Estimation of large-scale dimension densities

Corinna Raab and Jürgen Kurths
Institut für Physik, Universität Potsdam, 14415 Potsdam, Germany
(Received 12 October 2000; published 20 June 2001)

We propose a technique to calculate large-scale dimension densities in both higher-dimensional spatiotemporal systems and low-dimensional systems from only a few data points, where known methods usually have an unsatisfactory scaling behavior. This is mainly due to boundary and finite-size effects. With our rather simple method, we normalize boundary effects and get a significant correction of the dimension estimate. This straightforward approach is based on rather general assumptions. So even weak coherent structures obtained from small spatial couplings can be detected with this method, which is impossible by using the Lyapunov-dimension density. We demonstrate the efficiency of our technique for coupled logistic maps, coupled tent maps, the Lorenz attractor, and the Roessler attractor.

DOI: 10.1103/PhysRevE.64.016216 PACS number(s): 05.45.−a

I. INTRODUCTION

To characterize low-dimensional deterministic systems, there exists a large number of accepted methods [1,2]. Among these, the Grassberger-Procaccia algorithm [3] is a well-known method, but several pitfalls may result from this technique [4]; it fails especially for higher-dimensional systems, because with the growing dimension of the attractor, the number of data points needed for the calculation is increasing exponentially [5,6]. So the Grassberger-Procaccia algorithm can only be seriously used for low-dimensional systems [7]. For spatially extended systems with higher dimensions, Grassberger [8] and Mayer-Kress and Kaneko [9] proposed to calculate dimension densities with the Grassberger-Procaccia algorithm, but a criticism has been that due to systematic errors, the dimension density is underestimated [10,11]. Due to this fact, it is often impossible to distinguish between chaos and noise. Another way of characterizing spatiotemporal chaotic systems is to use the Karhunen-Loeve decomposition [12] or to estimate the spectrum of Lyapunov exponents [13]. The scaling behavior of the fractal dimension was studied analytically and numerically by Politi and Witt [14]. Another method of calculating dimension densities based on random attractors was introduced by Bauer *et al*. [15]. But a general problem of calculating dimensions of high-dimensional systems was shown by Olbrich *et al*. [16]. In weakly coupled map lattices, they observed a sequence of plateaus with increasing values of the dimension for decreasing scales. To get such results, they needed very long time series, which are often not available. In this paper, we present a normalization method that is intended for rather small data sets and covers only the region of large scales. We show that for distributed systems that exhibit some spatial correlations, as, for example, coupled-map lattices, it is possible to calculate a normalized large-scale dimension density, which enables us to compare the spatial coupling strength in different systems.

II. DIMENSION ANALYSIS

First, we consider a spatiotemporal system and use the original local coordinates of the system. Applying the Grassberger-Procaccia algorithm to a few of these coordinates, we obtain results that depend strongly on the chosen coordinates, because spatially separated coordinates usually show fewer correlations than neighboring ones. It is, however, important to note that both possibilities of chosen coordinates lead to the same systematic errors produced by boundary effects. To overcome this difficulty, we propose a special normalization of the neighboring coordinates to the more separated ones. This method leads to a suitable correction of these errors and enables us to estimate dimension densities in the rather large scales of a system from surprisingly few data points. Because of that, it even works for high-dimensional systems and makes it possible to identify different couplings in coupled-map lattices.

The dynamical systems that will be studied here are represented by m-dimensional time series consisting of vectors $\{\vec{x}(t)=(x_1(t),x_2(t),\ldots,x_m(t))\}$. The first step to estimate the dimension of such an attractor with the Grassberger-Procaccia algorithm is to calculate the correlation integral $C(r,m)$ by

$$C(r,m)=\frac{1}{N(N-1)}\sum_{i\neq j}\theta(r-|\vec{x}(t_i)-\vec{x}(t_j)|), \qquad (1)$$

where θ is the Heaviside function. The correlation dimension D_2 is then defined as

$$D_2=\lim_{r\to 0}\lim_{m\to\infty}\frac{d\log C(r,m)}{-d\log(r)} \qquad (2)$$

if this limit exists. Because it is impossible to reach the limit $r\to 0$ in numerical calculations, one has to estimate this dimension from larger distances, i.e., the right-hand side of Eq. (2) becomes a distance-dependent function $D_2(r,m)$. For low-dimensional attractors, there often exists a rather large region in $\log_2(r)$ where this $D_2(r,m)$ is nearly constant, i.e., there is a plateau in the plot of $D_2(r,m)$ against $\log_2(r)$. This part is referred to as the scaling region [3]. An important property of this analysis is that for large distances, compared to the diameter of the attractor, $D_2(r,m)$ decreases because of boundary effects. Here boundary means the border of the

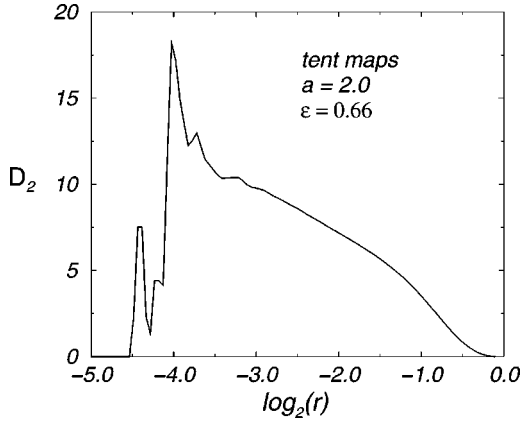

FIG. 1. Dimension of coupled tent maps [Eq. (6)] calculated with the Grassberger-Procaccia algorithm for an embedding dimension of $m=20$ and $N=50\,000$ data points; it does not show any scaling region.

shape of the attractor, outside of which no points of the attractor are located, and not the fractal boundary to which every point of the attractor belongs. These boundary effects occur in the calculation of the correlation integral because usually a point close to the boundary of the attractor has many fewer neighbors within a region of radius r than a point in the center of the attractor. This effect is stronger the larger r is. So for large scales, one counts fewer data points than expected in calculating $C(r,m)$, and because of that $D_2(r,m)$ is decreasing, i.e., we get a systematic underestimate. On the other hand, for small distances the dimension is fluctuating rather irregularly due to the finite amount of data. In this region, the data points are not dense enough to find a statistically relevant amount of neighbors (see Figs. 1 and 2).

A. High-dimensional systems

For higher dimensions, it is impossible to reach the scaling region without an extraordinary amount of data and CPU power. An optimistic estimation of the minimal number of needed data points N is given by [6]

$$2\log(N) > D_2 \log\left(\frac{1}{r}\right), \tag{3}$$

where r is the scaling region in units of the diameter of the attractor. That means that for higher-dimensional systems, N is not only growing because of the growing dimension, but it is also increasing because the scaling region is shifted towards smaller values of r. If there is no exceptionally large data set available, one only gets a region with strongly fluctuating values (finite-size effects) and a region with decreasing values (boundary effects), so that no value for the correlation dimension can be estimated (Fig. 1).

For the following calculations, we use coupled-map lattices (CML's) as a paradigmatic example of spatiotemporal systems [17,18]. They are defined by a nonlinear local mapping,

$$x_i(t+1) = f(\tilde{x}_i(t)) \tag{4}$$

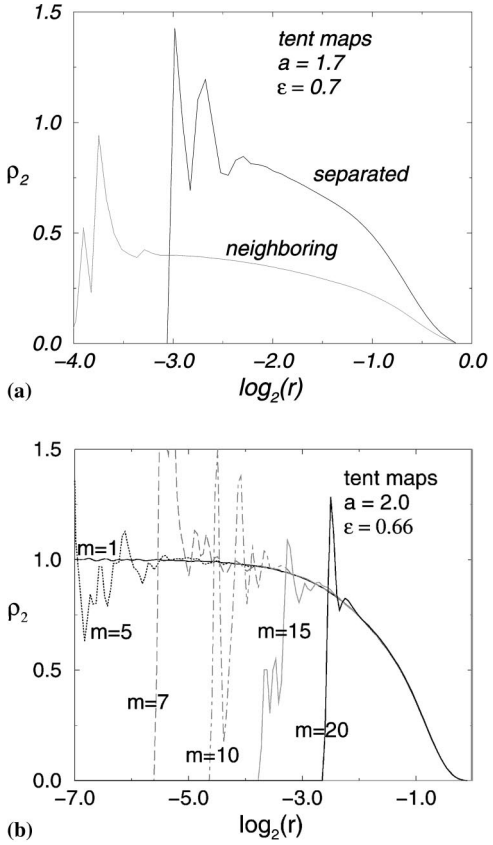

FIG. 2. Left: Difference of the dimension density [Eq. (7)] obtained from separated and neighboring coordinates calculated with the Grassberger-Procaccia algorithm (system as in Fig. 1). Right: The dimension density for separated coordinates is plotted for different numbers m of used coordinates, i.e., for different embedding dimensions. For large scales, the dimension density does not depend on m. With increasing m, only the region with strong fluctuations is shifted to larger scales. $N=50000$ data points.

and a spatial coupling of the nearest neighbors,

$$\tilde{x}_i(t) = F(x_{i-1}(t), x_i(t), x_{i+1}(t), \varepsilon) \tag{5}$$

with $i=1,\ldots,N$ and periodic boundary conditions,

$$x_1(t) = F(x_N(t), x_1(t), x_2(t), \varepsilon),$$

$$x_N(t) = F(x_{N-1}(t), x_N(t), x_1(t), \varepsilon).$$

As functions of local dynamics, we use tent maps $[f_a(x) = a|x-1/2|]$ or logistic maps $[f_a(x) = ax(1-x)]$, which are ordered in a chain consisting of 1000 elements that are diffusively coupled with a coupling strength ε by

$$x_i(t+1) = f_a\{(1-\varepsilon)x_i(t) + (\varepsilon/2)[x_{i-1}(t) + x_{i+1}(t)]\}. \tag{6}$$

These systems are often characterized quantitatively by the density of the Lyapunov dimension, because the equations of motion are known so that one can calculate the Lyapunov exponents [2,19]. To compare our approach with

this Lyapunov dimension density, we also use dimension densities in the following calculations defined by

$$\rho_2(r,m)=D_2(r,m)/m. \tag{7}$$

For this dimension density applies $0\leqslant\rho_2(r,m)\leqslant 1$.

For a calculation of the correlation function (1) from the original coordinates, it is practically impossible to include all the 1000 coordinates, so it is necessary to select only some of them. The result then depends strongly on the chosen coordinates. Directly neighboring coordinates give smaller values of the dimension density $\rho_2(r,m)$ than coordinates that are far enough separated (see Fig. 2). Here ''far enough'' means that the dimension density does not change if the distance of the coordinates is increased. In the special case of coupled maps, this is fulfilled if one takes every tenth coordinate. The values of the dimension density for neighboring coordinates are significantly smaller because they are spatially more correlated than separated coordinates.

The main step of our approach is to use spatially separated coordinates for the normalization. The crucial condition for this is that there is no correlation between the separated coordinates. The calculation of the dimension density for up to 20 separated coordinates of the CML demonstrates that they are indeed uncorrelated because the dimension density for large scales has the same feature for different numbers of coordinates; only the region with fluctuating values is shifted to larger scales for more coordinates (see Fig. 2). However it is important to note that for both sorts of coordinates, i.e, the separated and the neighboring ones, it is impossible to estimate a value of the dimension because there is no scaling region due to the finite amount of data and boundary effects, as mentioned above.

In the following, we propose how to calculate a new normalized large-scale dimension density using the different correlation properties between separated and neighboring coordinates in a CML.

Due to the independence of separated coordinates, the dimension density for small scales reaches $\rho_2(r,m)=D_2^s(r,m)/m=1$. Values of $\rho_2^s(r,m)<1$ calculated with the Grassberger-Procaccia algorithm for large scales are the result of boundary effects if one assumes that the dimension density of separated coordinates is the same for all scales. Because of the correlations between neighboring coordinates, we find smaller values of the dimension density for those coordinates. We use this difference between separated and neighboring coordinates for our method of calculating dimension densities.

So we define the *large-scale dimension density* $\rho_{\mathrm{ls}}(r,m)$ by normalizing the dimension of the neighboring coordinates $D_2^n(r,m)$ to the dimension of the separated coordinates $D_2^s(r,m)$:

$$\rho_{\mathrm{ls}}(r,m)=D_2^n(r,m)/D_2^s(r,m), \tag{8}$$

where *n* denotes neighboring and *s* denotes separated coordinates. An important property of this $\rho_{\mathrm{ls}}(r,m)$ is that it measures spatial dependence in the system, i.e., $\rho_{\mathrm{ls}}(r,m)=1$ if there is no spatial dependence at all, but deviations from 1 indicate the degree of spatial dependence. Quantitative results and their interpretation will be discussed for several examples below.

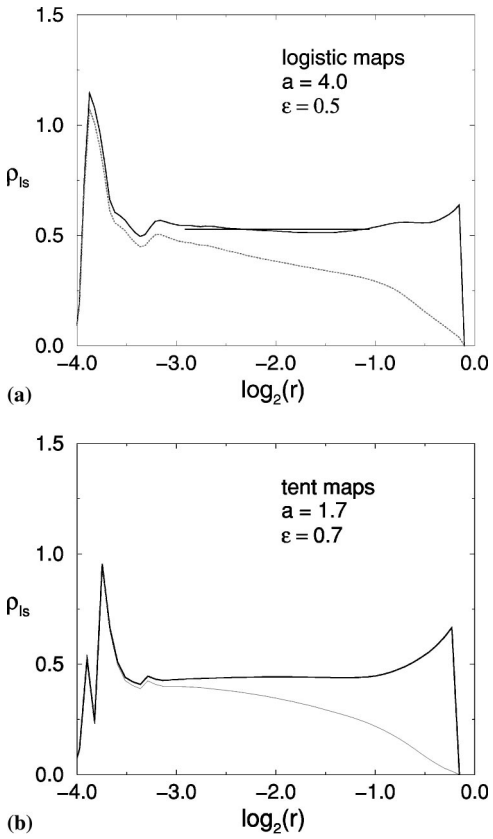

FIG. 3. Large-scale dimension density [Eq. (9)] of coupled maps ($m=20$, $N=50\,000$) compared with that obtained from the original Grassberger-Procaccia algorithm (dotted).

As shown above, the dimension density for separated coordinates does not depend on the number of the used coordinates, and we can calculate it with the one-dimensional curve of the separated coordinates. In this case, it obviously holds that $D_2^s(r,1)=D_2^n(r,1)$ and consequently $D_2^s(r,m)=mD_2^n(r,1)$. Hence, one coordinate out of the neighboring coordinates works as well. So the large-scale dimension density is

$$\rho_{\mathrm{ls}}(r,m)=\frac{D_2^n(r,m)/m}{D_2^n(r,1)}. \tag{9}$$

This means another reduction of the data needed to calculate the dimension density and also enables us to use our method for low-dimensional systems, as shown later.

This simple procedure leads to a surprisingly well-expressed plateau yielding an estimate of ρ_{ls}. The normalized curve is shown in Fig. 3 and compared with the original Grassberger-Procaccia algorithm. The normalized curve has a rather large scaling region, which enables us to estimate a reliable value for the large-scale dimension density of the attractor by averaging all values of this region. For CML's, this means scales of about $\frac{1}{2}$ to $\frac{1}{8}$ of the diameter of the attractor.

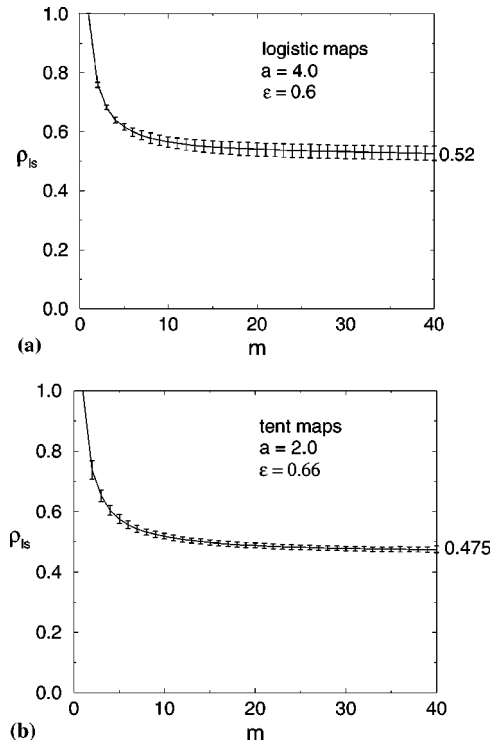

FIG. 4. The values calculated out of the plateau of the normalized curve (see Fig. 3) are plotted for different m. The bars are the standard deviation. The value is decreasing for an increasing number m of used coordinates, but for $m>20$ one gets a quite good upper limit.

These results depend on the number of used coordinates, i.e., on the embedding dimension m. For an increasing m, the value is decreasing as shown in Fig. 4. However, for dimensions $m>20$, the value is not decreasing much, so calculations with $m=20$ yield a quite good upper limit for the large-scale dimension density.

Compared with the Lyapunov dimension density, calculated out of the Lyapunov exponents via the Kaplan-Yorke formula [20], our method almost always gives smaller values. This may result from the large scales we are using for our estimation. It is important to note that for couplings $\varepsilon<0.3$, the Lyapunov dimension density is always 1, so that one cannot distinguish between CML's with small but different coupling. It is an important advantage of our large-scale dimension density that it is sensitive also to small couplings (Fig. 5) that cannot be detected by the Lyapunov dimension density.

B. Low-dimensional systems

As shown above, our simple method works well for high-dimensional systems. But it is also very efficient for low-dimensional systems if a special transformation of the coordinates is done with the data set. For the calculation of the large-scale dimension, we then need only a fraction of the data points needed by the Grassberger-Procaccia algorithm. Here the method works for scales of about $\frac{1}{4}$ to $\frac{1}{20}$ of the diameter of the attractor. In the following, we demonstrate this for the Lorenz and the Roessler system in a chaotic state.

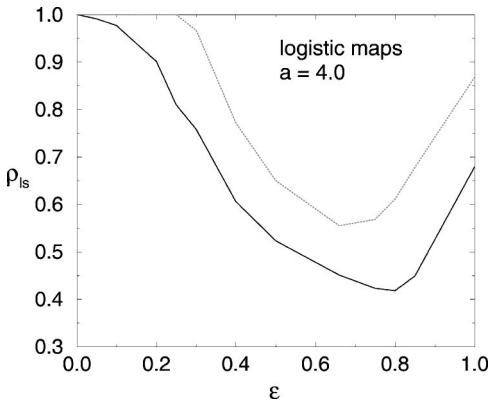

FIG. 5. The large-scale dimension density estimated with $m=20$ is compared to the Lyapunov dimension density (dotted) for a CML of 1000 logistic maps with different coupling constants. The dimension density even detects small couplings between the maps.

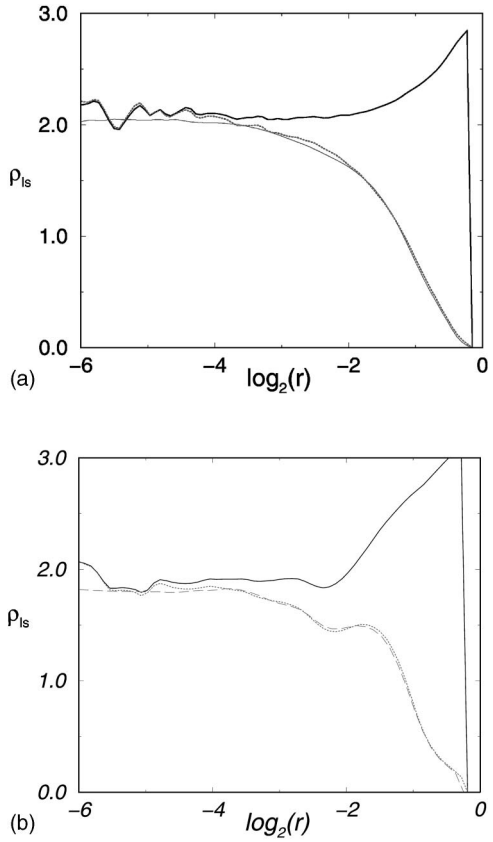

FIG. 6. Here the large-scale dimension $m\rho_{ls}$ (solid line) is compared to the correlation dimension D_2 calculated with the Grassberger-Procaccia algorithm (dotted line) for the (a) Lorenz attractor (1000 data points, $\sigma=10$, $r=28$, and $b=\frac{8}{3}$) and for the (b) Roessler attractor (2000 data points, $a=0.2$, $b=0.2$, and $c=5.7$). The dashed line also shows the curve for D_2 calculated with the Grassberger-Procaccia algorithm, but with 10 000 (Lorenz attractor) and 20 000 (Roessler attractor) data points. The integration step for both systems is $\Delta t=0.01$.

Looking at the data structure of the Lorenz attractor compared to that of the CML's, we see one important difference. For the CML's, every coordinate shows the same distribution of the values, but for the Lorenz attractor, every coordinate has a different distribution. This means that each coordinate generates a different curve $D_2^n(r,1)$ for normalization, which would give three different results. Therefore, we transform every coordinate of the data set to an average distribution of all coordinates. To get this distribution, we sort the values of every coordinate into numerical order and then average over the corresponding values. Due to this ranking, the dimension is invariant [21,22].

The calculation of the large-scale dimension $D_{ls}(r,m) = \rho_{ls}(r,m)*m$ of the Lorenz attractor with 1000 data points then gives $D_{ls}(r,m)=2.07$ with a standard deviation of $\sigma = 0.032$. Figure 6 shows the curves of the large-scale dimension compared with the original Grassberger-Procaccia algorithm calculated with 1000 and 10 000 data points. For 1000 data points, the curve calculated with the Grassberger-Procaccia algorithm does not show a plateau, so that no value for the dimension can be calculated. The value of the large-scale dimension is an average of the curve values between $-4<\log_2(r)<-2$. This is almost the same result one gets for averaging the values of the curve calculated with the Grassberger-Procaccia algorithm using 10 000 data points. Averaging the region in between $-6<\log_2(r)<-4$ gives $D_2(r,m)=2.06$ with $\sigma=0.027$.

The same calculations with the Roessler attractor give us the following results (see Fig. 6): $D_{ls}(r,m)=1.89$, $\sigma=0.03$, calculated with 2000 data points, with an average of $-4.7 <\log_2(r)<-2.7$. A calculation with the Grassberger-Procaccia algorithm, using 20 000 data points, gives us $D_2(r,m)=1.81$, $\sigma=0.02$, with an average of $-6.0<\log_2(r) <-3.5$.

So it is also possible to calculate the large-scale dimension in low-dimensional systems with the normalization method, even if the result might be slightly different from the correlation dimension calculated by the Grassberger-Procaccia algorithm (as for the Roessler attractor). But the advantage of the reduced amount of data more than compensates for this slight difference.

III. CONCLUSIONS

In this paper, we have presented a method to calculate large-scale dimension densities. The correction of the boundary effects leads to a plateau, which makes it possible to estimate the value of the large-scale dimension density. This can be used to compare the strength of correlation in spatially extended systems at different local points. It is not certain that the large-scale dimension density gives the same value that one would obtain by the Grassberger-Procaccia algorithm for small scales because of the effects shown by Olbrich *et al*. [16]. But it is outside of the scope of this paper to calculate exact values of dimensions for all scales. The main intention is to get an instrument that works for small data sets.

In our future work, we will try some applications to observations of complex spatiotemporal behavior with spatial correlations, such as data of climate models, which are much more inhomogeneous than CML's are, so that it is necessary to transform the coordinates as shown for the low-dimensional systems. But the available data sets are also very limited [23].

[1] H.D.I. Abarbanel, *Analysis of Observed Chaotic Data* (Springer, New York, 1996).

[2] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis* (Cambridge University Press, Cambridge, England, 1997).

[3] P. Grassberger and I. Procaccia, Phys. Rev. Lett. **50**, 346 (1983).

[4] P. Grassberger, T. Schreiber, and C. Schaffrath, Int. J. Bifurcation Chaos Appl. Sci. Eng. **1**, 521 (1991).

[5] L.A. Smith, Phys. Lett. A **133**, 283 (1988).

[6] J.-P. Eckmann and D. Ruelle, Physica D **56**, 185 (1992).

[7] A. Jedynak, M. Bach, and J. Timmer, Phys. Rev. E **50**, 1770 (1994).

[8] P. Grassberger, Phys. Scr. **40**, 346 (1989).

[9] G. Mayer-Kress and K. Kaneko, J. Stat. Phys. **54**, 1489 (1989).

[10] A. Torcini, A. Politi, G.P. Puccioni, and G. D'Alessandro, Physica D **53**, 85 (1991).

[11] E. Olbrich, and H. Kantz, Phys. Lett. A **232**, 63 (1997).

[12] S.M. Zoldi, and H.S. Greenside, Phys. Rev. Lett. **78**, 1687 (1997).

[13] M.J. Bünner, and R. Hegger, Phys. Lett. A **258**, 25 (1999).

[14] A. Politi, and A. Witt, Phys. Rev. Lett. **82**, 3034 (1999).

[15] M. Bauer, H. Heng, and W. Martienssen, Phys. Rev. Lett. **71**, 521 (1993).

[16] E. Olbrich, R. Hegger, and H. Kantz, Phys. Lett. A **244**, 538 (1998).

[17] K. Kaneko, Physica D **41**, 137 (1990).

[18] A. Pikovsky and J. Kurths, Phys. Rev. Lett. **72**, 1644 (1994).

[19] G. Benettin *et al.*, C. R. Hebd. Seances Acad. Sci. **206**, 431 (1978).

[20] J. Kaplan and J. Yorke, in *Functional Differential Equations and Approximation of Fixed Points*, edited by H. Peitgen and H. Walther, Lecture Notes in Math. Vol. 730, (Springer-Verlag, Berlin, 1979). p. 228.

[21] B. Pompe, *Ranking and Entropy Estimation in Nonlinear Time Series Analysis*, in *Nonlinear Analysis of Physiological Data*, edited by H. Kantz, J. Kurths, and G. Mayer-Kress (Springer-Verlag, Berlin, 1998).

[22] B. Pompe, J. Stat. Phys. **73**, 587 (1993).

[23] C. Raab *et al*. (unpublished).