily decided to consider 5–10% of the transverse attractor extent as the maximum acceptable magnitude of a vector's transverse component.

6.4. *The accumulation of orientation errors*

The problem of the accumulation of orientation errors is reasonably well understood. Consider for simplicity a very nearly two-dimensional system with a $(+,0,-)$ spectrum, such as the Lorenz attractor. Post-transient data traverse the subspace characterized by the positive and zero exponents. Length propagation with replacement on the attractor is clearly susceptible to orientation error that will mix contributions from the positive and zero exponents in some complex, system dependent manner. Now consider the nth replacement step (see fig. 4a) with an orientation change within the plane of the attractor of ϑ_n. Further, let the angle the replacement vector makes with respect to the vector v_1' be $\bar{\vartheta}_n$. We make the crucial assumption that vectors are propagated for a time t that

is long enough that growth along directions v_1' and v_2' are reasonably well characterized by the exponents λ_1 and λ_2 respectively. Then for the new replacement vector

$$L(t_n) = L\left(v_1' \cos \bar{\vartheta}_n + v_2' \sin \bar{\vartheta}_n\right) \tag{11}$$

and at the next replacement

$$L'(t_{n+1}) = L\left(v_1'(\cos \bar{\vartheta}_n)2^{\lambda_1 t_r} + v_2'(\sin \bar{\vartheta}_n)2^{\lambda_2 t_r}\right), \tag{12}$$

where t_r is the time between successive replacement steps $(t_{n+1} - t_n)$. The contribution to eq. (9) from this evolution is then

$$\tfrac{1}{2} \log_2 \left[\cos^2 \bar{\vartheta}_n 2^{2\lambda_1 t_r} + \sin^2 \bar{\vartheta}_n 2^{2\lambda_2 t_r}\right] \tag{13}$$

and the angle the next replacement vector $L(t_{n+1})$ makes with v_1' is

$$\bar{\vartheta}_{n+1} = \arctan\left(b \cdot \tan \bar{\vartheta}_n\right) + \vartheta_{n+1}, \tag{14}$$

where

$$b = 2^{(\lambda_2 - \lambda_1)t_r}. \tag{15}$$

If we assume all angles are small compared to unity and set $\bar{\vartheta}_0 = \vartheta_0$, eq. (14) implies that

$$\bar{\vartheta}_n = \sum_{m=0}^{n} \vartheta_{n-m} b^m. \tag{16}$$

If the orientation changes have zero mean and are uncorrelated from replacement to replacement then an average over the changes gives

$$\left\langle \bar{\vartheta}_n^2 \right\rangle = \frac{\vartheta_M^2 (1 - b^{2n+2})}{1 - b^2}, \tag{17}$$

where ϑ_M is an angular change on replacement on the order of the ANGLMX parameter in the fixed evolution time program of appendix B. From eqs. (9), (13), and (17) we find the fractional error in λ_1 to be

$$\frac{\Delta\lambda_1}{\lambda_1} = \frac{-\vartheta_M^2}{2(\ln 2) N_t \lambda_1 t_r}\left[N_t - \frac{b^2(1 - b^{2N_t})}{1 - b^2}\right], \tag{18}$$

where N_t is the total number of replacement steps. If there is no degeneracy, i.e., $b^2 \ll 1$, eqs. (17) and (18) show that orientation errors do not accumulate, i.e, there is no N_t dependence, and our fractional error in λ_1 is

$$\frac{\Delta\lambda_1}{\lambda_1} = \frac{-\vartheta_M^2}{2(\ln 2)\lambda_1 t_r}. \tag{19}$$

For the Lorenz attractor, b^2 has a value of about 0.33 for an evolution time of one orbit, so an orientation error of about 19 degrees results in a 10% error in λ_1. If we can manage to evolve the vector for two orbits, the permissible initial orientation error is about 27 degrees, and so on. We see that a given orientation error at replacement time shrinks to a value negligible compared to the next orientation error, provided that propagation times are long enough. Orientation errors do not accumulate because there is no memory of previous errors.

This calculation may be generalized to an attractor with an arbitrary Lyapunov spectrum and a similar result is obtained. The ease of estimating the ith exponent depends on how small the quantity $2^{(\lambda_{i+1} - \lambda_i)t_r}$ is. Problems arise when successive exponents are very close or identical. Hyperchaos, with a spectrum of [0.16, 0.03, 0.00, $\simeq -40$] bits/s and an orbital period of about 5.16 s, has an easily determinable first exponent, but distinguishing λ_2 from λ_3 is more difficult.

7. Data requirements and noise

7.1. *Probing small length scales*

As we have already pointed out, the infinitesimal length scales on which the definition of Lyapunov exponents rely are inaccessible in ex-

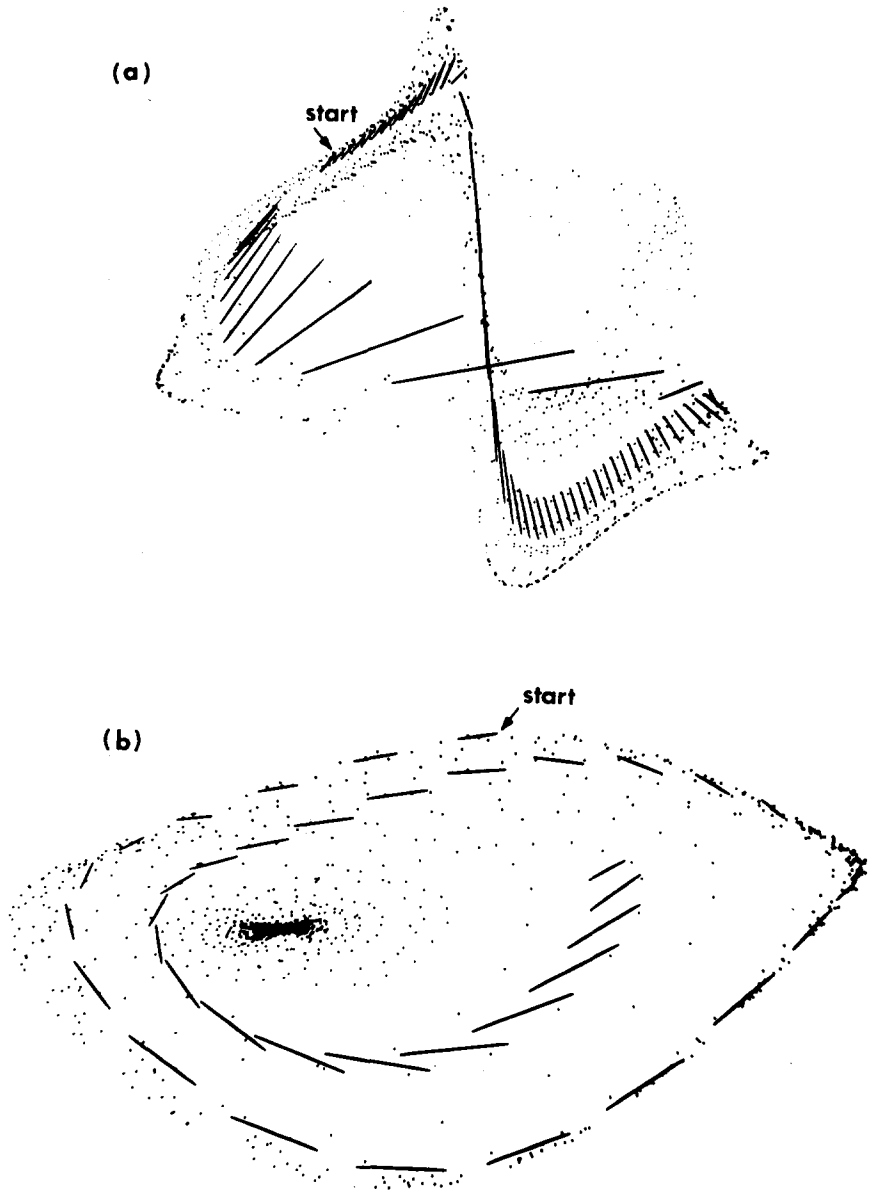

Fig. 7. Experimental data for two different Belousov–Zhabotinskii systems shows chaos on large and small length scales respectively. In the Texas attractor [2] a), the separation between a single pair of points is shown for one orbital period. In the French attractor [36]; b), the separation between a pair of points is shown for two periods. Estimation of Lyapunov exponents is quite difficult for the latter system.

perimental data. There are three somewhat related reasons why this is so: (1) a finite amount of attractor data can only define finite length scales; (2) the stretching and folding that is the chaotic element of a flow may occur on a scale small compared to the extent of the attractor; and (3) noise defines a length scale below which separations are meaningless. We discuss each of these problems in turn and then consider whether exponent estimation is possible in spite of them.

The finiteness of a data set means that the fixed evolution time program undoubtedly allows principal axis vectors to grow far too large on occasion and also to completely lose the proper phase space orientation, yet we almost invariably obtain accurate exponent estimates for noise-free model systems defined by small data sets. This is because *on the time scale of several orbital periods*, orbital divergences may be moderately well characterized by Lyapunov exponents in sufficiently chaotic systems. Averaging many such segments together in our algorithms is therefore likely to mask *infrequent* large errors.

The problem of "chaos on a small length scale" is a system dependent one. Consider a system such as the Rossler attractor in which chaos generation occurs on a relatively large length scale. Here it is quite easy to distinguish between true exponential divergence of nearby orbits and a temporary divergence due to local changes in the attractor's shape. If, however, the Rossler attractor were "crossed" with a periodic motion of sufficiently large amplitude, we would lose the ability to detect the mechanism for chaos as it would manifest itself only on length scales that we must regard as suspiciously small. For such a system it is difficult to conceive of any means of recovering exponents from experimental data.

We have observed this problem to some degree for the Couette–Taylor system, which makes a transition from motion on a 2-torus to chaos. In such a system chaos can arise from small stretches and folds on the torus. When we use the interactive program to monitor the evolution of lengths in the Couette–Taylor attractors, we seem to observe this effect; that is, the separation vectors do not exhibit dramatic growth but instead oscillate in magnitude. Such an oscillation could indicate a stretching and folding so that we might wish to attempt a replacement, or it could simply indicate a variation of the attractor's shape, which should be ignored. In figs. 7a and 7b we show experimental data for attractors of the large scale [2] and small scale [36] varieties, both arising from the Belousov–Zhabotinskii system. Exponent estimation in the latter case is quite difficult. The presence of external noise on length scales as small as the chaos generation mechanism will of course further complicate exponent calculations.

Even though infinitesimal length scales are not accessible, Lyapunov exponent estimation may still be quite feasible for many experimental systems. The same problem arises in calculations of the fractal dimension of strange attractors. Fractal structure does not exist in nature, where it is truncated on atomic scales, nor does it exist in any computer representation of a dynamical system, where finite precision truncates scaling. In these calculations we hope that as the smallest *accessible* length scales are approached, scaling converges to the zero length scale limit. Similarly, provided that chaos production is nearly the same on infinitesimal and the smallest accessible length scales, our calculations on the small scales may provide accurate results. A successful calculation requires that one has enough data to approach the appropriate length scales, ignores anything on the length scale of the noise, and has an attractor with a macroscopic stretching/folding mechanism.

7.2. *Noise*

The effects of noise in our algorithms fall into two categories which we have named the "statistical" and the "catastrophic". The former category deals with such problems as point-to-point jitter that cause us to estimate volumes inaccurately; this was the motivation for avoiding highly skewed replacement elements. Catastrophes can arise even in the absence of noise either from too low an embedding dimension (section 6.1), or from too little data compounded with high attractor complexity (section 6.2). In the presence of noise, catastrophes occur because noise drives a faraway data point into the replacement "arena." Noise in physical systems can be broken into two categories: measurement noise, i.e., simple lack of resolution, and dynamical noise, i.e., fluctuations in the state of the system or its parameters which enter into the dynamics. In the former case it is

clear that the system possesses well defined exponents that are potentially recoverable. Strictly speaking, in the latter case Lyapunov exponents are not well defined, but some work [37] has suggested that a system may be characterized by numbers that are the Lyapunov exponents for the noise-free system averaged over the range of noise-induced states.

Our first study of the effects of noise on our algorithms involved adding dynamical noise to the Hénon attractor, that is, a small uniformly distributed random number was added to each coordinate as the map was being iterated. These data were then processed with the fixed evolution time program. For noise of sufficiently large amplitude, λ_1 could not be accurately determined. Specifically, the average initial separation between replacement points grew with the noise level (noise causing diffusion of the 1.26-dimensional attractor into the two-dimensional phase space) and the large final separations were not much affected by the noise. The result was an underestimate of the positive exponent. A nearly identical result was obtained for the addition of measurement noise (addition of a random term to each element of the time series, after the entire series has been generated) to the Hénon attractor.

It is ironic that measurement noise is not a problem unless large amounts of data are available to define the attractor. Noise is only detectable when the point density is high enough to provide replacements near the noise length scale. This suggests a simple approach to the noise problem, simply avoiding principal axis vectors whose magnitude is smaller than some threshold value we select. If this value is chosen to be somewhat larger than the noise level, the fractional error in determining initial vector magnitudes may be reduced to an acceptable level. Avoiding noisy length scales is not a trivial matter, as noise may not be of constant amplitude throughout an attractor and the noise length scale may be difficult to determine. Again, this approach can only work if scales larger than the noise contain accurate information about orbital divergence rates in the zero length scale limit. In fig. 6d we confirm that a straightforward small distance cutoff works in the case of the Rossler attractor by showing stationarity of the estimated exponent over a broad range of cutoff values.

7.3. *Low pass filtering*

Another approach to reducing the effects of noise, closely related to the use of a small distance cutoff, involves low pass filtering of the data before beginning exponent estimation. Rather severe filtering may be possible for systems with a once-per-orbit chaos producing mechanism – the filter cutoff approaching (orbital period)$^{-1}$. Filtering can be expected to distort shapes, eliminate small scale structure, and scramble phase, but we do not expect the divergent nature of the attractor to be lost.

A demonstration of the use of filtering for the Belousov–Zhabotinskii attractor is shown in fig. 8. Filtering dramatically altered the shape of the reconstructed attractor, but the estimated values of λ_1 differed by at most a few percent for reasonable cutoff frequencies. A similar calculation for the Rossler attractor indicated that the low-frequency cutoff could be moved all the way down to the attractor's sole large spectral feature before the exponent estimate showed any noticeable effect. Results for the Lorenz attractor, with its much more complicated spectral profile, are not quite as impressive. An analytical proof of the low pass filtering invariance of Lyapunov exponents (with conditions on the cutoff frequency relative to the orbital period and the replacement frequency) has proved elusive. Of course, low pass filtering fails to help with exponent estimation if there is substantial contamination of the data at frequencies lower than the filter cutoff. In a simple study of multiperiodic data with added white noise [3] the estimated exponent returned (very nearly) to zero for a sufficient amount of filtering. It thus appears that in some cases external noise can be distinguished from chaos by this procedure.

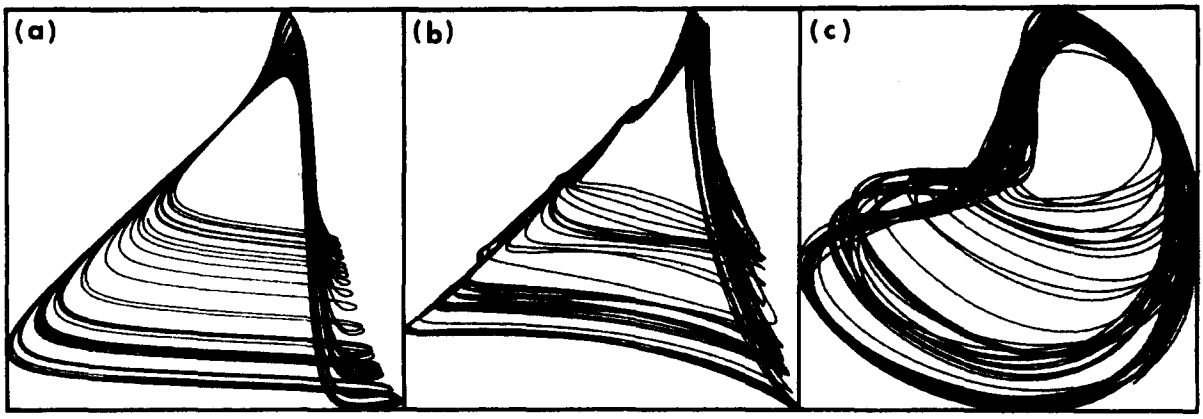

Fig. 8. a) Unfiltered experimental data for the Belousov–Zhabotinskii reaction [2]; b) the same data, low-pass filtered with a filter cutoff of 0.046 Hz, compared to the mean orbital frequency of 0.009 Hz. Our estimate of λ_1 for these data was only 5% lower than the estimate from the unfiltered data. Replacement frequencies in the region of stationarity for these results were approximately 0.005 Hz. c) the data are over-filtered at 0.023 Hz. λ_1 differs by only 20% from the exponent estimate for unfiltered data.

7.4. *Data requirements*

We now address the important questions of the quality and quantity of experimental data required for accurate exponent calculation. The former question is more easily disposed of – resolution requirements are so highly system dependent that we cannot make any general statements about them. In one study with the fixed evolution time program, the largest exponent was repeatedly computed for Rossler and Lorenz attractor data, the resolution of which was decreased one bit at a time from 16 bits. In each case the estimates were reasonably good for data with as few as 5 bits resolution. In fig. 9 we show the results of bit chopping for these systems as well as for Belousov–Zhabotinskii data. As a conservative rule of thumb we suggest a minimum of 8 meaningful bits of precision be used for exponent calculations. We strongly suggest that the resolution of experimental data be artificially lowered as we did for the model systems. If a plot of λ_1 versus resolution does not show an initial plateau for at least one or two bits, the initial data are suspect for the purpose of exponent calculations.

The amount of data required to calculate Lyapunov exponents depends on three distinct factors: the number of points necessary to provide an adequate number of replacement points, the number of orbits of data necessary to probe stretching (but not folding) within the attractor, and the number of data points per orbit that allow for proper attractor reconstruction with delay coordinates.

We first estimate how many points are required to "fill out" the structure of an attractor to provide replacement points. A simple minded estimate of this factor depends on the following factors: the fractal dimension of the attractor, the number of non-negative exponents in the system, the number of exponents we are attempting to compute (the dimension of each volume element), and the geometric requirements for acceptable replacement points. A more accurate calculation of this number will depend on such detailed information as the attractor's fractal structure and its probability density, which are not typically available for experimental data and the effective noise level in the system (which depends on both the actual level of contamination and the dimension of the reconstructed attractor). We assume that our data are uniformly distributed over a d-dimensional attractor of extent L and ignore noise-induced diffusion of the data. Thus, the density of

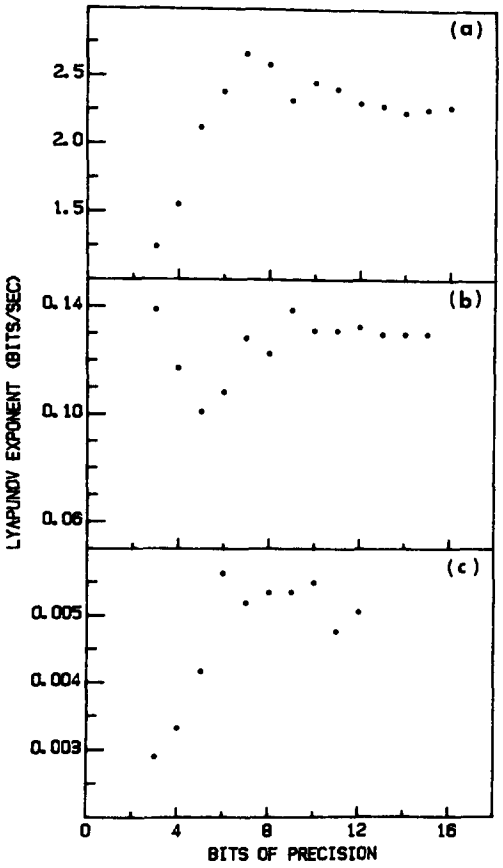

Fig. 9. The results of bit chopping (simulated measurement noise) for the a) Lorenz; b) Rossler; and c) Belousov–Zhabotinskii systems. For each system at least 5 bits of precision in the data are required for accurate exponent estimates.

points, ρ, is

$$\rho = \frac{N}{L^d}. \tag{20}$$

The mean number of replacement points located in a region of length ε (SCALMX) and angular size ϑ (ANGLMX) is given by

$$N_r = V_d(\varepsilon, \vartheta)\rho \tag{21}$$

where V_d, the volume of a d-dimensional search cone, is proportional to $\varepsilon^d \vartheta^{d-1}$, with d the (nearest integer) dimension of the attractor. N_r may be set to 1 for λ_1 estimation. Combining these expressions and solving for N, we obtain

$$N(\lambda_1) \propto \frac{1}{\vartheta^{d-1}} \left[\frac{L}{\varepsilon}\right]^d. \tag{22}$$

A nearly identical calculation for the number of points required for area replacement results in

$$N(\lambda_1 + \lambda_2) \propto \frac{1}{\vartheta^{d-2}} \left[\frac{L}{\varepsilon}\right]^d, \tag{23}$$

and in general,

$$N\left(\sum_{i=1}^{j} \lambda_i\right) \propto \frac{1}{\vartheta^{d-j}} \left[\frac{L}{\varepsilon}\right]^d. \tag{24}$$

We have ignored several prefactors that might modify these estimates by at most an order of magnitude. Also, the variance of the density of points is so high that the data requirement should probably be substantially increased to ensure that replacements are almost always available when needed, not just on the average. Perhaps surprisingly, the number of points required for estimating $\lambda_1 + \lambda_2$ is not significantly larger than $N(\lambda_1)$ (our estimate actually predicts it to be smaller). While we must double the number of points in the attractor to have a good chance of finding a pair of replacement points rather than a single one, the search volume for area replacement is actually larger (a larger solid angle of a potential replacement sphere is acceptable) than the search volume for length replacement. For area evolution there are pairs of points that define highly skewed replacement elements, but these are sufficiently unlikely that we can ignore their effect on $N(\lambda_1 + \lambda_2)$. For calculation of exponents past λ_2, the required point density does not change significantly. In our numerical work, in a best case scenario $L/\varepsilon \approx 5$, and the maximum value of ϑ is about 0.6 radians. In a worst case calculation L/ε is about 10, and ϑ is about 0.3 radians. From these values eq. (24) predicts to first order that between $\approx 10^d$ and $\approx 30^d$ points are necessary to fill out a d-dimensional attractor, independent of how many non-negative exponents we are calculating.

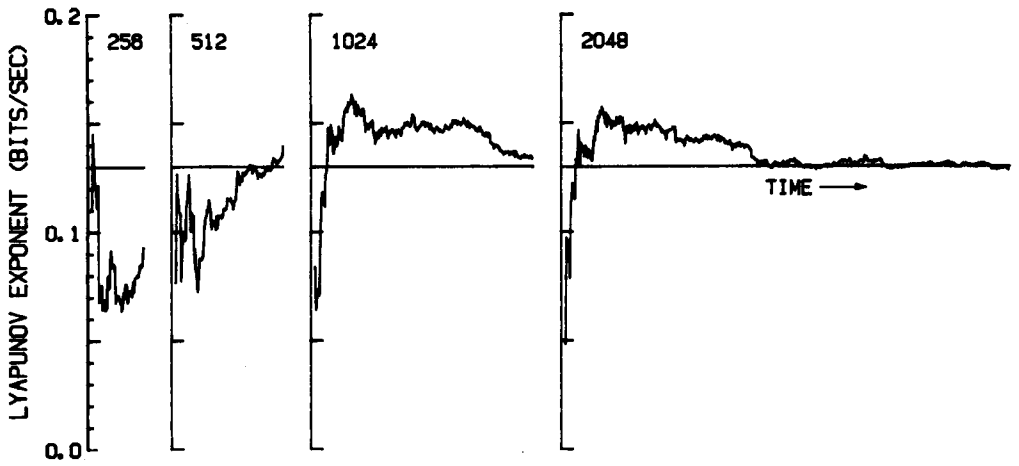

Fig. 10. The temporal convergence of λ_1 is shown as a function of the number of data points defining the Rossler attractor. The sampling rate is held constant; the longer time series contain more orbits. Note that lengthening the time series not only allows more time for convergence but also provides more replacement candidates at each replacement step. (Here the embedding dimension was 3, the embedding delay (τ) was a sixth of an orbit, and the evolution time-step was three-quarters of an orbit.)

The next factor we consider is the number of orbits of data required. The analysis is simplest if chaos arises through a once per orbit stretching and folding mechanism, which may be represented by a discrete mapping in one or more dimensions (as, for example, in the Lorenz, Rossler, hyperchaos, and Belousov–Zhabotinskii attractors). Exponent convergence requires that volume elements be operated on many times by the mapping until the element has sampled the slope profile of the map, suitably weighted by the map's probability density. The Lorenz and Rossler attractors have simple underlying 1-D maps; on the order of 10 to 100 map points are required for adequate slope profiles [13]. For these attractors, we expect between 10 and 100 orbits worth of data will be required for estimating λ_1 or for confirming that $\lambda_2 = 0$. No additional orbits are required for area propagation in a system defined by a 1-D map. If the system had an underlying 2-D map as hyperchaos does, we might expect, depending on the complexity of the map, that roughly the square of this number of orbits would be required. This would provide the same sampling resolution for the slope profile of the map (see ref. 13) in each dimension. In general, for a system defined by an n-D map, the number of orbits of data required to estimate any non-negative exponent is given by a constant, C, raised to a power which is the number of positive exponents. The number of positive exponents is approximately the dimension of the attractor minus one, thus the required number of orbits is about C^{d-1}. C is a system dependent quantity depending on the amount of structure in the map, perhaps in the range 10 to 100.

The last and simplest point we consider is the required number of points per orbit, P. There is no benefit to choosing P any larger than is absolutely necessary. We might try to choose P so small that in an evolution of a single step, the average replacement vector would grow to as large a separation as we care to allow. In the Lorenz attractor, for example, we might decide to allow the average replacement vector to grow by a factor of 32 in a single time step, so that we would have one data point per 6 orbits. The problem is that with data this sparse we are unlikely to obtain a good reconstruction of our attractor. Often, the relationship $m\tau \approx 1$ is used in reconstructions, where m is the embedding dimension and τ is the delay in units of orbital periods. We assume that reconstruction is performed in an approximately

d-dimensional space, and the delay corresponds to a single sample time, so that $\tau = 1/P$. We then obtain a requirement of about d points per orbit†.

When the number of points per orbit is multiplied by the number of orbits, we obtain a required number of points ranging from $d \times 10^{d-1}$ to $d \times 100^{d-1}$, which we can compare to the point density requirement of between 10^d and 30^d points. Since all three requirements must be met, the larger of these two quantities determines the amount of data required. In each of these two ranges of values the complexity of the underlying map (if any) determines which end of the range is appropriate. Therefore we may conclude that for up to about a 10-dimensional system the required number of data points ranges from 10^d to 30^d. We compute this range for several systems: Hénon attractor ($d = 1.26$), 30–100 points; Rossler attractor ($d = 2.01$), 100–1000 points; hyperchaos ($d = 3.005$), 1000–30000 points; delay differential equation ($d = 3.64$), 4000–200000 points. We see that the amount of data required to estimate non-negative exponents rises exponentially fast with the dimension of the attractor, the identical problem with calculations of fractal dimension by all algorithms of the distance scaling variety [35]. Fig. 10 shows the convergence of our λ_1 estimate as the number of points used is increased for the Rossler attractor. It is important to note that while it may take 32000 points to define an attractor, it is generally not necessary to evolve completely through the data before the exponent estimate converges. For example, see fig. 10.

8. Results

We now present our results for the various model and experimental systems on which our algorithms have been tested. We emphasize that no explicit use was made of the differential equations defining the model systems, except to produce a dynamical observable (the x-coordinate time series) which was then treated as experimental data. For the equations that define each system see table I. The quoted uncertainty values for each system were calculated either from the known values of the exponents or from the variation of our results with changes in parameters.

Hénon attractor

For the Hénon map, we obtained the positive exponent to within 5% with only 128 points defining the attractor.

Rossler attractor

For the Rossler attractor, we found the first exponent with a 5% error using 1024 points. The second exponent was measured as less than 6% of the first with 2048 points defining the attractor. Six points per mean orbital period were used to define the attractor.

Lorenz attractor

The Lorenz system was the most difficult test of the fixed evolution time program because its ill-defined orbital period made it difficult to avoid catastrophic replacements near the separatrix. In using the interactive program the operators simply

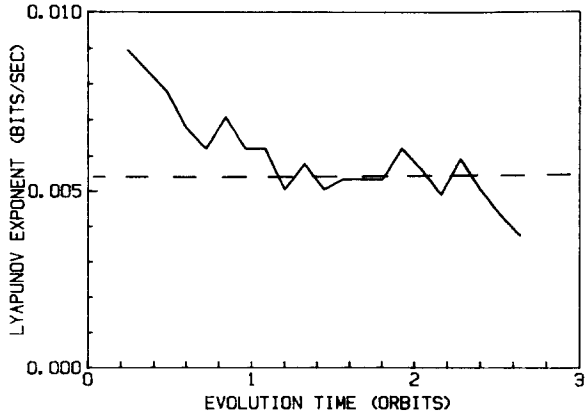

Fig. 11. Stationarity of λ_1 with evolution time is shown for Belousov–Zhabotinskii data [2] (compare to fig. 6b).

†We note that d points per orbit is a very small number compared to the sampling rate required for λ_1 estimation with an underlying 1-D map. Construction of a map requires that orbital intersections with the Poincaré section be determined with high accuracy, often necessitating 100 or more points per orbit. Our technique thus allows a factor of about 10 times more orbits for a given size data file.

avoided that region at replacement time. The interactive runs determined the positive exponent to within 3%, and measured the second exponent as less than 5% of the first, using 8192 points. The fixed evolution time code measured the first exponent to within 5% and found the second exponent to be less than 10% of the first, using 8192 points in both cases.

Hyperchaos

For this system we obtained the largest exponent to within 10% using 8192 points and the sum of the two positive exponents to within 15% using 16384 points.

Delay differential equation

Using 65536 points, we computed the largest of the two positive exponents to within 10% and found the sum of the first two exponents to within 20%.

Belousov–Zhabotinskii reaction

In fig. 11 we show the result of our algorithm used on a time series of 65536 points spanning 400 orbital periods. The system was in a chaotic regime near a period-three window. The exponent calculated by the algorithm is stable over a range of parameter values. We also calculated the exponent using 1-D map analysis [2] as a comparison. Our algorithm gives a result in the plateau region of 0.0054 ± 0.0005 bits/s, while the 1-D map estimation yields a result of 0.0049 ± 0.0010 bits/s. Thus the estimates are in agreement.

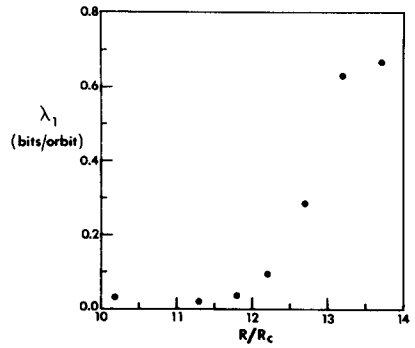

Fig. 12. The largest Lyapunov exponent for experimental Couette–Taylor data is shown as a function of Reynolds number. The shape of this curve (but not its absolute magnitude, see reference [3]) was independently verified by the calculation of the metric entropy h_μ – which is equal to λ_1 if there is a single positive exponent [38].

Couette–Taylor

For the Couette–Taylor experiment we computed the largest Lyapunov exponent as a function of Reynolds number from data sets (at each Reynolds number) consisting of 32768 points spanning about 200 mean orbital periods. Our results are given in fig. 12. Earlier studies of power spectra and phase portraits had indicated that the onset of chaos occurred at $R/R_c \simeq 12$, where R_c marks the transition to Taylor vortex flow. This onset of chaos is confirmed and quantified by the calculation of λ_1.

9. Conclusions

The algorithms we have presented can detect and quantify chaos in experimental data by accurately estimating the first few non-negative Lyapunov exponents. Moreover, our numerical studies have shown that deterministic chaos can be distinguished in some cases from external noise (as in the Belousov–Zhabotinskii attractor) and topological complexity (as in the Lorenz attractor). However, this requires a reasonable quantity of accurate data, and the attractor must not be of very high dimension.

As with other diagnostics used in chaotic dynamical systems, the calculation of Lyapunov exponents is still in its infancy, but we believe that the approach to exponent estimation that we have described here is workable. We encourage experimentation with our algorithms.

Acknowledgements

We thank J. Doyne Farmer for helpful discussions and for calculating the Lyapunov spectrum

of the delay differential equation; Mark Haye for assistance in programming the Vector General; and Shannon Spires for programming support, and for producing the Lyapunov exponent film. This research is supported by the Department of Energy, Office of Basic Energy Sciences contract DE-AS05-84ER13147. H. Swinney acknowledges the support of a Guggenheim Fellowship and J. Vastano acknowledges the support of an Exxon Fellowship.

Appendix A

Lyapunov spectrum program for systems of differential equations

This program computes the complete Lyapunov spectrum of a nonlinear system whose equations of motion and their linearizations are supplied by the user in the routine FCN. The program is set up here for the three variable Lorenz system but is easily modified for any other system of equations.

```
      PROGRAM ODE
C
C     N = # OF NONLINEAR EQTNS., NN = TOTAL # OF EQTNS.
C
      PARAMETER N=3
      PARAMETER NN=12
      EXTERNAL FCN
C
      DIMENSION Y(NN),ZNORM(N),GSC(N),CUM(N),C(24),W(NN,9)
C
C     INITIAL CONDITIONS FOR NONLINEAR SYSTEM
C
      Y(1) = 10.0
      Y(2) = 1.0
      Y(3) = 0.0
C
C     INITIAL CONDITIONS FOR LINEAR SYSTEM (ORTHONORMAL FRAME)
C
      DO 10 I = N+1,NN
         Y(I) = 0.0
   10 CONTINUE
      DO 20 I = 1,N
         Y((N+1)*I) = 1.0
         CUM(I) = 0.0
   20 CONTINUE
C
C     INTEGRATION TOLERANCE, # OF INTEGRATION STEPS,
C     TIME PER STEP, AND I/O RATE
C
      TYPE*, 'TOL,NSTEP,STPSZE,IO?'
      ACCEPT*,TOL,NSTEP,STPSZE,IO
C
C     INITIALIZATION FOR INTEGRATOR
C
      NEQ = NN
      X = 0.0
      IND = 1
```

```
C
      DO 100 I = 1,NSTEP
         XEND = STPSZE*FLOAT(I)
C
C     CALL ANY ODE INTEGRATOR - THIS IS AN IMSL ROUTINE
C
         CALL DVERK (NEQ,FCN,X,Y,XEND,TOL,IND,C,NEQ,W,IER)
C
C     CONSTRUCT A NEW ORTHONORMAL BASIS BY GRAM-SCHMIDT METHOD
C
C     NORMALIZE FIRST VECTOR
C
         ZNORM(1) = 0.0
         DO 30 J = 1,N
            ZNORM(1) = ZNORM(1)+Y(N*J+1)**2
   30    CONTINUE
         ZNORM(1) = SQRT(ZNORM(1))
         DO 40 J = 1,N

            Y(N*J+1) = Y(N*J+1)/ZNORM(1)
   40    CONTINUE
C
C     GENERATE THE NEW ORTHONORMAL SET OF VECTORS.
C
         DO 80 J = 2,N
C
C     GENERATE J-1 GSR COEFFICIENTS.
C
            DO 50 K = 1,(J-1)
               GSC(K) = 0.0
               DO 50 L = 1,N
                  GSC(K) = GSC(K)+Y(N*L+J)*Y(N*L+K)
   50       CONTINUE
C
C     CONSTRUCT A NEW VECTOR.
C
            DO 60 K = 1,N
               DO 60 L = 1,(J-1)
                  Y(N*K+J) = Y(N*K+J)-GSC(L)*Y(N*K+L)
   60       CONTINUE
C
C     CALCULATE THE VECTOR'S NORM
C
            ZNORM(J) = 0.0
            DO 70 K = 1,N
               ZNORM(J) = ZNORM(J)+Y(N*K+J)**2
   70       CONTINUE
            ZNORM(J) = SQRT(ZNORM(J))
C
C     NORMALIZE THE NEW VECTOR.
C
            DO 80 K = 1,N
               Y(N*K+J) = Y(N*K+J)/ZNORM(J)
   80    CONTINUE
```

```
C
C     UPDATE RUNNING VECTOR MAGNITUDES
C
         DO 90 K = 1,N
            CUM(K) = CUM(K)+ALOG(ZNORM(K))/ALOG(2.)
   90    CONTINUE
C
C     NORMALIZE EXPONENT AND PRINT EVERY IO ITERATIONS
C
         IF (MOD(I,IO).EQ.0) TYPE*,X,(CUM(K)/X,K = 1,N)
C
  100 CONTINUE
C
      CALL EXIT
      END
C
      SUBROUTINE FCN (N,X,Y,YPRIME)
C
C     USER DEFINED ROUTINE CALLED BY IMSL INTEGRATOR.
C
      DIMENSION Y(12),YPRIME(12)
C
C     LORENZ EQUATIONS OF MOTION
C
      YPRIME(1) = 16.*(Y(2)-Y(1))
      YPRIME(2) = -Y(1)*Y(3)+45.92*Y(1)-Y(2)
      YPRIME(3) = Y(1)*Y(2)-4.*Y(3)
C
C     3 COPIES OF LINEARIZED EQUATIONS OF MOTION.
C
      DO 10 I = 0,2
         YPRIME(4+I) = 16.*(Y(7+I)-Y(4+I))
         YPRIME(7+I) = (45.92-Y(3))*Y(4+I)-Y(7+I)-Y(1)*Y(10+I)
         YPRIME(10+I) = Y(2)*Y(4+I)+Y(1)*Y(7+I)-4.*Y(10+I)
   10 CONTINUE
C
      RETURN
      END
```

See section 3 and refs. 8, 9 for a discussion of the ODE algorithm.

Appendix B

Fixed evolution time program for λ_1

A time series (of length NPT) is read from a data file, along with the parameters necessary to reconstruct the attractor, namely, the dimension of the phase space reconstruction (DIM), the reconstruction time delay (TAU), and the time between the data samples (DT), required only for normalization of the exponent. Three other input parameters are required: length scales that we consider to be too large (SCALMX) and too small (SCALMN) and a constant propagation time (EVOLV) between replacement attempts. SCALMX is our estimate of the length scale on which the local structure of the attractor is no longer being probed; SCALMN is the length scale on which noise is expected to appear. We also supply a maximum angular error to be accepted at replacement time (ANGLMX), but we do not consider this a free parameter as its selection is not likely to have much effect on exponent estimates. It is usually fixed at 0.2 or 0.3 radians.

The calculation is initiated by carrying out an exhaustive search of the data file to locate the nearest neighbor to the first point (the fiducial point), omitting points closer than SCALMN†. The main program loop, which carries out repeated cycles of propagating and replacing the first principal axis vector is now entered. The current pair of points is propagated EVOLV steps through the attractor and its final separation is computed. The log of the ratio of final to initial separation of this pair updates a running average rate of orbital divergence. A replacement step is then attempted. The distance of each delay coordinate point to the evolved fiducial point is then determined. Points closer than SCALMX but further away than SCALMN are examined to see if the change in angular orientation is less than ANGLMX radians. If more than one candidate point is found, the point defining the smallest angular change is used for replacement. If no points satisfy these criteria, we loosen the larger distance criterion to accept replacement points as far as twice SCALMX away. If necessary the large distance criterion is relaxed several more times, at which point we tighten this constraint and relax the angular acceptance criterion. Continued failure will eventually result in our keeping the pair of points we had started out with, as this pair results in no change whatsoever in phase space orientation. We now go back to the top of the main loop where the new points are propagated. This process is repeated until the fiducial trajectory reaches the end of the data file, by which time we hope to see stationary behavior of λ_1. See section 6 for a discussion of how to choose the input parameters.

The fixed evolution time code for $\lambda_1 + \lambda_2$ estimation is too long to present here (350 lines of Fortran) but we discuss its structure briefly. This program begins by reading in a dynamical observable and many of the same input parameters as the code for λ_1 estimation. A number of parameters are also required to evaluate the quality of replacement triples: the maximum allowed triple "skewness", the maximum angular deviation of each replacement vector from the plane defined by the last triple, and weighting factors for the relative importance of skewness, size of replacement vectors, and angular errors in choosing replacement vectors.

The structure of this program is very similar to the program for λ_1: locate the two nearest neighbors of the first delay coordinate point, determine the initial area defined by this triple, enter the main program loop for repeated evolution and replacement. Triples are evolved EVOLV steps through the attractor and replacement is performed. Triple replacement is a more complicated process than pair replacement, which involved minimizing a single angular separation and a length. Our approach to triple replacement is a two step process; first we find a small set of points which, together with the fiducial trajectory, define small separation vectors and lie close to the required two-dimensional subspace. We then determine which of all of the possible pairs of these points will make the best replacement triple. In the first step, the qualifications of up to 20 potential replacement points are saved. Separation and orientation requirements of replacement points are varied so that a moderate number of candidates is almost always obtained. In the second step every possible pair of these points is assigned a score based on how close the replacement triple is to the old two-dimensional subspace and how numerically stable the orientation of the triple is believed to be. (It is possible that the individual replacement *vectors* lie very close to the old two-dimensional subspace but that the replacement *area element* is nearly orthogonal to the same subspace!) The relative importance of replacement lengths, skewness, orientation changes, etcetera, are weighted by the user chosen factors. The highest scoring pair of points is used in the replacement triple. The calculations in this program involve dot products and the Gram–Schmidt process and so are independent of the dimension of the reconstructed attractor – no complicated geometry is required in the coding.

†Such an exhaustive search is very inefficient for large arrays; then more efficient algorithms should be employed. See, for example, ref. 39.

```
      PROGRAM FET1
      INTEGER DIM,TAU,EVOLV
      DIMENSION X(16384),PT1(12),PT2(12)
C
C     **DEFINE DELAY COORDINATES WITH A STATEMENT FUNCTION**
C     **Z(I,J)=JTH COMPONENT OF ITH RECONSTRUCTED ATTRACTOR POINT**
C
      Z(I,J) = X(I+(J-1)*TAU)
C
      OPEN (UNIT=1,FILE='INPUT.',TYPE='OLD')
C
      TYPE*, 'NPT,DIM,TAU,DT,SCALMX,SCALMN,EVOLV ?'
      ACCEPT*,NPT,DIM,TAU,DT,SCALMX,SCALMN,EVOLV
C
C     **IND POINTS TO FIDUCIAL TRAJECTORY**
C     **IND2 POINTS TO SECOND TRAJECTORY**
C     **SUM HOLDS RUNNING EXPONENT ESTIMATE SANS 1/TIME**
C     **ITS IS TOTAL NUMBER OF PROPAGATION STEPS**
C
      IND = 1
      SUM = 0.0
      ITS = 0
C
C     **READ IN TIME SERIES**
C
      DO 10 I = 1,NPT
         READ (1,*) X(I)
   10 CONTINUE
C
C     **CALCULATE USEFUL SIZE OF DATAFILE
C
      NPT = NPT - DIM*TAU - EVOLV
C
C     **FIND NEAREST NEIGHBOR TO FIRST DATA POINT**
C
      DI = 1.E38
C
C     **DONT TAKE POINT TOO CLOSE TO FIDUCIAL POINT**
C
      DO 30 I = 11,NPT
C
C     **COMPUTE SEPARATION BETWEEN FIDUCIAL POINT AND CANDIDATE**
C
         D = 0.0
         DO 20 J = 1,DIM
            D = D+(Z(IND,J)-Z(I,J))**2
   20    CONTINUE
         D = SQRT(D)
C
C     **STORE THE BEST POINT SO FAR BUT NO CLOSER THAN NOISE SCALE**
C
         IF (D.GT.DI.OR.D.LT.SCALMN) GO TO 30
         DI = D
         IND2 = I
   30 CONTINUE
```

```
C
C     **GET COORDINATES OF EVOLVED POINTS**
C
   40 DO 50 J = 1,DIM
         PT1(J) = Z(IND+EVOLV,J)
         PT2(J) = Z(IND2+EVOLV,J)
   50 CONTINUE
C
C     **COMPUTE FINAL SEPARATION BETWEEN PAIR, UPDATE EXPONENT**
C
      DF = 0.0
      DO 60 J = 1,DIM
         DF = DF+(PT1(J)-PT2(J))**2
   60 CONTINUE
      DF = SQRT(DF)
      ITS = ITS+1
      SUM = SUM+ALOG(DF/DI)/(FLOAT(EVOLV)*DT*ALOG(2.))
      ZLYAP = SUM/FLOAT(ITS)
      TYPE*,ZLYAP,EVOLV*ITS,DI,DF
C
C     **LOOK FOR REPLACEMENT POINT**
C     **ZMULT IS MULTIPLIER OF SCALMX WHEN GO TO LONGER DISTANCES**
C
      INDOLD = IND2
      ZMULT = 1.0
      ANGLMX = 0.3
   70 THMIN = 3.14
C
C     **SEARCH OVER ALL POINTS**
C
      DO 100 I = 1,NPT
C
C     **DONT TAKE POINTS TOO CLOSE IN TIME TO FIDUCIAL POINT**
C
         III = IABS(I-(IND+EVOLV))
         IF (III.LT.10) GO TO 100
C
C     **COMPUTE DISTANCE BETWEEN FIDUCIAL POINT AND CANDIDATE**
C
         DNEW = 0.0
         DO 80 J = 1,DIM
            DNEW = DNEW+(PT1(J)-Z(I,J))**2
   80    CONTINUE
         DNEW = SQRT(DNEW)
C
C     **LOOK FURTHER AWAY THAN NOISE SCALE, CLOSER THAN ZMULT*SCALMX**
C
         IF (DNEW.GT.ZMULT*SCALMX.OR.DNEW.LT.SCALMN) GO TO 100
C
C     **FIND ANGULAR CHANGE OLD TO NEW VECTOR**
C
         DOT = 0.0
         DO 90 J = 1,DIM
            DOT = DOT+(PT1(J)-Z(I,J))*(PT1(J)-PT2(J))
   90    CONTINUE
```

```
            CTH = ABS(DOT/(DNEW*DF))
            IF (CTH.GT.1.0) CTH = 1.0
            TH = ACOS(CTH)
C
C     **SAVE POINT WITH SMALLEST ANGULAR CHANGE SO FAR**
C
            IF (TH.GT.THMIN) GO TO 100
            THMIN = TH
            DII = DNEW
            IND2 = I
  100 CONTINUE
      IF (THMIN.LT.ANGLMX) GO TO 110
C
C     **CANT FIND A REPLACEMENT - LOOK AT LONGER DISTANCES**
C
      ZMULT = ZMULT+1.
      IF (ZMULT.LE.5.) GO TO 70
C
C     **NO REPLACEMENT AT 5*SCALE, DOUBLE SEARCH ANGLE, RESET DISTANCE**
C
      ZMULT = 1.0
      ANGLMX = 2.*ANGLMX
      IF (ANGLMX.LT.3.14) GO TO 70
      IND2 = INDOLD + EVOLV
      DII = DF
  110 CONTINUE
      IND = IND+EVOLV
C
C     **LEAVE PROGRAM WHEN FIDUCIAL TRAJECTORY HITS END OF FILE**
C
      IF (IND.GE.NPT) GO TO 120
      DI = DII
      GO TO 40
  120 CALL EXIT
      END
```

References

[1] See the references in: H.L. Swinney, "Observations of Order and Chaos in Nonlinear Systems," Physica 7D (1983) 3 and in: N.B. Abraham, J.P. Gollub and H.L. Swinney, "Testing Nonlinear Dynamics," Physica 11D (1984) 252.

[2] J.-C. Roux, R.H. Simoyi and H.L. Swinney, "Observation of a Strange Attractor," Physica 8D (1983) 257.

[3] A. Brandstater, J. Swift, H.L. Swinney, A. Wolf, J.D. Farmer, E. Jen and J.P. Crutchfield, "Low-Dimensional Chaos in a Hydrodynamic System," Phys. Rev. Lett 51 (1983) 1442.

[4] B. Malraison, P. Atten, P. Berge and M. Dubois, "Turbulence-Dimension of Strange Attractors: An Experimental Determination for the Chaotic Regime of Two Convective Systems," J. Physique Lettres 44 (1983) L-897.

[5] J. Guckenheimer and G. Buzyna, "Dimension Measurements for Geostrophic Turbulence," Phys. Rev. Lett. 51 (1983) 1438.

[6] J.P. Gollub, E.J. Romer and J.E. Socolar, "Trajectory divergence for coupled relaxation oscillators: measurements and models," J. Stat. Phys. 23 (1980) 321.

[7] V.I. Oseledec, "A Multiplicative Ergodic Theorem. Lyapunov Characteristic Numbers for Dynamical Systems," Trans. Moscow Math. Soc. 19 (1968) 197.

[8] G. Benettin, L. Galgani, A. Giorgilli and J.-M. Strelcyn,

"Lyapunov Characteristic Exponents for Smooth Dynamical Systems and for Hamiltonian Systems; A Method for Computing All of Them," Meccanica 15 (1980) 9.

[9] I. Shimada and T. Nagashima, "A Numerical Approach to Ergodic Problem of Dissipative Dynamical Systems," Prog. Theor. Phys. 61 (1979) 1605.

[10] R. Shaw, "Strange Attractors, Chaotic Behavior, and Information Flow," Z. Naturforsch. 36A (1981) 80.

[11] J.L. Hudson and J.C. Mankin, "Chaos in the Belousov–Zhabotinskii Reaction," J. Chem. Phys. 74 (1981) 6171.

[12] H. Nagashima, "Experiment on Chaotic Response of Forced Belousov–Zhabotinskii Reaction," J. Phys. Soc. Japan 51 (1982) 21.

[13] A. Wolf and J. Swift, "Progress in Computing Lyapunov Exponents from Experimental Data," in: Statistical Physics and Chaos in Fusion Plasmas, C.W. Horton Jr. and L.E. Reichl, eds. (Wiley, New York, 1984).

[14] J. Wright, "Method for Calculating a Lyapunov Exponent," Phys. Rev. A29 (1984) 2923.

[15] S. Blacher and J. Perdang, "Power of Chaos," Physica 3D (1981) 512.

[16] J.P. Crutchfield and N.H. Packard, "Symbolic Dynamics of Noisy Chaos," Physica 7D (1983) 201.

[17] P. Grassberger and I. Procaccia, "Estimation of the Kolmogorov Entropy from a Chaotic Signal," Phys. Rev. A28 (1983) 2591.

[18] R. Shaw, The Dripping Faucet, (Aerial Press, Santa Cruz, California, 1984).

[19] J.D. Farmer, E. Ott and J.A. Yorke, "The Dimension of Chaotic Attractors," Physica 7D (1983) 153.

[20] S. Ciliberto and J.P. Gollub, "Chaotic Mode Competition in Parametrically Forced Surface Waves"—preprint.

[21] P. Grassberger and I. Procaccia, "Characterization of Strange Attractors," Phys. Rev. Lett. 50 (1983) 346.

[22] H. Haken, "At Least One Lyapunov Exponent Vanishes if the Trajectory of an Attractor Does Not Contain a Fixed Point," Phys. Lett. 94A (1983) 71.

[23] E.N. Lorenz, "Deterministic Nonperiodic Flow," J. Atmos. Sci. 20 (1983) 130.

[24] O.E. Rossler, "An Equation for Hyperchaos," Phys. Lett. 71A (1979) 155.

[25] M. Hénon, "A Two-Dimensional Mapping with a Strange Attractor," Comm. Math. Phys. 50 (1976) 69.

[26] O.E. Rossler, "An Equation for Continuous Chaos," Phys. Lett. 57A (1976) 397.

[27] M.C. Mackey and L. Glass, "Oscillation and Chaos in Physiological Control Systems," Science 197 (1977) 287.

[28] J. Kaplan and J. Yorke, "Chaotic behavior of multidimensional difference equations," in: Functional Differential Equations and the Approximation of Fixed Points, Lecture Notes in Mathematics, vol. 730, H.O. Peitgen and H.O. Walther, eds. (Springer, Berlin), p. 228; P. Frederickson, J. Kaplan, E. Yorke and J. Yorke, "The Lyapunov Dimension of Strange Attractors," J. Diff. Eqs. 49 (1983) 185.

[29] L.-S. Young, "Dimension, Entropy, and Lyapunov Exponents," Ergodic Theory and Dynamical Systems 2 (1982) 109. F. Ledrappier, "Some Relations Between Dimension and Lyapunov Exponents," Comm. Math. Phys. 81 (1981) 229.

[30] D.A. Russell, J.D. Hanson and E. Ott, "Dimension of Strange Attractors," Phys. Rev. Lett. 45 (1980) 1175.

[31] D. Ruelle, private communication.

[32] After R. Shaw, unpublished.

[33] N.H. Packard, J.P. Crutchfield, J.D. Farmer and R.S. Shaw, "Geometry from a Time Series," Phys. Rev. Lett. 45 (1980) 712.

[34] F. Takens, "Detecting Strange Attractors in Turbulence," in Lecture Notes in Mathematics, vol. 898, D.A. Rand and L.-S. Young, eds. (Springer, Berlin, 1981), p. 366.

[35] H.S. Greenside, A. Wolf, J. Swift and T. Pignataro, "The Impracticality of a Box-Counting Algorithm for Calculating the Dimensionality of Strange Attractors," Phys. Rev. A25 (1982) 3453.

[36] J.-C. Roux and A. Rossi, "Quasiperiodicity in chemical dynamics," in: Nonequilibrium Dynamics in Chemical Systems, C. Vidal and A. Pacault, eds. (Springer, Berlin, 1984), p. 141.

[37] J.P. Crutchfield, J.D. Farmer and B.A. Huberman, "Fluctuations and Simple Chaotic Dynamics," Phys. Rep. 92 (1982) 45.

[38] R. Bowen and D. Ruelle, "The Ergodic Theory of Axiom-A Flows," Inv. Math. 29 (1975) 181. D. Ruelle, "Applications conservant une mesure absolument continué par rapport à dx sur [0,1]," Comm. Math. Phys. 55 (1977) 47.

[39] D.E. Knuth, The Art of Computer Programming, vol. 3 – Sorting and Searching, (Addison–Wesley, Reading, Mass., 1975).