ELSEVIER

31 January 1994

PHYSICS LETTERS A

Physics Letters A 185 (1994) 77–87

A robust method to estimate the maximal Lyapunov exponent of a time series

Holger Kantz

Fachbereich Physik, Universität Wuppertal, Gauss-strasse 20, 42097 Wuppertal, Germany

Received 31 March 1993; revised manuscript received 23 November 1993; accepted for publication 6 December 1993
Communicated by A.P. Fordy

Abstract

A very simple method to compute the maximal Lyapunov exponent of a time series is introduced. The algorithm makes use of the statistical properties of the local divergence rates of nearby trajectories. It does not depend explicitly on the knowledge of the correct embedding dimension or on other parameters.

1. Introduction

Within what is called nonlinear time series analysis (see, e.g., Ref. [1]) it is of great interest to measure the Lyapunov characteristic exponents which, if positive, by definition are the most striking evidence for chaos. Whereas this is comparatively easy if one knows the equations of motion in the phase space of a system, this is a hard task if the only information at hand is a series of univariate measurements equidistant in time, i.e. a time series $\{x_t\}, t = 1, \ldots, T$.

Apart from the fact that there may be "defects" in the data like contamination with noise or too few data items, there are fundamental problems. First of all a univariate measurement is always a projection from the phase space of the system where the dynamics is deterministic to a one-dimensional interval of (possibly even discrete) values. Therefore, to reveal the properties of the dynamics, a new state space has to be constructed in which the mapping from one point of the trajectory to the successive one is unique. Usually time-delay coordinates are used. Embedding theorems [2] guarantee that for an appropriate delay (depending on the data) at most $2d + 1$ delay coordinates are enough, if d is the fractal dimension of the attractor. For noisy data delay coordinates may be suboptimal and more sophisticated techniques may be necessary [3].

The other difficulty one has to cope with is the lack of knowledge about the underlying dynamics, which is contained only implicitly in the trajectory. One can use an algorithm which directly exploits the definition of the Lyapunov exponent in the state space and therefore does not need the underlying equations of motion. However, by this in practice only the maximal exponent can be computed. Alternatively, when applying a tangent space method, one has to reconstruct the dynamics from the data. Both approaches will be briefly reviewed in the following.

Let $\boldsymbol{x}(t)$ be the time evolution of some initial condition $\boldsymbol{x}(0)$ in an appropriate state space. Then the maximal Lyapunov exponent is found with probability one by

$$\lambda_{\max} = \lim_{t\to\infty}\lim_{\epsilon\to 0}\frac{1}{t}\ln\left(\frac{|\boldsymbol{x}(t)-\boldsymbol{x}_\epsilon(t)|}{\epsilon}\right),$$
$$|\boldsymbol{x}(0)-\boldsymbol{x}_\epsilon(0)| = \epsilon, \tag{1.1}$$

for almost all difference vectors $\boldsymbol{x}(0) - \boldsymbol{x}_\epsilon(0)$. To compute $\lambda_{\max}$ numerically one can either apply Eq. (1.1) directly in real space by searching for pairs of

SSDI 0375-9601(93)E0978-J

neighbouring trajectories and look how they diverge, or one can evaluate Eq. (1.1) in the tangent space. Let $\boldsymbol{u}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_\epsilon(t)$ and $J(t)$ be the Jacobian of the map from $\boldsymbol{x}(t)$ to $\boldsymbol{x}(t+1)$. An obvious way to compute the $n \leqslant D$ largest Lyapunov exponents, where D is the dimension of the phase space, is to iterate an n-bein of vectors $\boldsymbol{u}_i(t), i = 1, \ldots, n$, using $\boldsymbol{u}_i(t+1) = J^{(t)}\boldsymbol{u}_i(t)$. From time to time one has to reorthogonalize the n-bein by means of a Gram–Schmidt procedure. The sums of the logarithms of the stretching or contraction factors of the vectors divided by the total length of the time series are the n different Lyapunov exponents. (For a slightly different algorithm in the same spirit see Ref. [4].)

The first algorithm to compute Lyapunov exponents for a time series was introduced in 1985 by Wolf et al. [5]. In delay coordinates of appropriate dimension one looks for a point of the time series which is closest to its first point. This is considered as the beginning of a neighbouring trajectory, given by the consecutive delay vectors. Then one computes the increase in the distance between these two trajectories in time. When the distance exceeds some threshold, for this point of the time series a new neighbouring trajectory is searched for, whose distance is as small as possible under the constraint that the new difference vector points more or less into the same direction as the old one. This is the direction of the local eigenvector associated to $\lambda_{\max}$. The logarithms of the stretching factors of the difference vectors are averaged in time to yield the maximal Lyapunov exponent.

In principle one should be able to compute all Lyapunov exponents by looking at the time evolutions of not distances but areas of (hyper)-surfaces, but in reality this method is limited to the maximal one. Experience shows that even for the maximal exponent this algorithm does not yield very precise results. One reason is that if no optimal new neighbouring trajectory is found, one has to use a bad one. One could think of disregarding parts of the trajectory where only "bad" neighbours are found but then the information about the unstable direction also gets lost. If the data are noisy one has to demand that the initial distance between the reference trajectory and a new neighbour is larger than the noise level, otherwise one would interpret fluctuations due to noise as deterministic divergence.

Another drawback is that the embedding dimension is an important parameter. For too small a dimension the exponent is severely overestimated, since trajectories may diverge simply because they are not neighbours in the true phase space. While too large a dimension does not in principle cause problems, it does in practice. Typically the initial distance between the neighbouring trajectories increases with the embedding dimension. Therefore the small length scales are less well explored and the number of time steps until a new neighbouring trajectory has to be searched for decreases. This induces larger errors due to the deviations of directions.

Apart from this real space or direct method one can consider working in the tangent space. As shown before, the spectrum of all Lyapunov exponents can be determined by the product of the Jacobians along the trajectory. To make use of this, the map $\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t))$ must at least approximately be known. There exist several approaches to extract this map from a time series.

Such a way was first followed by Sano and Sawada [6] and a short time later by Eckmann et al. [7], who made a more careful analysis of the possible failures of their method. In both these papers, the map $\boldsymbol{f}(\boldsymbol{x})$ is determined in an m-dimensional delay embedding space by locally linear fits to the data. From this the local Jacobians are constructed and the standard algorithm is used. An extension of this is due to Brown et al. [8], who use higher order polynomials for the fit and find improvements.

More recently, nonlinear global fits of the dynamics were used. The Jacobians of this global function are computed at each point of the time series and multiplied in their time ordering. Different methods to determine a global function are in use, e.g. fitting radial basis functions [9] or neural nets [10]. The results are very promising, at least for "synthetic data", but there are still possible failures. The quality of the fit $\boldsymbol{f}(\boldsymbol{x})$ is usually determined by the prediction error $e = \sum[\boldsymbol{f}(\boldsymbol{x}(t)) - \boldsymbol{x}(t+1)]^2$. A small value for e does not guarantee that a trajectory generated by iteration of $\boldsymbol{f}$ lies on the original attractor. If $\boldsymbol{f}$ does not produce the correct attractor, the stability properties of the measured trajectory under the fitted dynamics will be wrong and so are the Lyapunov exponents.

Again the main drawback of this whole class of algorithms is that a correct guess of the embedding dimen-

sion is indispensable. If the embedding dimension is too small, exponents may be completely wrong. On the other hand, a too large embedding dimension creates spurious exponents which in many situations are hard to identify. There are examples (e.g. Refs. [7,1]) where even the largest exponent thus found is spurious. In many cases the minimal dimension for a delay embedding is larger than the dimension of the true phase space. In such a case the tangent space methods inevitably yield spurious exponents. A promising method to detect the spurious exponents is demonstrated in Ref. [9].

In certain situations an independent algorithm to compute at least the maximal exponent with a high confidence level can be useful. This is especially the case when dealing with realistic, i.e. slightly noisy or high-dimensional data, where the other methods can yield wildly different results when being applied with different embedding dimensions or when parameters determining the fit of $\boldsymbol{f}(\boldsymbol{x}(t))$ are varied. In Section 2 we present such an algorithm. In Section 3 we compare it to the algorithm of Wolf et al. and the correlation method to determine the Kolmogorov–Sinai entropy.

2. A new method to estimate the maximal exponent

Similar to Wolf et al. we want to make use of the fact that the distance between two trajectories typically increases with a rate given by the maximal Lyapunov exponent. Of course this is true only asymptotically and we have to be more precise.

First of all, an exponential divergence sets in only after some transient time, since an arbitrary difference vector has to turn into the most unstable direction. Eventually, this will happen with probability one.

Secondly, the divergence rate of trajectories naturally fluctuates along the trajectory, with the fluctuations given by the spectrum of effective Lyapunov exponents. The maximal effective exponent λ_τ is defined in analogy to Eq. (1.1),

$$\lambda_\tau(t) = \lim_{\epsilon\to 0}\frac{1}{\tau}\ln\left(\frac{|\boldsymbol{x}(t+\tau)-\boldsymbol{x}_\epsilon(t+\tau)|}{\epsilon}\right),$$
$$\boldsymbol{x}(t)-\boldsymbol{x}_\epsilon(t) = \epsilon\boldsymbol{w}_u(t), \qquad (2.1)$$

where $\boldsymbol{w}_u(t)$ is the local eigenvector associated with the maximal Lyapunov exponent $\lambda_{\max}$. The value for $\lambda_\tau(t)$ depends on the structure in tangent space and thus is position dependent. It is approximately the same for all trajectories inside a small neighbourhood. By definition the average of $\lambda_\tau(t)$ along the trajectory is the true Lyapunov exponent.

Now let us take an arbitrary point of the time series in m-dimensional delay coordinates, $\boldsymbol{x}_t = (x_{t-m+1},\ldots,x_t)$. All delay vectors of the series falling into the ϵ-neighbourhood $\mathcal{U}_t$ of $\boldsymbol{x}_t$ will be considered as the beginning of neighbouring trajectories, which are simply given by the points of the time series consecutive in time. If we measured the distance between neighbouring trajectories in their true phase space, we would see exactly the fluctuations of the divergence rate described by the distribution of effective Lyapunov exponents (at least after some transient time, when the difference vector points into the most unstable direction). Starting from a univariate time series, we could realize this situation by measuring the distance in the embedding space. But apart from the fact that we have to fix the dimension in which we search for neighbours we do not want to distinguish any particular embedding dimension. Therefore we define the distance between a reference trajectory $\boldsymbol{x}_t$ and a neighbour $\boldsymbol{x}_i$ after the relative time τ by

$$\mathrm{dist}(\boldsymbol{x}_t,\boldsymbol{x}_i;\tau) = |x_{t+\tau}-x_{i+\tau}|, \qquad (2.2)$$

i.e. the modulus of the difference of the τth *scalar* component of the two trajectories. These distances are (generally nonlinear) projections of the difference vectors in the true phase space onto a one-dimensional subspace spanned by the observable (in the simplest case the observable itself is one coordinate of the phase space). Therefore they are modulated with $|\cos\phi|$, where ϕ is the angle between the eigenvector corresponding to $\lambda_{\max}$ and the local direction of the subspace on which the observable lives. Like the effective Lyapunov exponent the angle ϕ depends on the position in phase space and thus is nearly the same for all neighbours $\boldsymbol{x}_i \in \mathcal{U}_t$ of a given reference trajectory $\boldsymbol{x}_t$, if the distance in phase space is sufficiently small. In Fig. 1 this is shown for a set of trajectories of an experimental time series.

In order to measure the maximal Lyapunov exponent we fix t, search for all neighbours $\boldsymbol{x}_i$ inside an ϵ-neighbourhood $\mathcal{U}_t$ and compute the average of the distances between all neighbouring trajectories and the reference trajectory $\boldsymbol{x}_t$ as a function of τ. As defined

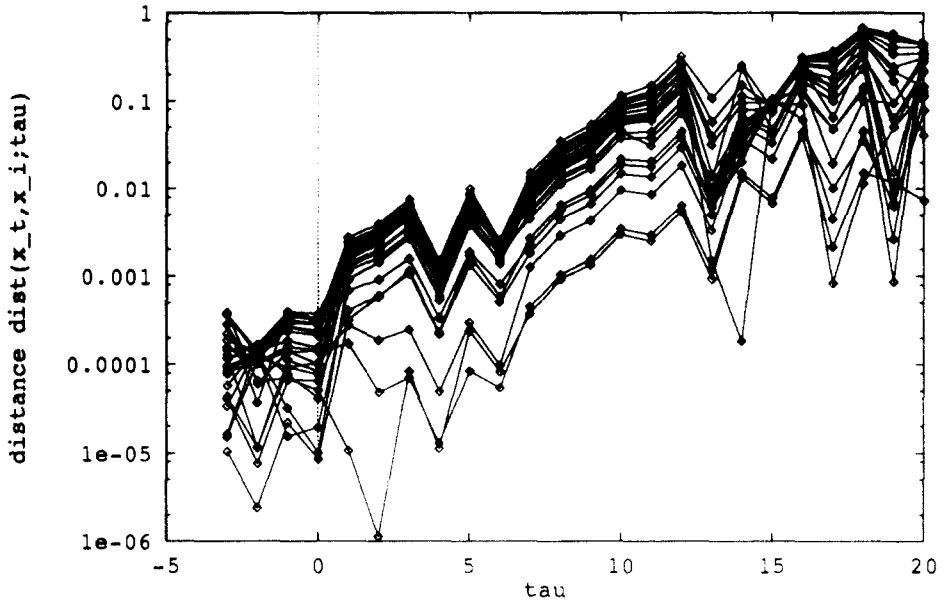

Fig. 1. The distances $\mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)$ (Eq. (2.2)) for 25 neighbouring trajectories of a single reference trajectory $\boldsymbol{x}_t$. The embedding dimension is $m = 4$ and $\epsilon = 0.0004$. The points for $\tau \leqslant 0$ correspond to the elements of the delay vectors $\boldsymbol{x}_i \in \mathcal{U}_t$, which by construction are $\leqslant \epsilon$. The few crossings on small scales indicate some noise and the fact that the difference vectors have to turn into the unstable direction, whereas crossings on large scales are caused by foldings of the attractor. The data are a Poincaré section of an experimental time series of the Zürich NMR laser experiment after noise reduction. Further details about the data can be found in Ref. [12]. Note that all curves are subject to the same local structures in phase space, reflected by being almost parallel for $0 < \tau < 12$. This can be found only for highly deterministic data.

in Eq. (2.2), τ is the relative time referring to the time index of the starting point. This is nothing but a (linear) average over the lines in Fig. 1, which contains all the above mentioned fluctuations. To get rid of them we take the logarithm of these average distances, which yields the local effective Lyapunov exponent plus a fluctuation given by the angle ϕ. Now we average in t over the full length of the time series. The local angles are averaged out and the effective exponents are averaged to the true Lyapunov exponent. Using a sophisticated algorithm for searching neighbours like the one in Refs. [1,11] this can be done very fast.

Thus we compute

$$S(\tau) = \frac{1}{T}\sum_{t=1}^{T} \ln\left(\frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t} \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)\right). \qquad (2.3)$$

Initially, the difference vectors in the true phase space are pointing in any direction, therefore the distance behaves like $\mathrm{dist} = \sum_i a_i \exp(\lambda_i t)$, where λ_i are the effective Lyapunov exponents in the different stable and unstable directions. Thus for small τ we cannot expect a scaling behaviour. For an intermediate range of τ, $S(\tau)$ increases linearly with the slope λ which is our estimate of the maximal Lyapunov exponent. This is the scaling range, where on the one hand τ is large enough such that nearly all distance vectors point into the unstable direction and on the other hand the corresponding distances $\mathrm{dist}(\tau)$ are smaller than the size of the attractor. When they reach the latter size, $S(\tau)$ asymptotically tends towards a constant (which is $< \ln 1/4$ if the diameter of the attractor is normalized to 1) since the distances cannot grow any more.

If the data are noisy, the typical distance between two nearby trajectories is of the order of the noise level. If we choose ϵ smaller than the noise amplitude and if we find neighbours for this value, $S(\tau)$ jumps from a value smaller than $\ln \epsilon$ to a value given by the noise level at $\tau = 1$. If this value is not too large, one can still find a scaling range, and the exponent thus found is not affected by the noise.

To see this let us analyze Eq. (2.3) in more detail. Due to the measurement one observes the projection $\mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau) = \Delta_{i,\tau} |\cos \phi_{t+\tau}|$, where $\Delta_{i,\tau}$ is the length of the difference vector in the true phase space. Additive noise in the observables, which we assume to be independently identically distributed (iid), creates a corresponding additive term $\sigma_{i,\tau}$ in the distances $\mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)$. For the average over the neighbourhood of a single trajectory $\boldsymbol{x}_t$ one finds

$$\begin{aligned} &\ln[\langle \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)\rangle_{i\in\mathcal{U}_t}] \\ &= \ln\left(\frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t} |\Delta_{i;\tau}\cos(\phi_{t+\tau}) + \sigma_{i,\tau}|\right) \\ &= \ln\left(\frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t} |\Delta_{i;\tau-1}\, \mathrm{e}^{\lambda_{t+\tau}}\cos(\phi_{t+\tau}) + \sigma_{i,\tau}|\right). \end{aligned} \tag{2.4}$$

If the noise level is sufficiently smaller than the deterministic part of the distances we find approximately

$$\begin{aligned} \ln[\langle \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)\rangle_{\mathcal{U}_t}] &\approx \lambda_{t+\tau} + \ln(\langle \Delta_{i;\tau-1}\rangle_{\mathcal{U}_t}) \\ &+ \ln|\cos(\phi_{t+\tau})| + \frac{\langle \sigma_{i,\tau}\rangle_{\mathcal{U}_t}}{\langle \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)\rangle_{\mathcal{U}_t}}. \end{aligned} \tag{2.5}$$

Thus the slope of the curve $S(\tau)$ is

$$\begin{aligned} s(\tau) &= \\ &\langle \ln\langle \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)\rangle - \ln\langle \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau - 1)\rangle\rangle_t \\ &\approx \lambda + \left\langle \frac{\langle \sigma_{i,\tau}\rangle}{\langle \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)\rangle}\right\rangle_t \\ &- \left\langle \frac{\langle \sigma_{i,\tau-1}\rangle}{\langle \mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau - 1)\rangle}\right\rangle_t. \end{aligned} \tag{2.6}$$

Due to the assumed independence of the noise of the deterministic structure of the data, the last two terms on the r.h.s. are of second order in the noise level. Thus the effect of noise on $s(\tau)$ is negligible if all $\langle \mathrm{dist}(\boldsymbol{x}_0, \boldsymbol{x}_i; \tau)\rangle$ are large compared to the noise level.

In summary, our numerical value for the maximal Lyapunov exponent is the slope of the curve $S(\tau)$ in the scaling region. It does not contain the embedding dimension explicitly, but nevertheless it enters. We have to fix a dimension m for the delay embedding space in which we search for neighbouring trajectories. If m is too small there is a nonzero probability that two trajectories are close in the embedding space but not in the true phase space. In this case their distance $\mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)$ may increase strongly until τ is large enough such that $\tau + m$ is a "good" embedding. From this point on the increase is again determined by the maximal Lyapunov exponent. However, the distance at this point can already be so large that the scaling region is strongly reduced. To avoid this we compute $S(\tau)$ for different values of m. Algorithmically this is very easy. We first search for neighbours in only 2 dimensions. When accumulating the distances $\mathrm{dist}(\boldsymbol{x}_t, \boldsymbol{x}_i; \tau)$ we do this in different histograms in $m_{\max}$, which is the maximal embedding in which $\boldsymbol{x}_i$ remains in the ϵ-neighbourhood of $\boldsymbol{x}_t$. So we end up with the two parameters m and ϵ which should be varied in some reasonable range without yielding contradictory results.

For short data sets, high embedding dimension and small ϵ, only very few neighbouring trajectories generally are found. Therefore the curves may fluctuate due to low statistics. This is not a serious problem unless no neighbour is found at all for many points $\boldsymbol{x}_t$. In such a situation only the parts of the attractor are explored where the natural measure is highest. If the spatial distribution of the effective Lyapunov exponents is inhomogeneous this may lead to a wrong estimate of the average. Note that in Eq. (2.6) the averages are taken in such a way that exponents are averaged according to the natural measure only if all points of the time series have neighbours. Thus when working with fixed ϵ-neighbourhoods, one has to keep track of how many points have no neighbours and compare the results with different ϵ. We do not recommend fixed mass neighbourhoods since this will lead to a mixture of length scales and reduce the scaling range.

To illustrate the algorithm we finish this section with some examples. First, we compute the Lyapunov exponent of the well-known Hénon map. In delay coordinates it reads $x_{t+1} = 1 - ax_t^2 + bx_{t-1}$, where we have chosen $a = 1.4$ and $b = 0.3$ as usual. After discarding the first part of the trajectory we record a time series of length $T = 2000$. For this in Fig. 2 the curves $S(\tau)$ are presented for different m and ϵ. The points for $\tau \leqslant 0$ are obtained for the elements of the embedding vectors, whose differences by construction are smaller than ϵ. At $\tau = 0$ the trajectories start to diverge. The average over the slopes of the straight segments of the curves, obtained with least squares fits, is $\lambda_{\mathrm{num}} = 0.418 \pm 0.003$, which should be compared

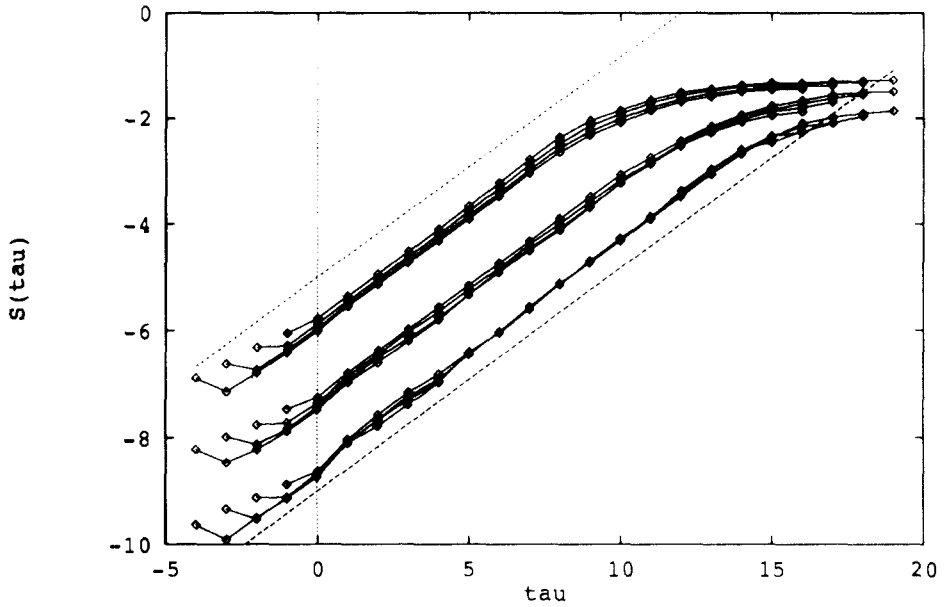

Fig. 2. $S(\tau)$ for a Hénon trajectory of length 2000. The different curves correspond to $\epsilon = 0.0005$, 0.002 and 0.008 (the three bunches from bottom to top) and embedding dimension $m = 2$–5. The dashed lines have slopes $\lambda_{\text{exact}} = 0.4169$. Again $\tau \leqslant 0$ corresponds to the components used to define the local neighbourhoods.

to the exact value 0.4169. (One should note that the "exact" value given in Refs. [8,10] is incorrect.)

In Fig. 3 we show the same for a Hénon trajectory with 1% uniformly distributed white noise added. Still the Lyapunov exponent is determined quite accurately, the same average as before yielding $\lambda_{\text{num}} = 0.41 \pm 0.015$. One can clearly observe the jump at $\tau = 0$ if ϵ is smaller than the noise level. To have enough neighbours to demonstrate this we have used a longer trajectory of length 4000, whereas the curves for $\epsilon = 0.016$ are obtained from only 2000 points. Although the result is very satisfying, one should in addition use an algorithm to reduce noise before computing the Lyapunov exponent, for example a principal component decomposition [13] or a noise reduction scheme as in Ref. [14]. Having computed $S(\tau)$ for small ϵ before and after the noise reduction, its success can be determined by a comparison of the heights of the jumps. The ratio is a direct measure of the dynamical error reduction [14].

The Hénon system, although nonhyperbolic, in another sense is very simple. Its true phase space is identical to the two-dimensional delay embedding space. Therefore we also consider a system without this property, namely the Ikeda map. It is given by

$$\begin{aligned} \varphi &= 0.4 - 6/(1 + x_t^2 + y_t^2), \\ x_{t+1} &= 1 + 0.9\,(x_t \cos\varphi - y_t \sin\varphi), \\ y_{t+1} &= 0.9\,(x_t \sin\varphi + y_t \cos\varphi). \end{aligned} \tag{2.7}$$

The two exact Lyapunov exponents are $\lambda_+ = 0.505$ and $\lambda_- = -0.715$. The dimension of the attractor is approximately 1.7 and it seems that a "good" delay embedding has to be at least three-dimensional. Consequently, in Fig. 4 we see that an embedding dimension $m = 2$ is not enough to see a clear scaling region, but we nevertheless do not obtain incorrect results. For the scaling regions of the curves with dimension $m = 3, \ldots, 6$ we fitted straight lines. The average of their slopes is $\lambda_{\text{num}} = 0.50 \pm 0.01$.

Finally, let us consider a system with more than one positive Lyapunov exponent. A very simple map is the generalized Hénon map [15], which in three dimensions reads

$$x_1' = a - x_2^2 - b x_3, \quad x_2' = x_1, \quad x_3' = x_2. \tag{2.8}$$

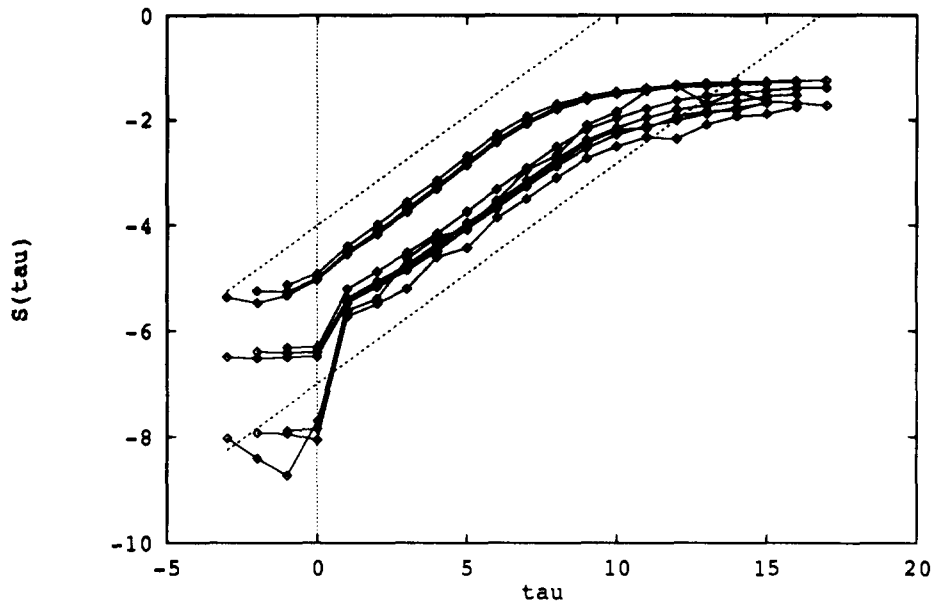

Fig. 3. $S(\tau)$ for a Hénon trajectory with 1% additive noise. The noise can be clearly seen in the jump of the curves from $S(\tau) \approx \ln \epsilon$ to $S(\tau) \approx -5.33$, which is the logarithm of the amplitude of the uniformly distributed noise. The three bundles of curves correspond to $\epsilon = 0.001$, 0.004, and 0.016, embedding dimension 2–4. The fluctuations in the low lying curves are mainly a consequence of too few neighbours. The two dashed lines have slope 0.4169.

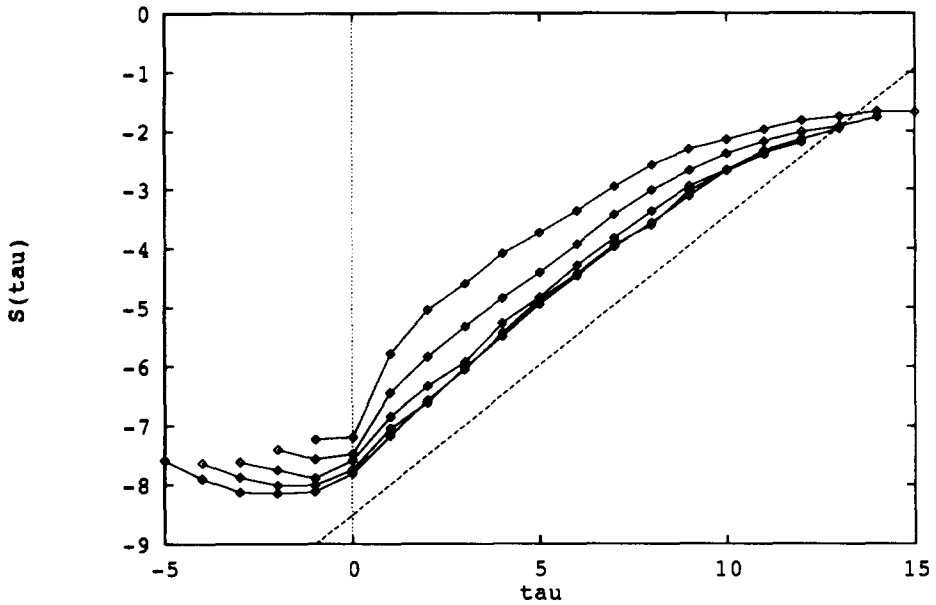

Fig. 4. $S(\tau)$ obtained from a time series of the y-variable of the Ikeda map of length 5000. The embedding dimension is $m = 2$–6, $\epsilon = 0.002$. For small m the scaling region is not clearly visible since the neighbouring trajectories are not close in the true phase space. The straight line has slope $\lambda_{\text{exact}} = 0.505$.

For $a = 1.79$ and $b = 0.1$ we find the "exact" exponents $\lambda_1 = 0.225$, $\lambda_2 = 0.187$, and $\lambda_3 = -2.715$. Using a trajectory of length $T = 4000$ ($T = 8000$) and as the observable $o = \sum_{i=1}^{3} x_i$, we find an exponent of 0.218 ± 0.007 (0.225 ± 0.004) when averaging over the slopes in the scaling regions of $S(\tau)$ for $m = 4, \ldots, 8$ and $\epsilon = 0.004$, 0.008, and 0.016.

3. Comparison with look-alike methods

Our approach is very much in the spirit of the algorithm of Wolf et al. [5], but has some very strong advantages. Obviously our algorithm is somewhat simpler, since we do not compare directions but only distances. As a consequence we require fewer (free) parameters. In the Wolf algorithm, one has the embedding dimension, an upper bound $s_{\max}$ of the distances when one has to choose a new neighbouring trajectory, an angle $\phi_{\max}$ by which the direction of the new difference vector may differ from the old one, and for noisy data, a distance s_{min} which is the minimal distance between the reference trajectory and its new neighbour.

More important, in our approach the statistical fluctuations are much smaller. For each point of the reference trajectory not one but several neighbouring trajectories are evaluated. Furthermore, along the contracting directions new trajectories continuously approach the central trajectory. To use all of them we average over all $\mathcal{U}_l$ along the whole time series. We can therefore give a much improved estimate of the exponent for small data sets.

Finally, since we record the τ dependence explicitly, we can cut off the effect of redirecting of difference vectors into the unstable directions as well as the saturation effects on large scales. When evaluating the scaling range of $S(\tau)$, one can estimate an error of λ. In cases where no true scaling range can be seen, we are warned that the notion of Lyapunov exponents is questionable for this data set since the average divergence rate is scale dependent. Our experience shows this is often to be the case in experimental data sets due to the lack of determinism in the data. In contrast to this the Wolf algorithm and all tangent space algorithms do not yield an error estimate other than by rerunning the procedure with different parameter values.

In Fig. 5 we show the results of the Wolf algorithm for the same Hénon trajectory of length $T = 2000$ used in fig. 2. We see that the exponent is underestimated by about 3% whereas the deviation using our algorithm is less than 0.3%. Note that the data used here are ideal in the sense that they are noise-free and really low-dimensional. If this is not the case, the results of the Wolf algorithm are much worse and may be very inconsistent for different choices of the parameters.

The sum of all positive Lyapunov exponents equals the Kolmogorov–Sinai entropy of a system [4]. A computational method for the latter exists which in some sense resembles the algorithm of this paper. All generalized order-q Renyi entropies are conveniently computed by evaluating the generalized correlation sum [16]. The Shannon entropy is obtained in the limit $q \to 1$.

$$h_1(\epsilon, \tau) = \frac{1}{N} \sum_{i=1}^{N} (\ln N - \ln n_i). \tag{3.1}$$

Here n_i are the number of vectors in τ-dimensional delay coordinates which are closer than ϵ to the ith point of the time series. The Kolmogorov–Sinai entropy h_{KS} is given by the limit

$$h_{\mathrm{KS}} = \lim_{\tau \to \infty} h_1(\epsilon, \tau + 1) - h_1(\epsilon, \tau). \tag{3.2}$$

When using the sup-norm we can reinterpret Eqs. (3.1) and (3.2) in the following sense. For fixed ϵ we look for all neighbours of a given point in a τ-dimensional embedding space for small τ. We then successively add one more coordinate and observe how many trajectories remain in the ϵ-neighbourhood. Assuming an exponential decrease, in Eq. (3.1) one simply averages over the escape rates of all points $\boldsymbol{x}_l$ in the time series. Thus the difference between this and our approach is that here the distance is kept fixed and the number of trajectories decreases exponentially, whereas we fix the number of trajectories and the distance diverges with the same rate.

It is obvious that in Eq. (3.1) finite sample corrections, derived in Ref. [17], are indispensable, since otherwise for large τ (i.e. typically small n_i) the discreteness of the "probabilities" n_i/N destroys the scaling behaviour. In particular the case $n_i = 0$ causes problems. Furthermore, the length scale is fixed by the cutoff ϵ. In the presence of noise or if the expansion

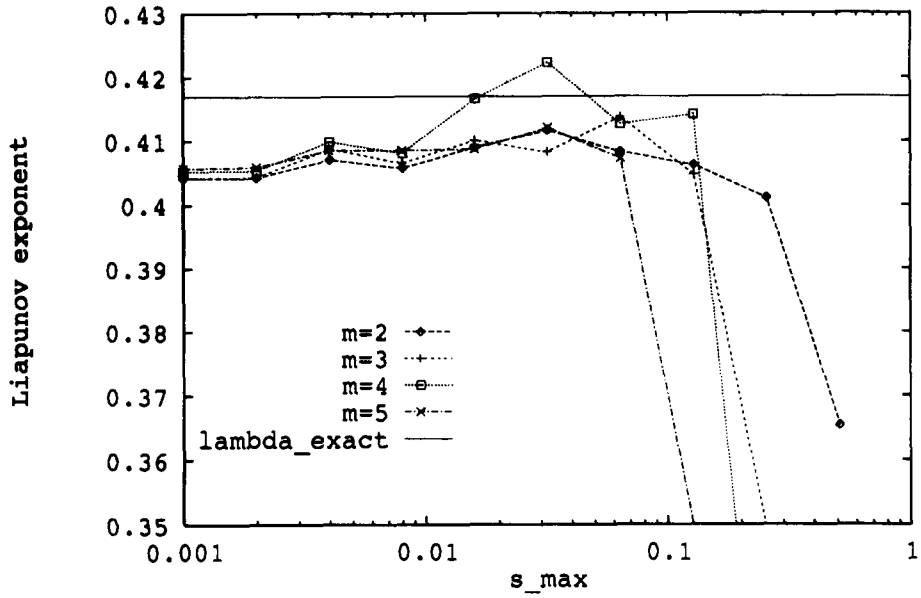

Fig. 5. Results of the algorithm of Wolf et al. for the same data as used in Fig. 2 (i.e. Hénon map, 2000 points). The computed maximal Lyapunov exponent is plotted as a function of the distance $s_{\max}$, at which a new neighbouring trajectory is searched for. The angle $\phi_{\max}$ is fixed by $\cos\phi_{\max} = 0.95$, and the embedding dimension m varies from 2 to 5.

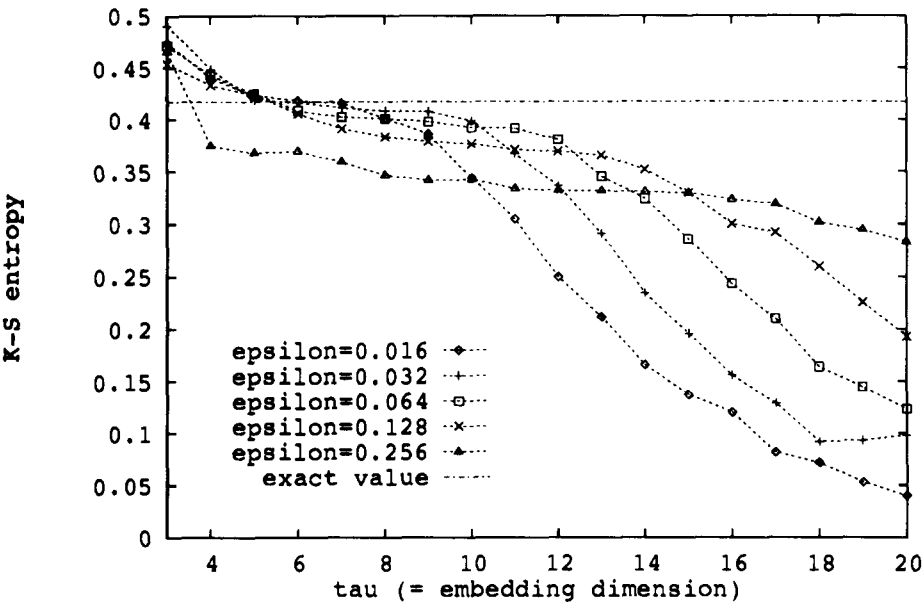

Fig. 6. Numerical estimates of the Kolmogorov–Sinai entropy of a Hénon trajectory of length 20000. 5000 points were used as centre points of the correlation algorithm. ϵ is measured in units of the attractor size. Obviously for large ϵ one has already saturation effects.

rate for other reasons is scale dependent, one has to rerun the algorithm with different ϵ.

The severest drawback of this procedure to compute the entropy is that for short data sets the number of remaining neighbours very quickly goes to zero and therefore a scaling range may be difficult to establish. This is demonstrated in Fig. 6, where we use ten times as much data as in Fig. 2 and yet the results are still not completely satisfactory. For only 2000 points an estimate for the entropy can hardly be obtained.

One possible improvement is to use a tube increasing exponentially with a rate β instead of a fixed diameter tube. Then one has to add β to the "entropy" the algorithm yields. Obviously results so obtained would be more stable since the number of neighbours decreases less rapidly. But now $\epsilon \exp(\tau_{\max}\beta)$ would have to remain smaller than the size of the attractor, and hence for large ϵ one is strongly limited in $\tau_{\max}$. Furthermore, with variable diameter tubes the data for various ϵ cannot be used to determine directly the information dimension d_1.

The correlation method is more powerful in the computation of h_2, the correlation entropy. In this case, the finite sample corrections are zero [17] and already short time series yield quite precise values of h_2. While h_2 is a lower bound for h_{KS}, it can generally only be used as a consistency check since in many cases it is significantly smaller than h_{KS}.

4. Conclusions

Our method directly exploits the exponential increase of small distances. They are measured in the one-dimensional space of the time series. Embeddings are only used to distinguish between false and true neighbours and hence the choice of an optimal embedding is not crucial. This algorithm is a useful tool in cases where a fast and robust method is required and when the more sophisticated tangent space algorithms are too time consuming or yield questionable results. Its particular strength lies in the estimate of an error. However, if for a given data set the scaling range found by this algorithm is sufficiently large, we strongly recommend the Sano–Sawada and Eckmann et al. tangent map algorithms to obtain the whole spectrum of exponents. For high quality data the latter algorithms are superior since they yield more information. We have shown that our algorithm also works in the presence of noise and for small data sets, and that it is an improvement with respect to the Wolf algorithm.

After submission of the manuscript a paper of Rosenstein, Collins and de Luca [18] was published in which mainly the same method of determining the maximal exponent is used. The main differences are that in Ref. [18] the distance between trajectories is defined as a (maximum or Euclidean) norm in the embedding space, which seems to reduce the scaling regions a bit, since it mixes the different delay coordinates. Secondly, they use only one neighbour per time t and not all neighbours within a certain neighbourhood, which might induce larger statistical errors, especially in the presence of noise.

Acknowledgement

We thank P. Grassberger, I. Hoffmann and T. Schreiber for valuable discussions. The time series underlying fig. 1 was conveyed by E. Brun, L. Flepp and J. Simonet. Since it is sometimes difficult for theoreticians to get access to experimental data we are strongly indebted for this. This work is supported by Deutsche Forschungsgemeinschaft, SFB 237.

References

[1] P. Grassberger, T. Schreiber and C. Schaffrath, Int. J. Bifurc. Chaos 1 (1991) 521.
[2] T. Sauer, J.A. Yorke and M. Casdagli, J. Stat. Phys. 65 (1991) 579.
[3] J.F. Gibson, J.D. Farmer, M. Casdagli and S. Eubank, Physica D 57 (1992) 1.
[4] J.P. Eckmann and D. Ruelle, Rev. Mod. Phys. 57 (1985) 617.
[5] A. Wolf, J.B. Swift, L. Swinney and A. Vastano, Physica D 16 (1985) 285.
[6] M. Sano and Y. Sawada, Phys. Rev. Lett. 55 (1985) 1083.
[7] J.P. Eckmann, S. Oliffson Kamphorst, D. Ruelle and S. Ciliberto, Phys. Rev. A 34 (1986) 4971.
[8] R. Brown, P. Bryant and H.D.I. Abarbanel, Phys. Rev. A 43 (1991) 2787.
[9] U. Parlitz, Int. J. Bifurc. Chaos 2 (1992) 155.
[10] R. Gencey and W.D. Dechert, Physica D 29 (1992) 142.
[11] T. Schreiber, in: SFI Studies in the science of complexity, Proc. Vol. XVII. Predicting the future and

understanding the past, eds. A.S. Weigend and N.A. Gershenfeld (Addison-Wesley, Reading, MA, 1993).
[12] H. Kantz, T. Schreiber, I. Hoffmann, T. Buzug, G. Pfister, L.G. Flepp, J. Simonet and E. Brun, Phys. Rev. E 48 (1993) 1529.
[13] D. Broomhead and G.P. King, Physica D 20 (1986) 217.
[14] P. Grassberger, R. Hegger, H. Kantz, C. Schafrath and T. Schreiber, Chaos 3 (1993) 127.
[15] G. Baier and M. Klein, Phys. Lett. A 151 (1990) 281.
[16] P. Grassberger and I. Procaccia, Phys. Rev. A 28 (1983) 2591.
[17] P. Grassberger, Phys. Lett. A 128 (1988) 369.
[18] M.T. Rosenstein, J.J. Collins and C.J. de Luca, Physica D 65 (1993) 117.