Computing the Lyapunov spectrum of a dynamical system from an observed time series

Reggie Brown and Paul Bryant
Institute for Nonlinear Science, University of California, San Diego, Mail Code R-002, La Jolla, California 92093-0402

Henry D. I. Abarbanel*
Department of Physics and Marine Physical Laboratory, Scripps Institution of Oceanography, University of California, San Diego, Mail Code R-002, La Jolla, California 92093-0402

(Received 21 May 1990; revised manuscript received 9 August 1990)

We examine the question of accurately determining, from an observed time series, the Lyapunov exponents for the dynamical system generating the data. This includes positive, zero, and some or all of the negative exponents. We show that even with very large data sets, it is clearly advantageous to use local neighborhood-to-neighborhood mappings with higher-order Taylor series, rather than just local linear maps as has been done previously. We give examples using up to fifth-order polynomials. We demonstrate this procedure on two familiar maps and two familiar flows: the Hénon and Ikeda maps of the plane to itself, the Lorenz system of three ordinary differential equations, and the Mackey-Glass delay differential equation. We stress the importance of maintaining two dimensions for converting the scalar data into time delay vectors: one is a global dimension to ensure proper unfolding of the attractor as a whole, and the other is a local dimension for capturing the local dynamics on the attractor. We show the effects of changing the local and global dimensions, changing the order of the mapping polynomial, and additive (measurement) noise. There will always be some limit to the number of exponents that can be accurately determined from a given finite data set. We discuss a method of determining this limit by numerically obtaining the singularity spectra of the data set and also show how it is often appropriate to make this choice based on the fractal dimension of the attractor. If excessively large dimensions are used, spurious exponents will be generated, and in some cases the accuracy of the true exponents will be affected. We present methods of identifying these spurious exponents by determining the Lyapunov direction vectors at particular points in the data set. We can then use these to identify numerical problems and to associate data-set singularities with particular exponents. The behavior of spurious exponents in the presence of noise is also investigated, and found to be different from that of the true exponents. These provide methods for identifying spurious exponents in the analysis of experimental data where the system dynamics may not be known *a priori.*

I. INTRODUCTION

Lyapunov exponents of dynamical systems are one of a number of invariants that characterize the attractors of the system in a fundamental way. They are independent of initial conditions on any orbit[1,2] and, thus, are properties of the attractor geometry and the dynamics. Attractors can be thought of as a distribution of points in a phase, or state, space characterized by the density of points. It is known that all moments of the density are invariants of the evolution which moves points on the attractor forward in time. The Lyapunov exponents are singled out by their easy interpretation with regard to the stability of the dynamics and their connection with the metric or Kolmogorov-Sinai entropy[3,4] of the system through the Pesin inequality.[1,5]

The determination of Lyapunov exponents from known differential equations or maps is numerically[6] subtle but reasonably straightforward in concept. They are determined by looking at some "fiducial" orbit $\mathbf{w}(k)\in\mathbb{R}^d$; $k=1,2,\ldots,N$, in the phase space of the system, and observing the evolution of small deviations $\delta\mathbf{w}(k)$ from the fiducial orbit. For our discussion we will assume the dynamics is a map from $\mathbb{R}^d$ to itself, which means we have discretized time in a flow, or have taken a Poincaré section of a flow, or are dealing with a map from the outset. If the orbit is taken to satisfy $\mathbf{w}(k+1)=\mathbf{f}(\mathbf{w}(k))$, then a perturbation to the orbit satisfies

$$\delta\mathbf{w}(k+1)=\underline{D}\,\mathbf{f}(\mathbf{w}(k))\delta\mathbf{w}(k)\ ,$$

where $\underline{D}\,\mathbf{f}(\mathbf{w})$ is the $d\times d$ Jacobian matrix evaluated along the fiducial orbit. If we take the product of these matrices for K steps along the orbit

$$\underline{D}\,\mathbf{f}^K=\underline{D}\,\mathbf{f}(K)\cdot\underline{D}\,\mathbf{f}(K-1)\cdots\underline{D}\,\mathbf{f}(1)\ ,$$

where $\underline{D}\,\mathbf{f}(K)=\underline{D}\,\mathbf{f}(\mathbf{w}(K))$, then the Oseledec[2] multiplicative ergodic theorem says that the Lyapunov exponents are the logarithms of the eigenvalues of the matrix

$$\lim_{K\to\infty}[(\underline{D}\,\mathbf{f}^K)^\dagger(\underline{D}\,\mathbf{f}^K)]^{1/2K}\ ,$$

where † signifies transpose. The only problem in implementing this comes from the ill-conditioned nature of the matrices $\underline{D}\mathbf{f}^K$, but with care this can be handled. The Lyapunov exponents that come out of this procedure we will identify as λ_i and always refer to in order of their numerical size: $\lambda_1 \geq \lambda_2 \geq \cdots$.

When dealing with experimental data, rather than known maps or flows, one has the initial problem of finding the analogue of $\underline{D}\mathbf{f}$. Once a good representation of this matrix is in hand, the numerical problems suggested above come to the fore. In measurements of experimental data one often has an accurate determination of only a single variable which represents the multivariate state of the system. There is a familiar technique known as phase-space reconstruction, originally suggested by Ruelle,[1] for recapturing the multivariate state. It has become the tool of choice for recreating the *vector* phase space in which to capture the behavior of the dynamical system that produces the observed *scalar* time series. The method uses scalar measured data $x(n)$, $n=1,\ldots,N_D$, taken at times $t_n = t_0 + n\tau$, and converts it to vectors in a d-dimensional Euclidean space R^d. Time delays are used to form d-dimensional vectors $\mathbf{y}(n)$ from the x's as

$$\mathbf{y}(n)=[x(n),x(n+T),\ldots,x(n+(d-1)T)]$$

(Refs. 1, 7, and 8). (We use the same lag for each component of the vector $\mathbf{y}$, though this is not necessary.) Using the entire time series $x(n)$, $n=1,\ldots,N_D$, one can construct a total of $N=N_D-d$, d-dimensional vectors. As they evolve in $\mathbb{R}^d$ these vectors, $\mathbf{y}(n)$, $n=1,\ldots,N$, define the attractor. This provides the fiducial trajectory for the analysis of Lyapunov exponents.

We will assume that the evolution of the $\mathbf{y}(k)$ is given by some map or rule which takes $\mathbf{y}(k) \rightarrow \mathbf{y}(k+T_2)$ via $\mathbf{y}(k+T_2)=\mathbf{F}(\mathbf{y}(k))$. T_2, the iteration step, may be chosen independently of T, and this choice may have a significant effect on the accuracy of the results. For each point $\mathbf{y}(n)$ on the fiducial trajectory there are other nearby points in $\mathbb{R}^d$. "Nearby" is measured by some norm in $\mathbb{R}^d$. If we let $\mathbf{y}^r(n;0)$ be the rth nearest neighbor to $\mathbf{y}(n)$, then the work of Eckmann *et al.*[9] (EKRC) for extracting the Lyapunov exponents from data involves determining *local linear maps* $\underline{D}\mathbf{F}(n)$, that map whole neighborhoods of small vectors

$$\mathbf{z}^r(n;0)=\mathbf{y}^r(n;0)-\mathbf{y}(n)$$

into succeeding neighborhoods of small vectors

$$\mathbf{z}^r(n;T_2)=\mathbf{y}^r(n;T_2)-\mathbf{y}(n+T_2) .$$

In our notation $\mathbf{y}^r(n;k)$ is the position of the rth nearest neighbor to $\mathbf{y}(n)$ after evolving forward to time $n+k$. Note that $\mathbf{y}^r(n;k) \neq \mathbf{y}^r(n+k;0)$ because a new set of nearest neighbors is to be found at each time step. For the local neighborhood-to-neighborhood map we need the locations, after one time step, of the neighbors at the previous step. To determine the Lyapunov spectrum of the dynamical system, one calculates the eigenvalues of the matrix $(\underline{D}\mathbf{F}^K)^\dagger \underline{D}\mathbf{F}^K$ where

$$\underline{D}\mathbf{F}^K=\underline{D}\mathbf{F}(K)\cdot\underline{D}\mathbf{F}(K-1)\cdots\underline{D}\mathbf{F}(1)$$

and the dots represent matrix multiplication.[1]

Our work begins where the EKRC paper ends. We have found[10] by examining this method and that of others which utilize rather similar approaches[11,12,13] that one can reliably determine only the largest Lyapunov exponent. The difficulty primarily arises from the fact that the attractor is a fractal structure which does not fill the phase space completely. Since all of the data points lie on the attractor, there will often be a lack of information about the behavior in certain directions which have more to do with transient decay than with the dynamics within the attractor itself. This can lead to gross inaccuracies in the determination of the Jacobian matrices $\underline{D}\mathbf{F}^K$. Inaccuracies in the individual $\underline{D}\mathbf{F}(k)$'s which comprise $\underline{DF}^K$ are severely magnified by the ill-conditioned nature of $\underline{D}\mathbf{F}^K$. With careful analysis, and sufficiently accurate data, substantial improvements can be made in the determination of the Lyapunov spectrum. We shall demonstrate this quite explicitly in what follows.

In this paper we examine several questions related to highly accurate determination of the Lyapunov exponents. These include:

(1) What improvements, if any, can be had by using higher-order local polynomial fits in the step-to-step mapping of small neighborhood intervals $\mathbf{z}^r(n;0)$ to small neighborhood intervals $\mathbf{z}^r(n;T_2)$? The idea is that by determining more than just the linear term in the local neighborhood mapping, we can stabilize and make more accurate the evaluation of the term we want. Another way to look at our procedure is that we have separated the problem of mapping neighborhoods to neighborhoods from the problem of finding the Jacobian. By having a richer local map, we are able to spread the burden of mapping local neighborhoods to local neighborhoods over all terms in the polynomial. In this way we can achieve greater accuracy in determining the Jacobian.

(2) What is the effect of data accuracy of Lyapunov exponents?

(3) Under certain conditions (such as when the local dimension is made too large), there will be spurious exponents which are largely a numerical artifact—how can these be identified or avoided?

(4) Can we accurately determine some of the negative Lyapunov exponents as well as any positive or zero exponents? Large negative exponents will cause the attractor to be "thin" in places, making the calculation of negative exponents difficult. (This thinness is often disguised to the naked eye by the global folding that is inherent in chaotic attractors.)

(5) Are some of the Lyapunov exponents fundamentally more important to the dynamics within the attractor than other (more negative) exponents, thus making them more easily calculated and setting a natural cutoff point for the number of exponents to calculate in high-dimensional systems?

(6) By what criteria may we choose the time intervals T and T_2?

An outline of the rest of this paper is as follows: In Sec. II we present a detailed discussion of the methods we used in examining the questions posed above. In Sec. III we present the results of our numerical experiments on four familiar dynamical systems:

(i) The Hénon map of the plane[14] to itself,

$$x_1(n+1)=1-ax_1(n)^2+x_2(n)\ ,$$
$$x_2(n+1)=bx_1(n)\ , \tag{1}$$

where a and b are given the standard values 1.4 and 0.3, respectively. For these parameters the accepted values for λ_1 and λ_2 are 0.408 and -1.62, respectively. They satisfy $\lambda_1+\lambda_2=\ln(b)$.

(ii) The Ikeda map of the complex z plane[15,16] to itself,

$$z(n+1)=p+Bz(n)\exp\{i\kappa-i\alpha/[1+|z(n)|^2]\}\ , \tag{2}$$

where $p=1.0$, $B=0.9$, $\kappa=0.4$, and $\alpha=6.0$. For these parameters we have calculated (using the map) that λ_1 and λ_2 are 0.503 and -0.719, respectively. The Lyapunov exponents satisfy $\lambda_1+\lambda_2=2\ln(B)$.

(iii) The Lorenz system of three ordinary differential equations,[17]

$$\begin{aligned}\frac{dx_1}{dt}(t)&=\sigma[x_2(t)-x_1(t)]\ ,\\ \frac{dx_2}{dt}(t)&=-x_1(t)x_3(t)+rx_1(t)-x_2(t)\ ,\\ \frac{dx_3}{dt}(t)&=x_1(t)x_2(t)-bx_3(t)\ ,\end{aligned} \tag{3}$$

where we take $\sigma=16$, $b=4$, and $r=45.92$. For these parameters the accepted values for the Lyapunov exponents λ_1, λ_2, and λ_3 are 1.50, 0.00, and -22.5, respectively. The large negative exponent makes this system a particularly challenging test for our time-series method. The Lyapunov exponents satisfy the rule

$$\lambda_1+\lambda_2+\lambda_3=-\sigma-b-1\ .$$

(iv) The time-delay equation of Mackey and Glass,[18,19]

$$\frac{dX}{dt}=\frac{aX(t-s)}{1+[X(t-s)]^c}-bX(t)\ , \tag{4}$$

where we used $a=0.2$, $b=0.1$, $c=10.0$, and $s=17.0$. This system is infinite dimensional (because it is a time-delay equation) and, thus, has an infinite number of Lyapunov exponents. The first three exponents have approximately the following values: 0.006, 0.0, and -0.04 (see Ref. 19, Fig. 10).

Our reasons for choosing this set of examples as our laboratory for analyzing Lyapunov exponents rest primarily on the fact they are all well studied, they exhibit low-dimensional chaos, and they represent a diverse set of phenomena. Other systems could, of course, be studied, but we have not done so. Section IV contains a summary and concluding remarks.

We end this section by noting that the answer to each of the items that will concern us in this paper is a positive one. We will demonstrate that when one works with clean data it is possible to determine the positive, zero, and one or more of the negative exponents. We will also show that the accuracy of the data matters a great deal. Inaccurate data is like noisy data, and the points in phase space are not properly located. This causes errors in the determination of the Jacobian matrices along an orbit and spoils one's ability to evaluate the full spectrum of Lyapunov exponents.

Since the Lyapunov exponents act as classifiers of the dynamical system with clear physical meaning, the ability to capture them from data is useful in any circumstance. The programs to do this starting from measurements of a scalar variable in an experiment are part of our results. We have not reproduced these codes in the paper, but we will make them available to any reader who wishes them.

II. METHODS AND THEORETICAL SETTING

A. Scalar time series

As is the case with previous works on this subject, we will focus our attention on the simplest possible format for the data—namely, an observed sequence of real numbers $x(n)$; $n=1,\ldots,N_D$, taken at times $t=t_0=n\tau$ (τ is the mean time between measurements). In addition, we may have some information regarding the intrinsic noise level or, alternatively, the number of significant digits of accuracy p. Except when the noise level or digits of accuracy is specifically stated, it may be assumed that the numerically generated data we analyze have at least eight digits of accuracy. The $x(n)$'s would probably be obtained by recording the value of x at periodic time intervals. Another possibility is that the data were obtained by Poincaré or stroboscopic section, i.e., one data point (or some fixed number of data points) is taken during each oscillatory cycle of the dynamical system according to some rule. Often, the maximum value of the variable is recorded each cycle. This is a good choice since the derivative is passing through zero and, therefore, there should be less measurement error. Since the system is presumably in a chaotic state, the time intervals between consecutive data points on the Poincaré section may not be constant. For our work we assume that the time series is long: $N_D\gg 1$.

We use the familiar time-lag method to construct a d-dimensional Euclidean space in which to reconstruct the dynamics of the system.[7,8] From the $x(n)$'s we make d-dimensional vectors $\mathbf{y}(n)$,

$$\mathbf{y}(n)=[x(n),x(n+T),\ldots,x(n+(d-1)T)]\ ,$$

where T is fixed. The vectors $\mathbf{y}(n)$, $n=1,\ldots,N$, evolve on an attractor that represents the dynamics of the physical system in question. Time evolution on the attractor is given by $\mathbf{y}(n)\rightarrow\mathbf{y}(n+1)$.

The calculation of Lyapunov exponents requires an estimation of the mapping from the neighborhood of one vector to the neighborhood of a subsequent vector with a specified time increment T_2 between these two vectors, i.e., $\mathbf{y}(n)\rightarrow\mathbf{y}(n+T_2)$. This second increment T_2 is often chosen to be equal to the first, T. In general, however, this is a poor choice for reasons we will discuss presently.

For data obtained by measurements at fixed time intervals, there will often be a zero exponent in the Lyapunov spectrum. (This must occur when the dynamical system under observation is described by a set of ordinary differential equations.) The zero exponent corresponds to displacements along the orbit. However, for Poincaré sectioned data, there can be no displacement in this direction, and thus the zero exponent will not be present. Acquiring data on a surface of section is *not* a trivial exercise for experimentalists. One must be very careful to avoid fluctuations in the sectioning data. Fluctuations are equivalent to errors and will disrupt the values of all of the calculated exponents.

B. Local and global dimensions

In our analysis we identify two dimensions for converting the scalar data into time-delay vectors. The first dimension we call the global dimension d_G. We call the second dimension the local dimension d_L.

The global dimension is often called the embedding dimension. It must be chosen large enough so that conversion of the scalar data set, $x(n)$, into d_G-dimensional vectors $\mathbf{y}(n)$, constitutes an embedding. In the generic case a formal result due to Mañé and Takens[8,7] says that if d_a is the dimension of the attractor, then a sufficient condition on d_G is given by

$$d_G > 2d_a \ . \tag{5}$$

If d_G is *not* chosen large enough, then the attractor (in the time-delay representation) will be folded in such a way that it crosses itself in certain places. Under these circumstances a "small" neighborhood in the crossover region would contain points from disparate portions of the attractor. Now, imagine choosing a point in the crossover region and calculating its nearest neighbors in this d_G-dimensional space. When neighbors are determined by their displacement (d_G-dimensional Euclidean norm), there will be no way of distinguishing the two disparate parts of the attractor. Hence two points could be considered "neighbors" when they are actually on opposite sides of the attractor. More formally, the time-delay representation in d_G dimensions would *not* be diffeomorphic to the original attractor. In general, the self-intersections will have dimension $2d_a - d_G$, and all self-intersections can be avoided by the sufficiency condition, Eq. (5).

This is illustrated in Fig. 1 where we show the mapping of a one-dimensional object ($d_a = 1$) in three-, two-, and one-dimensional spaces. The top illustrates a successful embedding of the object into $d_G = 3$ dimensions. We have avoided all self-intersections. The middle illustrates an attempt to embed the object into $d_G = 2$ dimensions. Notice that the region where the attractor intersects itself has been reduced to a discrete set of points. Finally, we attempt to embed the data into $d_G = 1$. For this situation the attractor is folded onto itself over its entire length, and the regions of intersection are line segments.

The local dimension d_L is the number of dimensions necessary to capture the geometry of a small neighborhood of the attractor *after* it has been successfully embedded (that is, in the time-lag representation the $\mathbb{R}^{d_G}$-dimensional vector $\mathbf{y}(n)$ evolve on an attractor that is diffeomorphic to the original attractor). The example shown in Fig. 1 is an attractor that is properly embedded in $\mathbb{R}^{d_G}$, where $d_G = 3$. However, the dynamics takes place on a one-dimensional subsurface of $\mathbb{R}^3$. In other words, in any small neighborhood of the attractor, the attractor itself (and hence the dynamics) is one dimensional and $d_L = 1$.

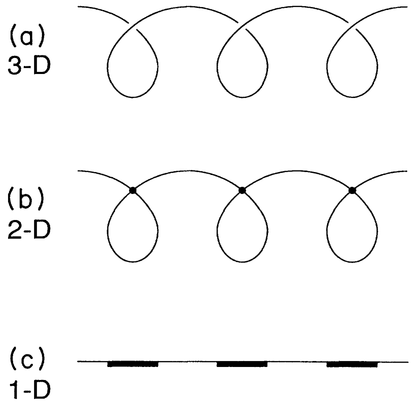

FIG. 1. Significance of the global dimension d_G; a simple illustration: (a) a line mapped into a 3D space can coil around without intersecting itself exactly, (b) projecting down to a 2D space, the line now has self-intersections of dimension 0, i.e., points; (c) projecting down to a 1D space, the line develops self-interactions of dimension 1, i.e., line segments. This last case is much more serious than (b) since the ability to distinguish true "neighbors" of a given point on the attractor (the line) has been lost for a large fraction of the total attractor. Thus an object that is 1D locally requires a global embedding dimension $d_G = 3$ to avoid all self-intersections.

The distinction between local and global dimensions is important in time-series analysis. For example, the Lorenz system described in Sec. I represents an attractor whose dimension d_a is slightly larger than 2. Hence, the attractor has a local dimension of $d_L = 3$ (the next integer greater than d_a). For this system it is known that the global dimension for embedding scalar data is also $d_G = 3$. The Lorenz equations represent a system where the sufficient condition given by Eq. (5) is not a necessary condition. This is not always the case. For example, the Ikeda map was originally derived as a relationship between the incoming and outgoing complex amplitudes in a laser cavity.[15] Therefore, one knows that its local dimension is $d_L = 2$, corresponding to the real and imaginary part of a complex phase. Yet some time-series representations require $d_G = 4$ to unfold the attractor (cf. Sec. III). This results in the dynamics evolving on a two-dimensional subsurface of a four-dimensional Euclidean space. In Sec. III we present results where we exploit the difference between local and global dimensions.

There are a variety of methods available for determining d_G, the global dimension for embedding the data. Perhaps the best-known method involves the Grassberger-Proccacia correlation integral.[20] This is the method we have used when calculating the results we present in Sec. III. Determining the local dimension d_L from data is a much more difficult issue. As we have mentioned above, there are some examples (Ikeda being one of them) where the dimension of the underlying dynamics is known, *a priori.* For these cases, just set d_L equal to that value. Or one may have knowledge of the dimension of the attractor, d_a. In that case d_L can be set to the next higher integer,

$$d_a \le d_L < d_a + 1 \ . \tag{6}$$

To determine d_L by this method we do not need the precise value of the attractor fractal dimension d_a; we only need know what the next larger integer is.

Our approach to determining an appropriate cutoff value for the number of exponents can be related to the Lyapunov dimension. The idea that there is a strong connection between the values of the Lyapunov exponents and the fractal dimension was originally explored by Kaplan and Yorke,[21,22] who introduced the concept of the Lyapunov dimension. This dimension measure, whose value is believed to be at least close, if not equal, to values obtained for dimension by other methods, has some important implications for determining Lyapunov exponents from a time series. The Lyapunov dimension is found as follows: Find the maximum number k of the Lyapunov exponents that can be added together before the sum $\sum_{m=1}^{k}\lambda_m$ becomes negative. The Lyapunov dimension is defined by

$$D_L = k + \frac{\sum_{m=1}^{k}\lambda_m}{-\lambda_{k+1}} \ . \tag{7}$$

Thus the dimension is determined by only a finite number of the exponents and does not depend on exponents beyond the $(k+1)$st. This strongly suggests that the first $k+1$ exponents must, in some sense, be fundamentally important to the character of the attractor. Exponents which are beyond $k+1$ are of lesser significance and their influence may tend to diminish as we look at the structure of the attractor on smaller and smaller scales. The attractor has a space-filling character along the first k Lyapunov directions and a fractal character along the $(k+1)$st direction. If any of the first k exponents are negative, they are of insufficient strength to promote the formation of fractal structure, since they cannot overcome the rate of volume expansion achieved by the positive exponents. Only when we add the $(k+1)$st Lyapunov direction to our observation does the evolution become dissipative, having a net rate of volume decrease so that collapse onto a fractal is possible. Displacements in additional directions will decay more rapidly because the corresponding exponents are more negative than the $(k+1)$st and may become unimportant at sufficiently small scales. This suggests both that these additional Lyapunov exponents will be harder to determine from data on the attractor and also that they are of less interest in terms of understanding the nature of the attractor. If one takes this point of view, then the "appropriate" choice for d is the integer that satisfies Eq. (6).

C. Taylor series

The difficulty in determining the negative exponents from a time series comes primarily from the fact that the attractor is often very "thin" at many locations in the directions associated with certain negative Lyapunov exponents. This occurs when the contraction rate per (approximate) cycle is very large. In theory one could always examine the data in such a small local neighborhood that this thinness would not be apparent, but in practice this would often require many orders of magnitude more data points than are obtainable. In addition, studying the data on such a fine scale will increase errors due to noise.

On the other hand, consider the behavior of a neighborhood of data points that is large compared to the thickness of the attractor yet small compared to the size of the whole attractor. In general, the data points will lie close to some curved subsurface within the local neighborhood, as is illustrated in Fig. 2. This curvature can cause severe problems for a linear mapping, as it attempts to represent correctly the true local mapping.

One solution to these problems is to go to a nonlinear mapping such as the Taylor-series expansion we use in this paper. The idea is to increase the order of the expansion to the point where a curved surface of that order can follow the curvature of the local data points quite closely. The tradeoff in this method is that the number of terms in a multidimensional Taylor expansion increases quite rapidly with the order. Therefore, given a fixed amount of data, the size of the local neighborhood required to unambiguously perform the fitting must increase. Because of this, there may be an optimal order of expansion beyond which the results begin to deteriorate.

One way to determine the appropriate order is simply to look at how accurately the calculated map fits the data. If the rms error in the fitting decreases as the order increases, then the increase in order was an improvement, while if the error increases it is a sign that going to higher order is not improving the result. Another attractive method for deciding how many terms in the Taylor expansion should be used is the minimum description length principle which evaluates the minimum code length required to capture both the description of the data and the cost of the parameters.[23]

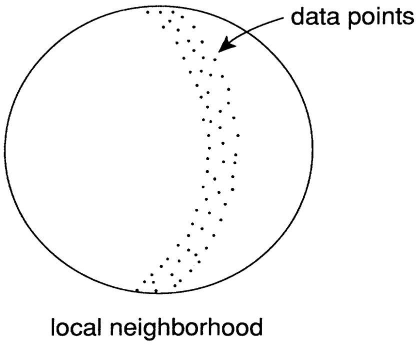

FIG. 2. Local neighborhood in 2D for a data set that is nearly singular in one direction. Illustrates data "curvature" which can be the source of severe errors in the calculation of Jacobian matrices when a strictly linear analysis is undertaken.

In general, it is often a good idea to keep track of any large errors that may occur at only a few points in the calculation. Such behavior is an indication that there may be "bad points" in particular neighborhoods of the attractor. We can loosely define bad points as those that are outside statistical norms and exercise large influence in the local fits to the data. Examples are outliers and points of high leverage. One can try to eliminate these by changing the global dimension, or by reducing the order of the expansion so that the neighborhoods become smaller, or by increasing the number of data points used in the analysis. Finally, if all else fails, it may be statistically acceptable to eliminate these points by hand.[24] For an excellent discussion of the issue of determining and eliminating bad points, we recommend Ref. 23.

D. Choice of the delay time T

In principle, any choice of the delay time T is acceptable in the limit of an infinite amount of data. In the more likely event of a finite amount of data, the choice of T is of *considerable* practical importance in trying to reconstruct the attractor that represents the dynamical system that generated the data. One method used by previous researchers is to choose T to be some fraction of the autocorrelation time of the $x(n)$'s.[9,10,25] This choice of delay time was also chosen as the evolution time T_2.[9,10] As we will discuss later, this choice of $T_2=T$ meant that the linear map of near neighbors at time n to near neighbors at time $n+T_2$ ($\tau=1$) has a particularly simple form. (For reasons we discuss in Sec. II this simple form is not a large advantage. Thus we do not recommend always setting $T=T_2$.) For some of the results we present, we used an information theoretic method developed by Fraser and Swinney based on work by Shaw and others.[26,27] The method yields the best time-lag representation of the attractor on which the vectors $\mathbf{y}(n)$ evolve. As a brief illustration of our use of their results, we digress to a case where the evolution vectors are two dimensional. In this case all the evolution vectors are of the form

$$\mathbf{y}(n)=(x(n),x(n+T)) .$$

We begin by recalling the definition of average mutual information and reviewing the meaning of this measurement on the data set. The average mutual information between two sets of measurements A and B—for us the measurements of $x(n)$ and the measurements of $x(n+T)$—is defined as follows. Let A be the ensemble of values $x(n)$ that are the first components of the evolution vectors $\mathbf{y}(n)$. Thus

$$A=\{x(n):n=1,\dots,N\} .$$

We will let a denote an arbitrary element of A. Similarly, let B be the ensemble of values that are the second components of the evolution vectors $\mathbf{y}(n)$. Thus,

$$B=\{x(n+T):n=1,\dots,N\} .$$

We will let b denote an arbitrary element of B.

Ensembles are sets associated with a probability distribution. Let $P_A(a)$ denote the probability of choosing a when making a selection from set A [the $x(n)$'s]. Likewise, $P_B(b)$ is the probability of choosing b when making a selection from set B [the $x(n+T)$'s]. The distributions P_A and P_B are completely determined by the time series $x(n)$ and $x(n+T)$.

Finally, we turn to the set of time-delayed evolution vectors $\mathbf{y}(n){:}n=1,\dots,N$. Imagine choosing a vector $\mathbf{Y}$ from the set of evolution vectors $\mathbf{y}(n){:}n=1,\dots,N$. The joint probability distribution $P_{A,B}(a,b)$ is the probability of getting a as the first component *and* b as the second component of $\mathbf{Y}$. [We emphasize that the joint probability distribution is a statement about the likelihood of a vector (a,b) appearing in our set of evolution vectors.] In the presence of noise [here represented by our having only p significant digits in each measurement of the $x(n)$] a particular numerical value may correspond to many different $x(n)$'s. A similar statement can be made concerning the $x(n+T)$'s.

The method evaluates the average mutual information $I(T)$ between the measurements of elements of the ensembles A and B as a function of the lag T. This average mutual information is defined by

$$I(T)=\sum_{\substack{a\in A\\ b\in B}} P_{A,B}(a,b)\log_2\left[\frac{P_{A,B}(a,b)}{P_A(a)P_B(b)}\right] . \tag{8}$$

It is a quantitative measure of the amount (in bits) one learns about the measurements B from measurements A. In other words, $I(T)$ is a measurement of how much one knows about the numerical value of the second component of an evolution vector $\mathbf{y}$ when one knows the numerical value of the first component. $I(T)$ is symmetric in A and B, and it is positive semidefinite, vanishing only when the ensembles A and B are independent. (In that case, $P_{A,B}$ factorizes into the product of P_A and P_B.)

We use the average mutual information as a *prescription* that makes T not too small and not too large. [If T is too small, then $x(n)$ and $x(n+T)$ would be basically the same measurement. If T is too large, then $x(n)$ and $x(n+T)$ are random with respect to each other.] We take T to be the value that yields the first local minimum of $I(T)$. By choosing T in this manner, we insure that the second components of the evolution vectors contain as much new information as possible about the attractor when compared to the first component.[26]

For some of the results in Sec. III we have performed the calculation indicated by Eq. (8) and determined T in the manner stated. Since T must be a multiple of the observation sampling time τ, we choose T to be the multiple of τ that corresponds to the first local minimum of $I(T)$. We then used this value of T for the time delay between each component of the vectors $\mathbf{y}$. We realize that this choice is not fully justified when $d>2$. To be absolutely correct, we should evaluate the average mutual information between d components of the vectors $\mathbf{y}$. This process has been described by Fraser.[26] The amount of data required to do this correctly can be prohibitively large (10^6 or more). Although we could generate that much data using numerical integration, we feel that it would *not* be representative of *typical* experimental situation. Thus, such a procedure would work against our desire to remain as faithful as possible to the analysis of experimental systems.

E. Generation of higher-order mappings

We now turn to the procedure we use to map small displacements around our orbit $\mathbf{y}(n), \mathbf{z}^r(n;0)$, into small displacements near the next time step $\mathbf{z}^r(n;T_2)$. We begin by considering the map which evolves $\mathbf{y}$ forward by time T_2,

$$\mathbf{y}(n+T_2)=\mathbf{F}(\mathbf{y}(n)) \ .$$

If $\mathbf{y}^r(n;0)$ is the rth nearest neighbor to $\mathbf{y}(n)$, then in our time step, T_2, the vector distance from $\mathbf{y}^r(n;0)$ to $\mathbf{y}(n)$ becomes

$$\begin{aligned}\mathbf{z}^r(n;T_2)&=\mathbf{y}^r(n;T_2)-\mathbf{y}(n+T_2)\\ &=\mathbf{F}(\mathbf{y}(n)+\mathbf{z}^r(n;0))-\mathbf{F}(\mathbf{y}(n)) \ ,\end{aligned}$$

where we have used the definition

$$\mathbf{z}^r(n;0)=\mathbf{y}^r(n;0)-\mathbf{y}(n) \ .$$

Let $z^r_\alpha(n;k)$ be the αth component of $\mathbf{z}^r(n;k)$. Expanding $\mathbf{F}$ in a Taylor series about the fiducial orbit $\mathbf{y}(n)$, we find

$$\begin{aligned}z^r_\alpha(n;T_2)=&\ DF_{\alpha\beta}(n)z^r_\beta(n;0)\\ &+DF^{(2)}{}_{\alpha\beta\gamma}(n)z^r_\beta(n;0)z^r_\gamma(n;0)+\cdots \ ,\end{aligned} \qquad (9)$$

where

$$DF_{\alpha\beta\gamma}(n)=\partial F_\alpha/\partial y_\beta \ ,$$

$$DF^{(2)}{}_{\alpha\beta\gamma}(n)=(1/2!)\partial^2 F_\alpha/\partial y_\beta \partial y_\gamma \ ,$$

etc., and all terms are evaluated along the orbit $\mathbf{y}(n)$. (Repeated indices are summed over.)

The Jacobian of the underlying dynamics is the first term on the right-hand side of Eq. (9), $DF_{\alpha\beta}(n)$. Earlier workers on this subject truncated Eq. (9) at this linear term.[9,13,12] They then sought to determine numerically the best $d\times d$ matrix, $DF_{\alpha\beta}(n)$, that satisfies

$$z^r_\alpha(n;T_2)=DF_{\alpha\beta}(n)z^r_\beta(n;0) \ .$$

This approach places a twofold burden on the Jacobian $\underline{D}\mathbf{F}$. On the one hand, it must yield the correct Lyapunov exponents (cf. Sec. I) while, on the other hand, it must map $\mathbf{z}^r(n;0)$ into $\mathbf{z}^r(n;T_2)$. It is this latter burden we will remove from $\underline{D}\mathbf{F}$, asking only that it yield the correct Lyapunov exponents. The accurate determination of the local Jacobians is especially important for our study since inaccuracies in them are magnified by the ill-conditioned nature of the final matrix $\underline{D}\mathbf{F}^K$ whose eigenvalues we seek.

A chaotic attractor is defined as having at least one positive Lyapunov exponent and one negative Lyapunov exponent. The Lyapunov exponents come from the eigenvalues of $\underline{D}\mathbf{F}$ iterated along the fiducial orbit. In Sec. I we noted that this Jacobian product will be ill conditioned. For ill-conditioned matrices small changes in the elements can produce large changes in the spectrum of eigenvalues.[24] By truncating Eq. (9) at the first term, one may introduce small changes in the elements of $DF_{\alpha\beta}(n)$ that would not appear were one to retain the higher-order terms. For example, suppose one were trying to fit a mapping of $\mathbf{z}^r(n;0)$ into $\mathbf{z}^r(n;T_2)$ using only $\underline{D}\mathbf{F}$ when in fact $\mathbf{F}$ is cubic in $\mathbf{y}$. Such an $\mathbf{F}$ would indicate that Eq. (9) has nonzero terms $\underline{D}\mathbf{F}^{(2)}$. In an effort to compensate for the missing terms, a truncated Eq. (9) would force $\underline{D}\mathbf{F}$ to change some of its values. This may lead to large changes in the calculated Lyapunov exponents.

This discussion provides the basic motivation for the work reported in this paper. We have examined the effects of retaining terms up to fifth order in $\mathbf{z}^r(n;0)$. We will determine the parameters $\underline{D}\mathbf{F}$, $\underline{D}\mathbf{F}^{(2)}$, etc., via a least-squares fit. It can be shown that the minimum number of parameters N_P needed for Taylor series of order N_{Tay} is given by

$$N_P=\left[\prod_{k=1}^{N_{\text{Tay}}}\frac{d+k}{k}\right]-1 \ , \qquad (10)$$

which grows rather rapidly with N_{Tay} and d. N_P is also the minimum number of neighbors required to calculate values for the fitting parameters in the expansion. (Using less than N_P neighbors would result in an underdetermined least-squares fit.) We use at least twice this number of neighbors in our least-squares fit of the residuals in Eq. (9).

We introduce our procedure by using the EKRC (Ref. 9) method as an illustration. They truncated Eq. (9) at first order,

$$z^r_\alpha(n;T_2)=DF_{\alpha\beta}(n)z^r_\beta(n;0) \ . \qquad (11)$$

Examining individual components of the $\mathbf{z}$'s, we find that they can be written as

$$z^r_\alpha(n;0)=x(n_r+(\alpha-1)T)-x(n+(\alpha-1)T)$$

and

$$\begin{aligned}z^r_\alpha(n;T_2)=&\ x(n_r+T_2+(\alpha-1)T)\\ &-x(n+T_2+(\alpha-1)T) \ .\end{aligned}$$

In this notation n_r is the n value associated with the rth nearest neighbor to $\mathbf{y}(n)$. For the special case $T=\tau=T_2$, we note that $z^r_\alpha(n;T_2)=z^r_{\alpha+1}(n;0)$ for $\alpha=1,2,\ldots,d-1$. This permits us to write the Jacobian in the form

$$\underline{D}\mathbf{F}=\begin{bmatrix}0 & 1 & 0 & 0 & \cdots & 0\\ 0 & 0 & 1 & 0 & \cdots & 0\\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1\\ DF_{d1} & DF_{d2} & DF_{d3} & DF_{d4} & \cdots & DF_{dd}\end{bmatrix} \ , \qquad (12)$$

where $DF_{d1},\ldots,DF_{dd}$ are to be determined.

We determine $DF_{d1},\ldots,DF_{dd}$ (the dth row of $\underline{D}\mathbf{F}$) by a least-squares fit. The equation to be fitted is Eq. (11) which for this special case reduces to

$$z^r_d(n;T_2)=DF_{d\beta}z^r_\beta(n) \ .$$

Since there are d unknown elements of the matrix $\underline{D}\mathbf{F}$, we need at least d nearest neighbors for the fitting. If we use less than d, the problem will be underdetermined. Let N_b be the number of neighbors actually used in the least-squares fitting procedure.

We now define a total of d $(\alpha=1,\ldots,d)$, N_b-dimensional vectors $\mathbf{V}^\alpha$ by $V^\alpha_r=z^r_\alpha(n;T_2)$ for $r=1,\ldots,N_b$. the least-squares fitting of Eq. (11) can be written as

$$\begin{bmatrix} V^\alpha_1 \\ V^\alpha_2 \\ \vdots \\ V^\alpha_{N_b} \end{bmatrix} = \begin{bmatrix} z^1_1(n;0) & z^1_2(n;0) & \cdots & z^1_d(n;0) \\ z^2_1(n;0) & z^2_2(n;0) & \cdots & z^2_d(n;0) \\ \vdots & & & \\ z^{N_b}_1(n;0) & z^{N_b}_2(n;0) & \cdots & z^{N_b}_d(n;0) \end{bmatrix} \times \begin{bmatrix} DF_{\alpha 1} \\ DF_{\alpha 2} \\ \vdots \\ DF_{\alpha d} \end{bmatrix} . \qquad (13)$$

For the special case $T=\tau=T_2$, one only need consider $\alpha=d$; however, for the general case we would need to solve Eq. (13) for $\alpha=1,\ldots,d$.

In keeping with the standard notation in the literature we will let $\underline{X}$ denote the $N_b\times d$ matrix connecting $\mathbf{V}$ to $\underline{D}\mathbf{F}$. the solution to the least-squares problem reduces to inverting this matrix $\underline{X}$. The advantage of writing the problem in this manner appears when $T\neq\tau$. Under these conditions every row of $\underline{D}\mathbf{F}$ in Eq. (11) is undetermined. Without the special form of $\underline{D}\mathbf{F}$ given by Eq. (12), we must solve Eq. (13) a total of d times, once for each value of $\alpha=1,\ldots,d$. The vector $\mathbf{V}^\alpha$ changes for each row of $\underline{D}\mathbf{F}$ but the matrix $\underline{X}$ remains unchanged. The computationally intensive part of the least-squares fit is the inversion of $\underline{X}$ which is only performed once. Hence the extra work involved in the least-squares fit for $T\neq T_2$ is negligible. For this reason we do not recommend using $T=T_2$ for the general case.

For higher-order Taylor-series representations of the local-neighborhood-to-local-neighborhood map, the procedure is similar. As an example, consider the case of truncating Eq. (9) at second order. The resulting equation is

$$z^r_\alpha(n;T_2)=DF_{\alpha\beta}z^r_\beta(n;0)+DF^{(2)}_{\alpha\beta\gamma}z^r_\beta(n;0)z^r_\gamma(n;0) \ .$$

Defining $\mathbf{V}^\alpha$, $\mathbf{B}^\alpha$, and $\underline{X}$ as

$$\mathbf{V}^\alpha=\begin{bmatrix} z^1_\alpha(n;T_2) \\ z^2_\alpha(n;T_2) \\ \vdots \\ z^{N_b}_\alpha(n;T_2) \end{bmatrix} ,$$

$$\mathbf{B}^\alpha=\begin{bmatrix} DF_{\alpha 1}(n) \\ \vdots \\ DF_{\alpha d}(n) \\ DF^{(2)}_{\alpha 11}(n) \\ DF^{(2)}_{\alpha 12}(n) \\ \vdots \\ DF^{(2)}_{\alpha dd}(n) \end{bmatrix} , \qquad (14)$$

$$\underline{X}=\begin{bmatrix} z^1_1(n;0) & \cdots & z^1_d(n;0) & z^1_1(n;0)z^1_1(n;0) & z^1_1(n;0)z^1_2(n;0) & \cdots & z^1_d(n;0)z^1_d(n;0) \\ z^2_1(n;0) & \cdots & z^2_d(n;0) & z^2_1(n;0)z^2_1(n;0) & z^2_1(n;0)z^2_2(n;0) & \cdots & z^2_d(n;0)z^2_d(n;0) \\ & \vdots & & & \vdots & & \\ z^{N_b}_1(n;0) & \cdots & z^{N_b}_d(n;0) & z^{N_b}_1(n;0)z^{N_b}_1(n;0) & z^{N_b}_1(n;0)z^{N_b}_2(n;0) & \cdots & z^{N_b}_d(n;0)z^{N_b}_d(n;0) \end{bmatrix} ,$$

then once again the least-squares fitting problem reduces to solving $\mathbf{V}^\alpha=\underline{X}\mathbf{B}^\alpha$ for known vectors $\mathbf{V}^\alpha$ and fixed matrix $\underline{X}$. By taking advantage of the fact that

$$z^r_\alpha(n;0)z^r_\beta(n;0)=z^r_\beta(n;0)z^r_\alpha(n;0) \ ,$$

we have avoided some double counting in our formulation of $\underline{X}$.

To calculate the Lyapunov exponents from the $\underline{D}\mathbf{F}$'s we use the QR decomposition technique discussed by EKRC.[9] The method recursively defines an orthogonal matrix $\underline{Q}(k)$ and an upper triangular matrix $\underline{R}(k)$, $k=1,\ldots,K$, via

$$\underline{D}\mathbf{F}(m+1)\cdot\underline{Q}(m)=\underline{Q}(m+1)\cdot\underline{R}(m+1) \ ,$$

where $\underline{Q}(0)=\underline{I}$ is the $d\times d$ identity matrix. The Lyapunov exponents are given by

$$T_2\lambda_i=\lim_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\ln R_{ii}(k),\quad i=1,2,\ldots,d \tag{15}$$

with T_2 taking its original dimensional value.

A couple of refinements to Eq. (15) are worth noting. First, there will be an initial transient regime that can be omitted from the sum by going from $k=k_0$ to $k=K+k_0$ where k_0 might typically be chosen to be about 100. Second, with the addition of each successive value of $\ln R_{ii}$ the value of the calculated exponents will fluctuate somewhat. This "end effect" can be greatly reduced by windowing the $\ln R_{ii}$ values by multiplying them by a function that goes smoothly to zero at the end points. The Hanning window

$$\{1-\cos[2\pi(k-k_0)/K]\}$$

seems to work quite well for this purpose.

F. Spurious exponents and data singularity

By increasing d, the dimension of the evolution vectors used in the calculation, one increases the number of exponents generated. If the original system producing the data has a phase space of dimension d_{orig} and $d>d_{\text{orig}}$, then at least $d-d_{\text{orig}}$ of these exponents must be numerical artifacts. There are two competing mechanisms by which the numerical values of the spurious exponents may be generated.

First, as noted by EKRC,[9] nonlinear terms in the dynamical system that originally generated the data (or terms of higher order than the polynomial being used to fit the data) cause the data to have more curvature than can be represented by the fitted map. If the noise in the data is sufficiently low, this extra curvature will be noticed. The fitted mapping will adjust its terms in an effort to compensate for this curvature. For highly accurate data, this tends to produce Lyapunov exponents whose numerical values are even larger than the largest true exponent of the system (cf. Sec. III).

As an example of the second mechanism, consider data from the three-dimensional Lorenz system with $d=6$-dimensional reconstructed evolution vectors. The system has three spurious directions. Noise will cause the data to appear to have local displacements into the spurious directions. This typically results in the generation of negative parasitic exponents with finite numerical value. In general, both mechanisms can be at work simultaneously.

It may be possible to choose an optimal value of d using the information theoretic methods of Abarbanel and Kadtke[28] and avoid the worry expressed in this subsection. However, as shown in Ref. 28, the sampling rate may not always allow this luxury. In that case, and perhaps in general as a double check on one's work, it is important to take additional steps to determine which of the exponents are likely to be spurious and what degree of confidence we should have in the remaining values calculated. The simplest and probably best thing to do is to examine the data set for singularities (thin directions). In linear approximation, this is optimally accomplished using the singular value decomposition (SVD) of the local sample covariance matrix of the displacements of all of the neighbors of a given point $\mathbf{y}(n)$ on the attractor.[29] This matrix is given in terms of our $\mathbf{z}^r$,

$$\underline{R}=\frac{1}{N_b}\sum_{r=1}^{N_b}\mathbf{z}^r(n)\mathbf{z}^r(n)^\dagger\ .$$

The square root of the smallest singular value gives the rms displacement of the data set in the thinnest direction. Projecting out this direction, the data can be examined in a $d-1$-dimensional subspace. Then, the next smallest singular value corresponds to the thickness in this subspace, etc. This procedure is not very satisfactory for the type of data set we expect to encounter. The curvature of the data set may artificially induce a thickness that is

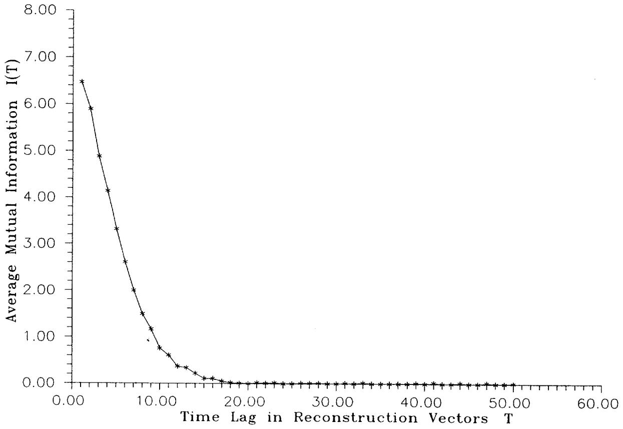

FIG. 3. The average mutual information $I(T)$ for the Hénon map. It is a characteristic of data that are inherently a map that the average mutual information has no minimum.

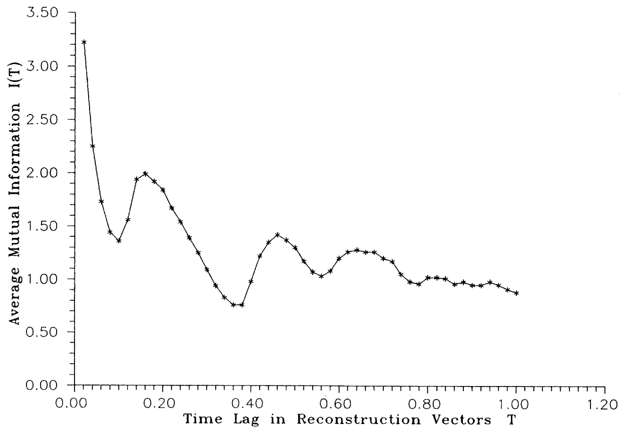

FIG. 4. The average mutual information $I(T)$ for the Lorenz system; the sampling time is $\tau=0.02$. We choose the first minimum at $T\approx0.1$ as the time lag for the phase-space reconstruction (Ref. 26).

TABLE I. The Lyapunov exponents for the Hénon map. These were computed in a local dimension of 2 from data with four digits of accuracy. In this table we show the effects of varying the order of the local polynomial fit to the neighborhood-to-neighborhood map. Global dimension $d_G=2$. Correct values are $\lambda_1=0.408$, $\lambda_2=-1.62$.

Order of polynomial	λ_1	λ_2
Linear	0.434 51	-1.5849
Quadratic	0.447 07	-1.5096
Cubic	0.446 85	-1.5486
Quartic	0.451 42	-1.4679

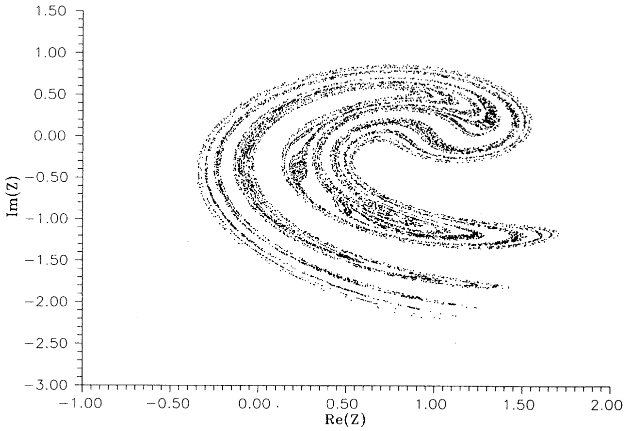

FIG. 5. The phase portrait for the Ikeda map.

significantly greater than the deviation of the data from a properly chosen curved surface.

There are a variety of ways of attempting to refine this singularity measurement. One method that works well in most cases is this: First, obtain an estimate of the most singular direction using SVD as previously described. Next, the coordinates are rotated so that this singular direction corresponds to one of the axes. Now express the small displacements along the singular direction as a function of the other coordinates, such as a Taylor expansion up to a certain order. The rms error in the fit is then an estimate of the data thickness and the linear part of the expansion coefficients is a correction to the singular direction. This procedure may be iterated to attempt to improve the fit and is usually convergent except when the data is highly singular in multiple directions. After finding the most singular direction, one can go on to analyze the remaining coordinates and identify the next most singular direction and so on, until a complete set of singular values and orthogonal singular directions has been identified.

In addition to obtaining the Lyapunov exponents, one can also obtain the direction vectors $\mathbf{L}_i$ associated with these exponents. These directions will be different at every location on the attractor but cannot be calculated locally as they are associated with the Lyapunov exponents which are a global property of the attractor. The $\mathbf{L}_i$ are defined by the requirement that a small displacement along any one of these directions followed forward in time will expand or contract on average at the rate given by the corresponding exponent and if followed backward in time will do so by the inverse rate. To calculate these vectors, we start by examining the "$\underline{Q}$ matrix" obtained at each step by the method of EKRC.[9] At each step the Jacobian matrix $\underline{D}\mathbf{F}$ is multiplied by the previous $\underline{Q}$ matrix and the product $\underline{D}\mathbf{F}\cdot\underline{Q}$ is then decomposed into a new orthogonal $\underline{Q}$ matrix and an upper (or right) triangular matrix $\underline{R}$,

$$\underline{D}\mathbf{F}\cdot\underline{Q}_{\text{old}}=\underline{Q}_{\text{new}}\cdot\underline{R}\ .$$

One then finds that the product of all the $\underline{D}\mathbf{F}$ matrices is given by the last $\underline{Q}$ matrix times the product of all of the $\underline{R}$ matrices $\underline{R}^F$, which is also upper triangular. To find the direction $\mathbf{L}_1$ associated with the largest exponent, we note that except for certain very special directions, an arbitrary initial displacement vector is expected upon repeated iteration to align itself with the desired direction. We can choose our arbitrary vector to be $(1,0,0,0,\dots)$ since there is no reason to assume that this direction is in any way special. Then we find that this vector evolves to $\underline{R}^F_{11}\underline{Q}_1$, where $\mathbf{Q}_1$ is the first column vector of the final $\underline{Q}$ matrix. So $\mathbf{Q}_1=\mathbf{L}_1$. (Note: the reader should not be worried that this result is not so obvious for other choices of the initial vector. They also work out because $\underline{R}^F$ is highly singular; it is "top heavy," i.e., the elements of the top row are much larger than the second row, etc.) Now consider an arbitrary initial plane which we could choose to be all points of the form $(a,b,0,0,0,\dots)$. We expect this to align itself, upon repeated iteration, with the plane defined by the first two Lyapunov direction vectors. Since $\underline{R}^F$ is upper triangular, we find that the resultant points all lie in the plane defined by $\mathbf{Q}_1$ and $\mathbf{Q}_2$. Thus we know that $\mathbf{L}_2$ lies in this plane. All of the $\mathbf{Q}$ vectors are

TABLE II. The Lyapunov exponents for the Hénon map. These were computed using a local cubic neighborhood-to-neighborhood map using six digits of accuracy in data. The effect of varying the dimension $d=d_L=d_G$ is shown. Correct values are $\lambda_1=0.408$, $\lambda_2=-1.62$.

d_G	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
2	0.444 90	$-1.609\,1$				
3	0.440 73	$-0.893\,24$	-1.6535			
4	0.441 99	$-0.307\,00$	$-0.803\,62$	-1.6246		
5	0.463 12	$-0.049\,209$	$-0.389\,60$	$-0.760\,73$	-1.6352	
6	0.464 82	0.143 95	$-0.227\,19$	$-0.424\,57$	$-0.871\,54$	-1.6449

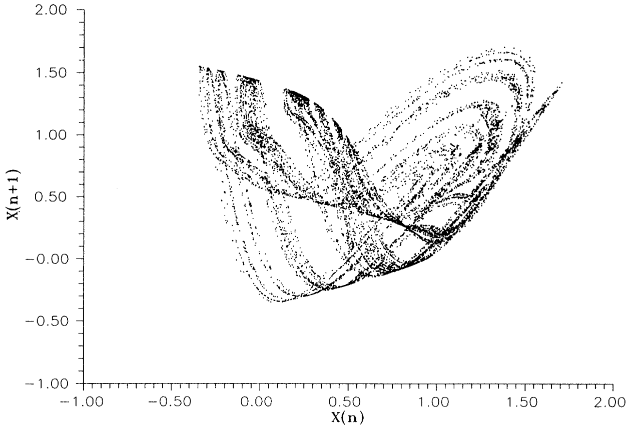

FIG. 6. An attempt to reconstruct the phase portrait of the Ikeda map in $d_G=2$. The lack of a one-to-one projection of the attractor onto the $[x(n),x(n+1)]$ plane is clear to the eye. This and the correlation function data below indicate that one should use global dimension $d_G>2$.

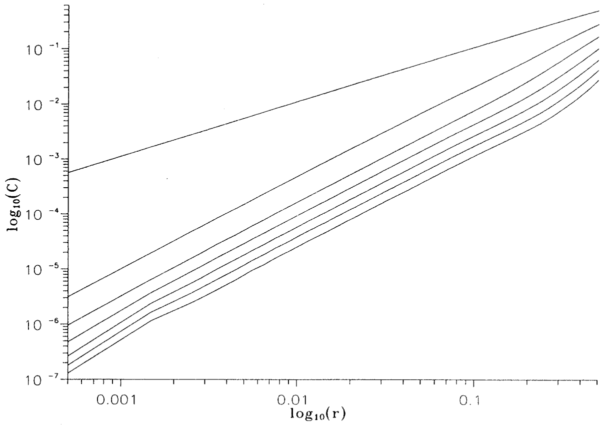

FIG. 7. The Grassberger-Procaccia correlation function for the Ikeda map for global dimensions $d_G=1,2,\ldots,7$. From the slope we obtain $d_a\approx 1.8$. It seems clear that the correlation function changes slightly at the low and high ends until dimension 3 or 4, indicating that $d_G=2$ is not a good global dimension.

orthogonal, while the **L** vectors are not. Similarly, it is found that $\mathbf{L}_3$ lies somewhere in the $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$, three-space, etc.

After startup transients have died away, the $\underline{Q}$ matrix depends only on the present location on the attractor and not on exactly how many previous calculation steps were used to get there. This independence is basically a result of the multiplicative ergodic theorem of Oseledec.[2] Also this explains why it is advantageous to initialize this matrix by taking a significant number of initial steps before taking contributions from the $\underline{R}$ matrices towards evaluation of the Lyapunov exponents.

It is useful to examine the "thickness" of the data set in the $\mathbf{Q}_i$ directions. This can be done by trying to fit the points to a subspace of dimension $d-1$, curved to the desired order, which is tangent at the origin to all but one of the **Q** vectors. This is a straightforward least-squares problem and the rms error of fit gives a measure of the thickness. If one direction is known to be significantly more singular than all others, then this will usually be nearly collinear with the last **Q** vector. If not, this is possibly a sign that this direction is too singular and the calculation is generating a spurious exponent.

To obtain the actual $\mathbf{L}_i$ at a given location on the attractor, it is necessary to have information about the dynamics a long time into the future as well as the past. For example, the last **L** vector is that one special direction that continues to evolve at the rate governed by the most negative exponent as it is followed into the future. We can obtain enough information to calculate all of the **L** vectors by studying the reverse as well as the forward dynamics of the system. To do this, we simply invert the $\underline{D}\mathbf{F}$ matrices (do not recalculate them from the reversed data set in case of errors or singularities involved in their calculation) and use these to obtain the reverse $\underline{Q}$ matrices $\underline{Q}_R$. We must be far enough from both ends of the calculation so that transients have died out in both $\underline{Q}$ and $\underline{Q}_R$. Then $\mathbf{L}_i$ will be known to be orthogonal to the last $d-i$ column vectors of $\underline{Q}$ and also to the last $i-1$ column vectors of $\underline{Q}_R$. This is sufficient information to determine $\mathbf{L}_i$ in nondegenerate cases. One method is to make a matrix consisting of all of the orthogonal vectors mentioned above plus one zero vector. This is then analyzed by singular value decomposition to obtain the desired vector.

Once the $\mathbf{L}_i$ have been determined, there are several things that can be done with them. First, they should be examined to see if two or more of them are nearly collinear. This can occur if a poor choice was made for the delay time (probably too small) or also if nonlinear effects are generating a large spurious exponent. Note that a more general analysis can be made by a SVD of the matrix of **L** vectors. We can also determine the data thickness in the $\mathbf{L}_i$ directions in the same way that we did previously for the $\mathbf{Q}_i$. This is particularly useful for identifying a spurious positive exponent. We can also determine

TABLE III. The Lyapunov exponents for the Ikeda map. In the first part of the table these were calculated using $d_L=2$ and $d_G=2$, showing results that depart strongly from the true values $\lambda_1\approx 0.503$ and $\lambda_2\approx -0.719$. Increasing d_G to 4 in the second part of the table, we eliminate the self-interactions of the attractor and now obtain acceptably accurate results for the two exponents. Correct values are $\lambda_1\approx 0.503$, $\lambda_2\approx -0.719$.

	$d_L=2$; $d_G=2$		$d_L=2$, $d_G=4$	
Order of fit	λ_1	λ_2	λ_1	λ_2
Linear	0.6104	−0.2358	0.5085	−0.7317
Quadratic	0.5348	−0.3779	0.5050	−0.7281
Cubic	0.5649	−0.4260	0.5123	−0.7356
Quartic	0.5912	−0.4881	0.5056	−0.7354
Quintic	0.6047	−0.4588	0.5115	−0.7490

TABLE IV. The Lyapunov exponents for the Ikeda map. These were computed with $d=d_L=d_G=4$, with data that has five digits of accuracy. In this table we vary the order of the local polynomial fit of the neighborhood-to-neighborhood map and show its effect on the two legitimate and two spurious exponents. We notice that the spurious positive exponent decays as the order of the polynomial fit is increased. For fourth-order fits the spurious exponents appear as λ_2 and λ_3. Correct values are $\lambda_1=0.503$, $\lambda_2=-0.719$.

Order of polynomial	λ_1	λ_2	λ_3	λ_4
Linear	0.879 59	0.469 44	−0.680 81	−1.1596
Quadratic	0.495 44	0.019 259	−0.611 09	−0.809 87
Cubic	0.495 97	−0.194 15	−0.652624	−0.769 63
Quartic	0.503 54	−0.154 18	−0.627 12	−0.788 07

the data thickness in the $\mathbf{L}_i$ directions in the same fashion as was done for the $\mathbf{Q}_i$. The results of this will be reported in Sec. III.

The validity of the calculated $\mathbf{L}_i$ vectors can be checked by using them to recalculate the Lyapunov exponents. These are now completely separated and can be obtained by observing the local stretching or contraction of the mapping along each of the $\mathbf{L}_i$ directions and averaging the logarithms of these over a large number of steps. This works quite well, but does not appear to give any improvement over the original calculation of the exponents.

Once it is clear that a given direction in the data set is nearly singular, there are several approaches that can be taken to deal with the situation:

(1) Carry through the calculation and attempt to identify the spurious exponent at the conclusion of the calculation.

(2) Reduce the value of d.

(3) Attempt to "null out" the singular direction by making all of the row vectors of the $\underline{D}\mathbf{F}$ matrices orthogonal to the local singular direction(s). This forces the spurious exponent(s) to go to minus infinity.

(4) One can project data onto a lower-dimensional space (of the correct dimension) in such a way as to optimize display of the nonsingular directions.

III. RESULTS OF NUMERICAL EXPERIMENTS

A. Time lags and Lyapunov exponents

In this section we present the results of our numerical experiments on the dynamical systems enumerated in Sec. I. For each of the systems we numerically generated a time series as follows:

The Hénon and Ikeda systems are maps of the plane into itself. We chose initial conditions (0.25,0.25) for the Hénon map and (0.1,0.1) for the Ikeda map. We then iterated these initial conditions forward in time using double precision. In each case we discarded the first 50 iterates of the map as representing transients before recording the data. Finally, we recorded 11 000 x_1 coordinates for the Hénon map and 21 000 values of the real part of z for the Ikeda map.

The Lorenz system is composed of three ordinary differential equations. It was numerically integrated forward in time by using fourth-order Runge-Kutta integration and double precision. We used two values for the sampling rate τ: one was 0.02, while the other was 0.05. For the sampling rate 0.02 the initial conditions were (17.83,12.34,10.32). After a few hundred steps to avoid transients, we recorded 60 000 and 20 000 values of x_1, respectively, for the two data sets.

The Mackey-Glass equation is a delay differential equa-

TABLE V. The Lyapunov exponents for the Ikeda map. The embedding and local dimension are the same as in the previous table; however, we have retained eight digits of accuracy. Notice that the negative Lyapunov exponent is determined to greater accuracy than the previous case. But the spurious positive exponent does not decay as fast as in the previous case. Correct values are $\lambda_1=0.503$, $\lambda_2=-0.719$

Order of polynomial	λ_1	λ_2	λ_3	λ_4
Linear	0.921 79	0.471 99	−0.672 51	−1.1499
Quadratic	1.5205	0.487 59	−0.587 14	−0.801 99
Cubic	0.648 34	0.424 11	−0.579 30	−0.739 51
Quartic	0.611 81	0.308 23	−0.601 21	−0.722 02

TABLE VI. In this table we display the Lyapunov exponents for the Lorenz system computed from 50 000 data points evaluated with a sampling time $\tau = T_2 = 0.02$ and a time delay $T = 5\tau = 0.1$. The data have five digits of accuracy and are analyzed using $d = d_L = d_G = 3$ for varying orders of the polynomial fit to the neighborhood-to-neighborhood map. Correct values are $\lambda_1 = 1.51$, $\lambda_2 = 0.0$, $\lambda_3 = -22.5$.

Order of polynomial	λ_1	λ_2	λ_3
Linear	1.4504	−0.005 712 3	−13.999
Quadratic	1.5027	−0.046 041	−19.448
Cubic	1.5121	0.006 964 1	−22.925
Quartic	1.5561	0.032 219	−23.465

tion having an effectively infinite-dimensional phase space. For a delay time $s = 17$, we used an integration time step of 0.1 so that we were always saving the previous 170 values of x. The output was sampled at time intervals $\tau = 5$, i.e., 50 numerical time steps. We used 20 000 values of x in our calculations. Runge-Kutta integration methods require evaluation of the derivative at half time steps, and this is complicated in this case because we need the value of the delayed variable which we have calculated only for integral time steps. Instead, we chose to perform the integration using a fourth-order predictor-corrector algorithm, which appears to give very accurate results. The accuracy can be tested observing the time evolution of the system for a set time interval (several oscillatory cycles long) using a relatively large time step. This is then repeated for successively smaller time steps, each time starting from the same initial condition. We generally set all of the saved values of x to 0.5 as our initial condition.

By using long time series, we avoid the possible effects of a finite sample size in our investigations. We have no precise criteria for determining the exact size of the data set needed for a particular dynamical system. As a rule of thumb, one would expect that the required number of points N_D should increase with the dimension of the attractor d_a as $\log_{10}(N_D) \propto d_a$.[30] We wanted to use data sets that are large enough to cover the attractor and thus avoid problems due to finite sample size. Yet at the same time we wanted to restrict N_D to sizes comparable to a large range of experimental situations. We think that our choices satisfy these two opposing goals, but it is well worth further study of the effects of the size of data sets on the evaluation of Lyapunov exponents.

TABLE VII. In this table we display the Lyapunov exponents for the Lorenz system computed from 20 000 data points evaluated with a sampling time $\tau = 0.05$ and a time delay $T = T_2 = 2\tau = 0.1$. The data have nine plus digits of accuracy and are analyzed using $d = d_L = 3$ and $d_G = 7$ for mapping orders 1 through 5.

Order of polynomial	λ_1	λ_2	λ_3
Linear	1.549	−0.094 70	−14.31
Quadratic	1.519	−0.026 47	−20.26
Cubic	1.505	−0.005 695	−22.59
Quartic	1.502	−0.002 847	−22.63
Quintic	1.502	−0.000 387	−22.40

As stated in Sec. II, the first part of the analysis of the time series is to determine the time lag T for the reconstruction. To assist in making this determination, we calculated the average mutual information function, Eq. (8), for the first three systems on our list. In this paper we report the result of this for the first and third systems. We are fortunate to have been provided with public domain C programs written by A. M. Fraser that accomplish this task. For the Hénon map we used an ensemble size of $2^{13} = 8192$. For the Ikeda map we used $2^{14} = 16\,384$ points, and for the Lorentz equation we used $2^{15} = 32\,768$ points. The results of our investigations on $I(T)$ are shown in Figs. 3 and 4.

Figure 3 shows the type of behavior common to data that are associated with maps. Each $*$ indicates a choice of T and the corresponding value of $I(T)$. The figure shows a monotonic decrease in $I(T)$ with T. Systems *known* to be flows and whose data yield this type of mutual information behavior can be explained by analogy with a Poincaré surface of section. Each point of the surface of section would represent an iterate of the map. Thus, by considering points on such a surface of section, one can model a flow via a map. Under these conditions each iterate of the map represents a large temporal evolution of the dynamics.[31] For data that are inherently best modeled by a map, we used $T = \tau = T_2 = 1$.

Figure 4 shows the type of behavior common to data that are associated with flows. Again, each $*$ indicates a choice of T and the corresponding value of $I(T)$. The individual choices for T are of the form $T = m\tau$ for increasing m. This is necessary since the only measurements available for the ensemble are the ones taken with a fixed sampling rate τ. The mutual information $I(T)$ demonstrates an oscillatory behavior overlaying a smooth monotonic decay for increasing T. For data that are inherently best modeled by flows, we have chosen T to be the first local minimum of I. For the Lorenz system this is at $T = 0.1$.

For all of our test cases we examined least-squares fits to the data up to and including fourth or fifth order. Our own experience as well as that of others[9] indicates that one should use at least twice the minimum number of nearest neighbors ($N_b = 2N_P$) in a least-squares fit. We used a *k-d* tree[32] to minimize the time required to find nearest neighbors. For our purpose we used the Euclidean norm, although other norms are possible. The *k-d* tree method of finding nearest neighbors requires $[O \sim \log_{10}(N)]$ time to compute.

As we stated above (cf. Sec. II), the least-squares fitting reduces to solve $\mathbf{V}^\alpha = \underline{X}\mathbf{B}^\alpha$ for known $\mathbf{V}^\alpha$ and $\underline{X}$. The inversion of $\underline{X}$ was accomplished using the QR algorithm without pivoting recommended in the LINPACK *Users' Guide*.[33] This process is by far the most time-consuming part of the calculation. The matrix $\underline{X}$ is $N_b \times N_b/2$. Even for moderate d, third- and fourth-order fitting produces

TABLE VIII. The Lyapunov exponents for the Lorenz system. These were computed using a local cubic neighborhood-to-neighborhood map using six digits of accuracy in the data. For all cases the local and global dimensions were equal ($d=d_L=d_G$). The data have sampling time $T_2=\tau=0.02$ and time lag $T=0.1$; 50 000 points were used in the calculations. Correct values are $\lambda_1=1.51$, $\lambda_2=0.0$, $\lambda_3=-22.5$.

d_G	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
3	1.5169	−0.007 994	−23.093			
4	1.5375	−0.070 352	−22.147	−108.21		
5	1.5631	−0.015 967	−21.600	−77.403	−114.83	
6	1.6123	0.009 517 1	−21.340	−60.057	−80.231	−115.50

large $\underline{X}$ matrices. At present we can think of no way to avoid this problem. Finally, the matrix inversion problem is sensitive to roundoff errors. We believe that this is due to the ill-conditioned matrices that one is inverting. To avoid as much of this problem as possible, we ran our Lyapunov exponent programs in double precision on a 32-bit machine. Thus, although the data inserted into the programs contained only p digits of accuracy ($p \leq 9$), the matrix calculations carried a higher accuracy.

The reader will notice that Eq. (15) is defined only in the limit as $K \rightarrow \infty$. Obviously we can only choose some value of K that we trust is large enough to yield a consistent result. We based our choice on the following criteria: Since we know the true evolution equations, we can calculate the exact Jacobian at each time step. We perform a QR decomposition on the Jacobian and determine what value of K yields consistent results. We settled on $K=1000$, 3000, and 5000 for the first three systems introduced in Sec. I. These values are much larger than necessary. Part of our rationale in their section was to avoid problems associated with small K. Of course, an experimentalist would not be able to do this. A common method for determining K for experimental data is to calculate the power spectrum and find the largest relevant frequency. This yields an extremely rough estimate of a dominant time scale in the problem. Not surprisingly, we recommend performing the calculation for values of K that are several hundred or several thousand times the time associated with this line. Then repeat the calculation using $2K$ Jacobians and look for convergence of the exponents calculated at K and $2K$.

TABLE IX. The Lyapunov exponents for the Mackey-Glass equation obtained from 20 000 data points. We used $d_L=3$ and $d_G=7$ to minimize self-intersection of the attractor. The order of the local mappings was increased from 1 to 4. $T_2=T=2\tau=10$. Correct values are $\lambda_1 \approx 0.006$, $\lambda_2 \approx 0.0$, $\lambda_3 \approx -0.04$. $x[y]$ represents $x \times 10^{-y}$.

Order of fit	λ_1	λ_2	λ_3
1	3.44[3]	−2.20[3]	−5.92[2]
2	6.41[3]	−7.05[4]	−5.07[2]
3	6.38[3]	−1.64[3]	−2.21[2]
4	6.72[3]	−7.59[4]	−4.28[2]

When performing our numerical experiments, we used two approaches to calculating Lyapunov exponents. The two approaches are intimately related to the local and global dimensions d_L and d_G, defined in Sec. II.

Under one approach the evolution vectors $\mathbf{y}(n)$ are of dimension $d=d_L<d_G$. The reconstruction is thus not globally diffeomorphic to the attractor that represents the true dynamics. However, we believe that the $\mathbf{y}$'s do capture the local dynamics of the attractor. We generate these evolution vectors by first embedding (via time delays) the data into d_G dimensions for the purpose of calculating nearest neighbors. Having found the nearest neighbors to a particular $\mathbf{y}(n)$ in the d_G-dimensional Euclidean space, we project down onto a d_L-dimensional subspace by eliminating all components of the $\mathbf{y}$'s beyond the first d_L. All subsequent calculations are performed in $d=d_L$ dimensions and d_L Lyapunov exponents are produced. This method is useful if, for example, one has *a priori* knowledge of the dynamics that generated the original data set or an unambiguous value for the dimension of the attractor. (See our discussion on local and global dimensions in Sec. II.)

Under the other approach the evolution vectors $\mathbf{y}(n)$ are of dimension $d=d_G$. Therefore, the $\mathbf{y}$'s evolve on an attractor that is diffeomorphic to the attractor that represents the true dynamics. Nearest neighbors to a particular $\mathbf{y}(n)$ are found in a d_G-dimensional Euclidean space and all calculations are performed in that space. The method produces $d=d_G$ Lyapunov exponents. This approach is useful when subsequent calculations involving the attractor require a global embedding—for example, if one wishes to perform constrainted predictions of temporal evolution on the attractor (cf. Sec. IV and Ref. 10). It is also an appropriate choice in the absence of *a prori* knowledge concerning underlying dynamical equations.

We have results on the Lyapunov exponent for Eqs. (1)–(4) as well as results on the attractor thicknesses for Eqs. (3) and (4). First, we discuss results for the Hénon map. For this example we did not investigate the difference between local and global dimensions. All of our calculations were performed in $\mathbb{R}^d$ where $d=d_G$. The Hénon map is naturally two dimensional. Indeed, since $x_2(n)$ is just proportional to $x_1(n-1)$, the two vectors

$$\mathbf{y}(n)=[x_1(n),x_1(n+1)]$$

TABLE X. Singular thickness values for the Lorenz system. The attractor was reconstructed in a Euclidean space of dimension $d_G=5$, and 250 neighbors for each point on the fiducial orbit were used. S_1 through S_5 are the most singular through least singular thickness values. The values must stabilize above the intrinsic noise level (here very small) in order to correspond to nonspurious exponents. Note that S_1 and S_2 continue to drop as the order is increased, indicating that a good choice for d_L would be 3 (since three values stabilized). $x\,[y]$ represents $x\times10^{-y}$.

Order of polynomial	S_1	S_2	S_3	S_4	S_5
Linear	2.13[2]	5.26[2]	3.62[1]	1.95	2.73
Quadratic	1.05[3]	9.17[3]	1.88[2]	1.95	2.73
Cubic	1.98[5]	7.73[4]	1.84[2]	1.97	2.69
Quartic	6.64[8]	1.08[4]	1.73[2]	1.93	2.71
Quintic	2.78[11]	2.78[6]	1.71[2]	1.92	2.72

are just a rotation of the original phase space. The accepted values of the Lyapunov exponents for this system are $\lambda_1=0.418$ and $\lambda_2=-1.62$. Table I shows the calculation of the Lyapunov exponents in $d=d_G=2$ for linear through quartic local polynomials for the neighborhood-to-neighborhood mapping; each entry has $p=4$ digits of accuracy in the data. For all of our cases we control the number of digits by rounding off the data after p significant digits. This is a *very* crude approximation of noise but it is convenient for our purposes. We also examined the effect of keeping more significant digits. Unlike subsequent cases, we found no significant differences in the values of the calculated Lyapunov exponents for $4<p\leq9$. Table II shows a different cut on the data. Here we see the Lyapunov exponents resulting from a cubic local map using $p=6$ for dimensions $d=d_G=1,\ldots,6$. As indicated above, the result of having too large a local dimension (for this experiment $d_L=d_G$) is to produce spurious negative exponents.

The Ikeda map is an excellent example of a system for which the use of separate local and global dimensions is important. Comparison of the attractor in the original phase space (Fig. 5) with the two-dimensional time-delay reconstruction (Fig. 6) clearly shows the self-intersection effect which was discussed previously. Since knowledge of the fractal dimension is useful for deciding on the values of d_L and d_G, we plot the Grassberger-Procaccia[20] correlation integral for embedding dimensions $d_G=1$ through 7 as seen in Fig. 7. From the slope we see that the fractal dimension is $d_a\approx1.8$. The fact that the slope of the linear portions of the curves continues to change slightly past a global dimension of $d_G=2$ is a result of the self-intersection of the attractor. On this basis, and also from Eq. (5), we would choose d_G to be at least 3 and preferably 4. An appropriate value for the local dimension is $d_L=2$, which agrees with the suggested value of Eq. (6) and also can be obtained by our method of data singularity analysis. (Calculation of the fractal dimension d_a and/or the embedding dimension d_G of an attractor using this method requires large data sets and has slow convergence. We have confidence in the value d_a stated because it is relatively far away from 2.0 as well as the fact that we have some *a priori* knowledge that the answer must be less than 2. In practice this method of determining the local dimension d_L may not yield unambiguous answers.) In Table III we show the results obtained when using the incorrect values $d=d_G=d_L=2$ and compare with those obtained with $d_G=4$ and $d=d_L=2$. The correct values are approximately $\lambda_1=0.503$ and $\lambda_2=-0.719$.

If both the local and global dimensions are set equal to 4 ($d=d_L=d_G=4$), the overlap problem is eliminated but we have to deal with two spurious exponents as shown in Table IV. We notice that for the linear polynomial fits a spurious positive exponent appears. However, as the order of the polynomial fit increases past 2, this spurious positive exponent disappears. Also, the value generated for the negative Lyapunov exponent is more accurately

TABLE XI. Singular thickness values for the Mackey-Glass equations, for 20 000 data points using $d_L=3$, $d_G=7$, and $T_2=T=2\tau=10$. 250 neighbors for each point on the fiducial orbit were used. Note that S_1 and S_2 continue to drop as the order is increased, indicating that a good choice for d_L would be 3 (since three values stabilized). $x\,[y]$ represents $x\times10^{-y}$.

Order of fit	S_1	S_2	S_3	S_4	S_5
Linear	1.80[3]	3.47[3]	2.06[2]	3.52[2]	5.60[2]
Quadratic	2.45[4]	1.10[3]	1.33[2]	3.45[2]	5.58[2]
Cubic	2.60[5]	3.07[4]	1.09[2]	3.32[2]	5.57[2]
Quartic	2.02[6]	1.12[4]	9.03[3]	3.28[2]	5.59[2]
Quintic	3.96[7]	3.57[5]	6.10[3]	3.50[2]	5.32[2]

TABLE XII. Lyapunov exponents and thicknesses of the attractor along the corresponding Lyapunov direction vectors for the Lorenz system. The calculation was done with 20 000 data points using $d_L=4$ and $d_G=7$ so that there is one spurious exponent. For second and above, the spurious exponent separates from the true ones and can be identified by its extremely small thickness value.

Order of polynomial fit; Lyapunov exponents and thicknesses				
1	$\lambda_1=1.936$	$\lambda_2=0.8019$	$\lambda_3=-1.1137$	$\lambda_3=-13.44$
	$\theta_1=0.4666$	$\theta_2=1.083$	$\theta_3=1.111$	$\theta_4=0.161$
2	$\lambda_1=4.364$	$\lambda_2=1.401$	$\lambda_3=-0.6559$	$\lambda_4=-20.57$
	$\theta_1=0.00256$	$\theta_2=0.4116$	$\theta_3=0.4539$	$\theta_4=0.001\,16$
3	$\lambda_1=18.04$	$\lambda_2=1.502$	$\lambda_3=-0.000\,55$	$\lambda_4=-22.77$
	$\theta_1=1.89[7]$	$\theta_2=0.1288$	$\theta_3=0.0910$	$\theta_4=2.7[4]$
4	$\lambda_1=26.96$	$\lambda_2=1.503$	$\lambda_3=-0.004\,84$	$\lambda_4=-22.55$
	$\theta_1=2.46[10]$	$\theta_2=0.0656$	$\theta_3=0.0656$	$\theta_4=5.57[5]$

reproduced. The advantage of higher-order polynomial fits is clearly demonstrated. For this table the number of significant digits is $p=5$. In Table V we see that the value calculated for the negative Lyapunov exponent moves closer to the correct value as the accuracy of the data increases. In this table the number of significant digits is $p=8$. This improvement is offset by the fact that the spurious positive exponent decays much more slowly than in Table IV. We believe that the value calculated for the positive exponent will eventually decay to the correct value as one increases the order of the polynomial fit beyond fourth order. The result would be two spurious negative exponents (cf. Sec. II). Recall that negative exponents are necessary to contract the d_G-dimensional phase space onto the $d_a=1.8$-dimensional attractor. The Ikeda map is a nice test of the effects of higher polynomials fits since the exponential in Eq. (2) indicates that any fit of finite order will not be able to completely capture the dynamics.

From the results so far the reader may not be very impressed as to the value of going to higher-order fittings, as the linear results do not look too bad for the two maps we have studied. As we will see, the results for flows can be much more impressive. We move now to the Lorenz system, for which the accepted values of the Lyapunov exponents are $\lambda_1=1.50$, $\lambda_2=0.0$, and $\lambda_3=-22.5$. In the case of data from the Lorenz equations we have two slightly different settings for the evolution time lag T_2 and dimensions. One of our choices uses an evolution time lag of $T_2=0.10$ while our other choice is 0.02. As a last issue before presenting our results, we remind the reader that data for the Lorenz equations can be globally embedded in $d_G=3$ dimensions, and the local dimension of the Lorenz attractor is $d_L=3$.

In Table VI we have calculated the Lyapunov exponents for data with $\tau=T_2=0.02$ and an embedding time, $T=0.10$. A total of 50 000 data vectors, **y** were used in the calculation, and we retained $p=5$ digits of accuracy. The order of the polynomial fits ranged from 1 to 4, and we use dimensions $d=d_L=d_G=3$. In Table VII we have calculated the Lyapunov exponents for data with $\tau=0.05$ and $T=T_2=0.10$. 20 000 points were used in the data set and the order of the polynomial ranges from 1 to 5. The data in Table VII is accurate to at least nine digits. Our choices for dimensions differed from those used in Table VI. Here we used a local dimension of $d=d_L=3$ and a global dimension of $d_G=7$. As the reader will observe, from Tables VI and VII, the negative exponent is very difficult to obtain. Yet, we see it dramatically snapping into place as we increase the order of polynomial fit to 3 and above. Also, note the improvement in accuracy of the zero exponent as the accuracy of the data is increased. As one can see, accuracy of the data is very important (unlike the Hénon map). Finally, in Table VIII we have the same Lorenz data as in Table VI but now show the λ's for dimensions $d=d_L=d_G=3,\ldots,6$ for a cubic polynomial fit and $p=6$ digits of accuracy. In this case the spurious exponents all are highly negative as shown. The large negative exponents will cause the d_G-dimensional phase space to collapse rapidly onto the attractor. It is our conjecture that the theoretically correct values of these spurious exponents are minus infinity. The presence of noise (although small) gives width to an attractor that is actually singular in the spurious $d-3$ directions. This small width results in finite values for the spurious Lyapunov exponents.

We show results for the Mackey-Glass equation in Table IX. For the parameters used the dimension of the attractor is known to be $d_a\approx 2.15$. We chose to calculate three Lyapunov exponents in this case, both because this is the choice indicated by Eq. (6) and because of the singularity results described in Sec. III B. Thus $d=d_L=3$. The sufficient condition implied by Eq. (5) suggested that we set the global embedding dimension to $d_G=7$. There are, of course, an infinite number of Lyapunov exponents for this system, but we only have enough information to accurately calculate the most significant three of these. The accepted values for the three Lyapunov exponents are $\lambda_1=0.006$, $\lambda_2=0.0$, and $\lambda_3=-0.04$. The time delays used to reconstruct the attractor were $T_2=T=2\tau=10$. Our results do not converge as rapidly as they did in the Lorenz case, probably due to the highly convoluted nature of this attractor.[19] However, the values of the calculated exponents are reasonably consistent as we increase the order of our polynomial fits to 3 and above. Furthermore, the results are in good agreement with previously calculated values.

B. Attractor thickness and spurious exponents

As discussed earlier, one method of deciding on the value for the local dimension d is by a singularity analysis of the data points in one or more small local neighborhoods. In Table X we show how the measurement of the data thickness improves as the order of the fitting is increased when using data from the Lorenz system. For the Lyapunov data shown, the dimension of the original system is 3 but the scalar data set is being converted to five dimensional vectors. As a result, the two extra dimensions must be highly singular, with data thickness limited by the intrinsic noise level which is very small for this numerically generated data. Thus the first two singular values decrease dramatically as the order is increased. The Lorenz attractor is very thin in places. This is demonstrated by the third singular value, which is a factor of 100 thinner than the final two values and has essentially converged after taking the fitting to second order. Since the third value is significantly higher than the first two, it can be assumed to be well above the intrinsic noise level and thus we would choose $d=3$ as a result of this analysis.

In Table XI we show data from the Mackey-Glass equation. For the parameters used, the attractor has dimension of about 2.15. We have reconstructed the data in a five-dimensional space. Here, none of the singular thickness values drop off as rapidly as the first two of the Lorenz system (even though the dimension of the attractors is about the same). This is due, in large part, to the fact that the system that generated this data has an infinite-dimensional phase space while the Lorenz system has a three-dimensional space. Hence, the Lorenz system is exactly singular in the extra dimensions. Nevertheless, the first two values have dropped off sufficiently to indicate that it would be inadvisable to try and obtain more than three Lyapunov exponents for these parameter settings.

In order to be able to include additional exponents, we need to compare the additional singular thickness values to the level of stochastic noise known to be in the data. If comparable in size, the exponent will be difficult to obtain. We may also compare the rms fitting error of our calculated map to the minimum thickness of the iterated data point set. If it is smaller, than there is a good chance that the calculation will produce a meaningful result.

In Table XII we analyze the Lorenz equations with a local dimension $d=4$, which we know must generate at least one spurious exponent. When using a second-order fitting to our local map, Eq. (9), the results are poor for all of the Lyapunov exponents, and the Lyapunov vectors $\mathbf{L}_2$ and $\mathbf{L}_3$ are found to be nearly collinear. Increasing the polynomial fitting to third order, we find that the last three exponents are very close to the true exponents, while the first is 10 times larger than the true value of the largest exponent. In this case there was no significant collinearity of the $\mathbf{L}_i$. The spurious nature of $\mathbf{L}_1$ can rapidly be identified by examining the local data thickness θ_1 in the $\mathbf{L}_1$ direction, which is over five orders of magnitude smaller than the thickness θ_2 for $\mathbf{L}_2$. A positive exponent should not exhibit any significant "thinnesss," while some thinness for the negative exponents is to be expected as exhibited here by θ_4.

C. Direct effects of noise on the Lyapunov spectrum

Although we have shown that it is possible to include singular directions in the calculation and later identify the questionable positive exponents, the presence of relatively small amounts of noise makes this more difficult. This is illustrated in Figs. 8 and 9 for the Lorenz system. We have added Gaussian white noise to the data points with the indicated standard deviation. In Fig. 8 we have used $d_L=3$, while in Fig. 9 we used $d_L=4$, which will give us one spurious exponent. In both cases we used a third-order expansion for the local mappings. The spurious exponent in Fig. 9 changes wildly as the added noise is increased, going from $+19$ (much higher than the actual largest exponent of 1.5) down to -6. This behavior is in fact another way of identifying a spurious exponent in extremely accurate data. While we can identify the spurious exponent in the $d_L=4$ case, the values of the other three exponents are significantly more noise resistant when the calculation is carried out in $d_L=3$. For this reason, it is prudent to add singular dimensions to the calculation one at a time and observe the effect on the values obtained at the previous step.

The wild behavior of the spurious exponents gives substance to the idea that working with a local dimension larger than needed is a very unwelcome situation in a noisy environment.[28] Since the nondynamical components of the vectors $\mathbf{y}(n)$ are totally dominated by noise, while the dynamical components are just contaminated, we would expect any perceived dynamics in a di-

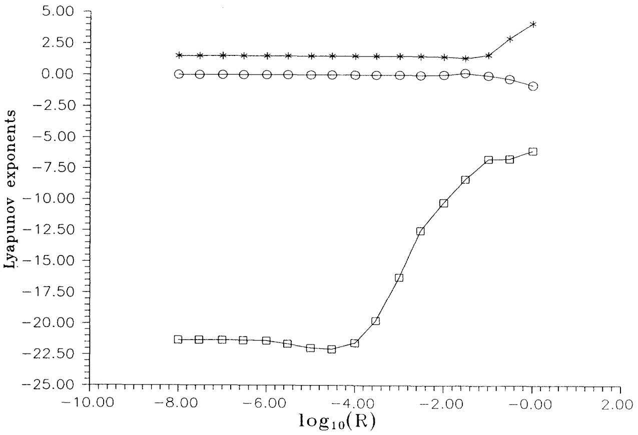

FIG. 8. The effect of external noise on the determination of Lyapunov exponents for the Lorenz system. In this figure the local dimension is $d_L=3$. On the horizontal axis is the $\log_{10}$ of $\mathbb{R}$ which equals the noise level divided by the signal level.

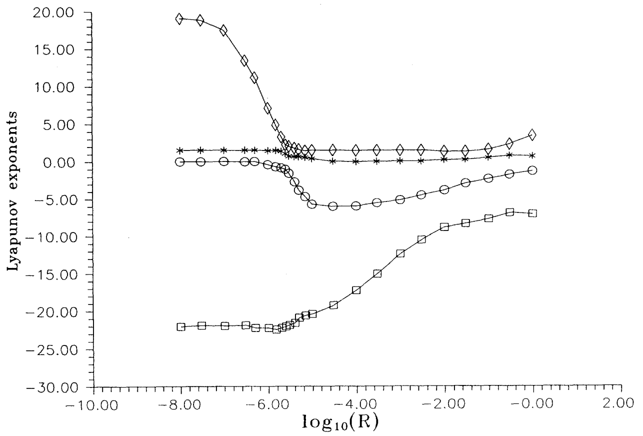

FIG. 9. The effect of external noise on the determination of Lyapunov exponents for the Lorenz system. In this figure the local dimension is $d_L = 4$. The spurious exponent wanders from about +19 to nearly −6 as the noise level is varied. Note that the exponents do not cross search other but prefer to switch roles as they become close. In the $d_L = 3$ case the correct exponents are more robust against the addition of noise. On the horizontal axis is the $\log_{10}$ of $\mathbb{R}$, which equals the noise level divided by the signal level.

mension above the minimum allowed to be completely unreliable.

Noise will also affect the data singularity measurements. As one would expect, the data thickness measures will not dip below the noise floor. Generally, if the dimension is increased sufficiently, the most singular directions will have thicknesses governed by the intrinsic noise level. Singular thickness values that tend to converge to a level that is above the noise level should correspond to meaningful exponents in the analysis. For meaningful results one must therefore reduce the value d until the most singular thickness level remaining is significantly higher than this baseline.

IV. SUMMARY AND CONCLUSIONS

In this paper we have addressed several questions related to the determination of the full spectrum of Lyapunov exponents for time-series data representing observations of a dynamical system:

(i) Can we relieve the Jacobian from the dual task of taking neighborhoods to neighborhoods accurately *and* giving precise values for the Lyapunov exponents by employing higher-order terms in the local map? What additional analytical power comes from using a local polynomial map for the evolution of neighborhoods in reconstructed phase space instead of the local linear maps used in the past?[9,11–13]

(ii) What can we say about the effects of accuracy of the data or, equivalently, noise in the observation on our ability to determine the Lyapunov exponents of the system?

(iii) What methods can we use for determining whether the Lyapunov exponents are actually representative of the dynamics or are artifacts of our having reconstructed the data in too large a space?

(iv) The Ikeda map is a system where the embedding dimension d_G necessary to reconstruct the attractor is larger than the number of original dynamical dimensions d_{orig}. For these situations, what interpretation can be placed on the $d_G - d_{\mathrm{orig}}$ spurious Lyapunov exponents?

Our answers to these questions are contained in the set of tables and the figures in this paper. Basically, we have a positive answer to each of the questions. We have shown the utility of higher-order (rather than linear) local maps from neighborhood to neighborhood in reconstructed phase space. The linear term (the Jacobian) is all that is needed for the determination of the Lyapunov exponents through Oseledec's multiplicative ergodic theorem.[1,2] In our examples, when one includes cubic or higher terms in the local polynomial map, the evaluation of the linear term settles down to a stable and accurate value. This is because the higher-order mapping is able to deal correctly with the local curvature of the data set. The tables make it quite evident that one can, in the low-dimensional cases considered and by extrapolation to few-dimensional attractors with confidence, determine the full set of Lyapunov exponents (or at least all those which are strongly involved in the dynamics within the attractor) from time-series data alone.

The accuracy of the data is also critical! The values of negative exponents in "thin" directions of the data set start to become affected when the noise level goes above about 10% of the thickness of the data set in the associated Lyapunov direction. We have demonstrated this in two ways: (1) We kept data accurate only to 1, 2, or up to 9 significant figures and (2) we explicitly added Gaussian white noise to the clean data.

On the third issue we have presented two slightly different ways to test the reality of the Lyapunov exponents determined from the observed data. We have shown here in the case of both the Lorenz system and the Mackey-Glass equation that evaluating the "thickness" of the singular directions in the local space provides strong evidence of spurious versus realistic exponents. When the exponent is spurious and positive, we should see a very thin extent of the attractor in the eigendirection associated with the exponent. A thick extent of the attractor is associated with a realistic positive exponent as significant stretching of orbits occurs in these directions. Our examples show clearly that one can, indeed, distinguish the two cases.

There are situations, such as prediction,[10] where one is required to work in a reconstructed phase space of dimension $d_G > d_{\mathrm{orig}}$. In these situations there will be $d_G - d_{\mathrm{orig}}$ spurious exponents. For the cases shown in the figures and tables these spurious exponents are usually negative. (The few examples of a positive exponent we attribute to using a polynomial fit that is too low in order to capture the curvature evident in the extremely accurate data we have used.) We interpret them as being necessary to contract a d_G-dimensional phase space onto an attractor whose dimension d_a is less than d_G. We anticipate that for situations such as this it will be necessary to use all d_G Lyapunov exponents when performing calculations in the d_G-dimensional reconstructed phase space.

At this stage the procedures require several cross checks, as none alone is sure to be unambiguous. Nonetheless, the tools are present for extracting these exponents from data. The Lyapunov exponents are invariants characterizing and classifying the attractor. Furthermore, they have clear physical interest in terms of predictability of the system observed through its measured time series. Therefore, our technique provides an important item in the analysis of signals from chaotic systems. As indicated elsewhere,[10] the constraints of Lyapunov exponents are required to assure oneself that when making predictions in chaos, the predictions are being made on the system observed. The more Lyapunov exponents one can reliably and believably determine, the more sensible the predictions one can achieve.

There are two directions that have not been explored by us in this paper which we would recommend to the reader. The first is the investigation of the ability to use these methods in the presence of short data sets. We have used only long data sets to establish the possibility of computing the local Jacobian matrices accurately. If one can also establish the dependence of the procedure on the length of the data set, that would be of some practical interest. The second is to extend the methods we have established to rational local maps of neighborhoods to each other. These are likely to have a better radius of convergence and may be useful in the investigation of the issue just mentioned about short data sets. From a computational viewpoint, rational maps are only as difficult as polynomial or Taylor expansions, being another form of least-squares problem. The main mathematical issue is the reliability of an extrapolation of the map to $\mathbf{z}=0$ where the evaluation of the required Jacobian matrix is to be made—i.e., on the fiducial orbit. Use of rational rather than polynomial maps for this would seen to have attractive virtues.

There is an interesting question that arose in the course of this work concerning the negative exponents. In some sense the negative exponents govern how orbits in the full d_G-dimensional phase space are drawn to the attractor. Our data, since we have passed all transients and deal only with stationary time series, lie entirely on the attractor. How, then, can we determine the negative exponents from data alone? The answer we have arrived at, with no pretense at rigor, in the face of the clear demonstration that one can indeed determine the negative as well as the positive exponents in many cases, rests on our assumed form of the neighborhood-to-neighborhood maps as polynomials. This implicitly assumes an analyticity of the maps in phase space which extends off the attractor, thus allowing us to learn something of phase space by continuation away from the given data. This point may merit a deeper mathematical investigation than we are prepared to provide.

Finally, we recall the possibility of using the minimum description length principle[23] for choosing the optimum number of coefficients in a polynomial (or rational) neighborhood map. This may prove quite essential when data sets become sparse.

ACKNOWLEDGMENTS

We are most appreciative for productive conversations with Andy Fraser, Ed Lorenz, and Jorma Rissanen about the material covered in this article. Fraser's average mutual information program was most useful for our work. This work was supported in part under the DARPA–University Research Initiative, URI Contract No. N0014-86-K-0758. Some of the computation reported in this paper was carried out at the NASA Ames Research Center's Numerical Aerodynamic Simulation Program under the auspices of the Joint Program in Nonlinear Science between the University of California and NASA/Ames.

[*] Also at Institute for Nonlinear Science, University of California, San Diego, La Jolla, CA.

[1] J.-P. Eckmann, and D. Ruelle, Rev. Mod. Phys. **57**, 617 (1985).

[2] V. I. Oseledec, Tr. Mosk. Mat. Obsc. Moscow Math. Soc. **19**, 17 (1968).

[3] A. N. Kolmogorov, Dokl. Akad. Nauk SSSR **119**, 861 (1958).

[4] Y. Sinai, Dokl. Akad. Nauk SSSR **124**, 386 (1959).

[5] Y. B. Pesin, Usp. Mat. Nauk **32**, 55 (1977) [Russian Math. Survey **32**, 55 (1977)].

[6] J. M. Greene and J. S. Kim, Physica D **24**, 213 (1987).

[7] F. Takens, in *Dynamical Systems and Turbulence, Warwick, 1980*, Vol. 898 of *Lecture Notes in Mathematics*, edited by R. Rand and L. S. Young (Springer, Berlin, 1981), p. 366.

[8] R. Mañé, in *Dynamical Systems and Turbulence, Warwick, 1980*, Vol. 898 of *Lecture Notes in Mathematics*, edited by D. Rand and L. S. Young (Springer, Berlin, 1981), p. 230.

[9] J.-P. Eckmann, S. O. Kamphorst, D.Ruelle, and S. Ciliberto, Phys. Rev. A **34**, 4971 (1986). We refer to this work as EKRC.

[10] H. D. I. Abarbanel, R. Brown, and J. B. Kadtke, Phys. Lett. **138A**, 401 (1980); Phys. Rev. A **41**, 1782 (1990).

[11] G. Benetinn, L. Galani, A. Giorgilli, and J.-M. Strelcyn, Meccanica **15**, 9 (1980).

[12] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Physica D **16**, 285 (1985).

[13] M. Sano, and Y. Sawada, Phys. Rev. Lett. **53**, 1082 (1985).

[14] M. Hénon, Commun. Math. Phys. **50**, 69 (1976).

[15] K. Ikeda Opt. Commun. **30**, 257 (1979).

[16] C. Grebogi, E. Ott, and J. A. Yorke, Phys. Rev. Lett. **57**, 1284 (1986).

[17] E. N. Lorenz, J. Atmos. Sci. **20**, 130 (1963).

[18] M. C. Mackey and L. Glass, Science **197**, 287 (1977).

[19] J. D. Farmer, Physica D **4**, 366 (1982).

[20] P. Grassberger, and I. Procaccia, Phys. Rev. Lett. **50**, 346 (1983); Physica D **9**, 189 (1983).

[21] J. L. Kaplan, and J. A. Yorke, *Chaotic Behavior of Multidimensional Difference Equations*, Vol. 730 of *Lecture Notes in Mathematics* (Springer, Berlin, 1979), pp. 228–237.

[22] P. Frederickson, J. Kaplan, E. Yorke, and J. Yorke, J. Diff. Eq. **49**, 185 (1983).

[23] J. Rissanen, IEEE Trans. Inf. Theory **IT-30**, 629(1984); *Stochastic Complexity in Statistical Inquiry,* (World Scientific, Singapore, 1989). We are most grateful to Dr. Rissanen for numerous discussions about the connection between information theory and dynamics.

[24] S. Chatterjee and A. S. Hadi, *Sensitivity Analysis in Linear Regression* (Wiley, New York, 1988).

[25] W. L. Ditto, S. Rauseo, R. Cawley, C. Grebogi, G.-H. Hsu, E. Kostelich, E. Ott, H. T. Savage, R. Segnan, M. L. Spano, and J. A. Yorke, Phys. Rev. Lett. **63**, 923 (1989).

[26] A. M. Fraser and H. L. Swinney, Phys. Rev. A **33**, 1134 (1986); A. M. Fraser, IEEE Trans. Inf. Theory **35**, 245 (1989); Physica D **34**, 391 (1989).

[27] R. Shaw, Z. Naturforsch. Teil A **36**, 80 (1981).

[28] H. D. I. Aberbanel and J. B. Kadtke (unpublished).

[29] W. H. Press, B. Flannery, S. Teulkolsky, and W. Vetterling, *Numerical Recipes* (Cambridge University Press, New York, 1986).

[30] Also see J.-P. Eckmann and D. Ruelle (unpublished).

[31] We numerically performed a surface-of-section experiment of the Duffing equation. After performing the mutual information calculation on the surface of section data, we find that the resulting graphs are virtually indistinguishable from Fig. 3.

[32] J. H. Friedman, J. L. Bentley, and R. A. Finkel, ACM Trans. Math. Software **3**, 209 (1977).

[33] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, LINPACK *Users' Guide* (SIAM, Philadelphia, 1979).