Printed ISSN 1330–0008
Online ISSN 1333–9125
CD ISSN 1333–8390
CODEN FIZAE4

INTRODUCING NONLINEAR TIME SERIES ANALYSIS IN UNDERGRADUATE COURSES

MATJAŽ PERC[1]

Department of Physics, Faculty of Education, University of Maribor, Koroška cesta 160, SI-2000 Maribor, Slovenia

Received 25 October 2004; Accepted 12 July 2006
Online 1 December 2006

This article is written for undergraduate students and teachers who would like to get familiar with basic nonlinear time series analysis methods. We present a step-by-step study of a simple example and provide user-friendly programs that allow an easy reproduction of presented results. In particular, we study an artificial time series generated by the Lorenz system. The mutual information and false nearest neighbour method are explained in detail, and used to obtain the best possible attractor reconstruction. Subsequently, the times series is tested for stationarity and determinism, which are both important properties that assure correct interpretation of invariant quantities that can be extracted from the data set. Finally, as the most prominent invariant quantity that allows distinguishing between regular and chaotic behaviour, we calculate the maximal Lyapunov exponent. By following the above steps, we are able to convincingly determine that the Lorenz system is chaotic directly from the generated time series, without the need to use the differential equations. Throughout the paper, emphasis on clear-cut guidance and a hands-on approach is given in order to make the reproduction of presented results possible also for undergraduates, and thus encourage them to get familiar with the presented theory.

PACS numbers: 01.50.-i, 05.45.Tp UDC 530.182
Keywords: nonlinear time series analysis, physics education

1. *Introduction*

Nonlinear time series analysis theory offers tools that bridge the gap between experimentally observed irregular behaviour and deterministic chaos theory. The theory of deterministic chaos offers an explanation for irregular behaviour of sys-

[1]Correspondence to: Matjaž Perc,University of Maribor, Department of Physics, Faculty of Education, Koroška cesta 160, SI-2000 Maribor, Slovenia; Tel.: +386 2 2293643, Fax: +386 2 2518180; E-mail address: matjaz.perc@uni-mb.si, E-page: http://fizika.tk

tems that are not influenced by stochastic inputs. To this purpose, characteristic quantities, such as Lyapunov exponents are usually derived from differential equations that describe temporal evolution of the system. These quantities are often referred to as invariants, since they do not depend on initial conditions, and so represent characteristic properties of the system. A positive maximal Lyapunov exponent, for example, is characteristic for chaotic systems, whereas systems with solely non-positive exponents are usually referred to as regular. The aim of this paper is to describe and provide user-friendly programs for basic nonlinear time series analysis methods that are in our opinion necessary to confirm or reject the presence of deterministic chaos in a time series, without the help of differential equations. Particular emphasis is given on a step-by-step guidance and reproducibility of obtained results, so that even individuals with little or no experience can get familiar with the presented material and develop an interest for this field of research.

In 1963, meteorologist Ed Lorenz [1] derived a fairly simple three-dimensional set of first-order nonlinear differential equations

$$\dot{x} = \sigma(y - x)\,, \tag{1}$$

$$\dot{y} = rx - y - xz\,, \tag{2}$$

$$\dot{z} = xy - bz\,, \tag{3}$$

which, in a very simplified way, model the convective rolls in the atmosphere. In an even more simplified way, this set of differential equations can be seen as a qualitative model for the weather. For certain values of parameters σ, r and b (for example, $\sigma = 10$, $r = 25$, $b = 8/3$), the system has a positive maximal Lyapunov exponent, and thus expresses irregular deterministic behaviour, which we term chaotic [2−4]. In view of this brief description of well-known results, one may be tempted to conclude that the weather on our planet is chaotic. This may be true, and we will make no attempts trying to disprove this conclusion. Nevertheless, some doubtful students may not be readily convinced and could argue that chaos is nothing more than a mathematical artefact; a phenomenon non-existing outside the simulations of our computer. The question is, can we, besides computer simulations and occasional poor weather forecast, offer any other more convincing evidence that the weather is indeed chaotic? The answer is conditionally affirmative, as we will elucidate below. The remedy lies in the nonlinear time series analysis [5−7], which enables us to extract characteristic quantities, i.e. invariants such as the maximal Lyapunov exponent, of a particular system solely by analysing the time course of one of its variables. In theory, it would then be possible to collect temperature measurements in a particular city for a given period of time and employ nonlinear time series analysis to actually confirm the chaotic nature of the weather. Despite the fact that this idea is truly charming, its realization is not feasible quite so easily, so let us face reality one step at a time.

Before one starts to employ nonlinear time series analysis methods, it is necessary to check some basic requirements a time series has to fulfil in order to qualify for such an undertaking. Blindly, of course, all data sets are good enough, and the programs will always do their job readily. However, the obtained positive maximal Lyapunov exponent, for example, cannot be considered as an indicator for chaos, if the studied time series doesn't result from a *stationary* process. The same statement can be made with respect to *determinism*. Thus, the time series must originate from a stationary deterministic process in order to justify the calculation of the maximal Lyapunov exponent. Only then, the obtained value of the exponent will have a meaning and truly posses the power to discriminate between a chaotic and regular system. Hence, only after we have established that the studied data set originates from a stationary deterministic process, we can move on to calculate the maximal Lyapunov exponent. The most basic step in this procedure is to construct a proper embedding space from the time series. For this purpose, we have to determine the proper embedding delay and embedding dimension. There exist two methods, developed in the framework of nonlinear time series analysis, that enable us to successfully perform the desired tasks. The mutual information method [8] yields an estimate for the proper embedding delay, whereas the false nearest neighbour method [9] enables us to determine a proper embedding dimension. All above-mentioned methods, as well as currently unfamiliar terms, will be accurately described in the next section.

The construction of a proper embedding space is, however, not only necessary to calculate the maximal Lyapunov exponent of a time series, but also to perform trustworthy stationarity and determinism tests. Therefore, we will first describe and demonstrate the usage of the mutual information method and the false nearest neighbour method, and afterwards introduce the stationarity and determinism test. Finally, the algorithm for the calculation of the maximal Lyapunov exponent will be presented and deployed. When trying to reproduce the results obtained in this article, we suggest following the same sequence of tasks as described above. This sequence seems reasonable also for any other data set.

All presented methods will be tested on a numerical time series generated by the Lorenz equations [Eqs. $(1-3)$] for parameter values $\sigma = 10$, $r = 25$, $b = 8/3$, sampled at a time interval of 0.01 s and occupying 40000 points, if not explicitly stated otherwise. Although it is much more interesting to study an experimental time series with unknown characteristics, there is much to be gained, especially from the educational point of view, by studying the time series of a known deterministic system. In particular, since the correct results are known from the chaos theory, it is easy to verify if the results obtained with nonlinear time series analysis methods are at least qualitatively correct. Furthermore, by testing the programs on familiar data sets, i.e. with a known number of degrees of freedom and the maximal Lyapunov exponent, the inexperienced user can get confidence in the operations executed by a program for various parameter settings, and thus familiarize with the available tool kit. The final result, which we are going to reproduce below using nonlinear time series analysis methods, is that the examined data set originates from a stationary deterministic process with three degrees of freedom and a positive maximal Lyapunov exponent, from which we can conclude that the system under study [1] is deterministically chaotic.

Before we finally delve into the beauty of nonlinear time series analysis, let us emphasize that this article is not of review type, meaning that it doesn't describe all relevant methods. It is rather a collection of carefully chosen methods, which should allow a person just getting familiar with the subject to get inspiring and above all meaningful results without having to delve too deep into the existing theory. An expert will surely miss a word or two on dimension estimates [10 − 12], prediction algorithms [13], noise reduction schemes [14,15], or surrogate data techniques [16,17], which all form an important part of nonlinear time series analysis. However, all these topics were left out because of extended amount of knowledge required to perform these tasks successfully, and also to assure increased readability an understanding among readers just getting familiar with the subject. Since a regular article can hardly be long enough to describe all relevant methods and techniques properly, the interested reader is advised to seek further information about these topics in already cited original research papers and books [5 − 7]. Some additional pointers to the relevant literature will also be given in the next section.

2. *Methods and implementation*

Following the sequence of tasks we have outlined above, let us first introduce the method known as delay coordinate embedding. It enables us to construct a phase space of a system from a single observed variable. This reconstructed phase space is usually referred to as the embedding space. Consider then the time course of a variable of the Lorenz system. How are we to reconstruct the phase space of the system not knowing the other two variables? The intuitive solution lies in the fact that all variables in a deterministic dynamical system are generically connected. With simpler words, we can say that they influence one another, as can be seen by observing Eqs. (1 − 3). For example, the time evolution of the variable x is through a subtraction directly dependent on the variable y, whereas the time evolution of the variable y similarly depends on the variable x as well as variable z. The direct consequence of this fact is the following. If at the time t only the value of the variable x is known, then another measurement of the variable x at a future time $t + \tau$ will implicitly carry some information also about variables y and z. By continuing the measurement of variable x at times $t + 2\tau$, $t + 3\tau$, $\ldots$, we thus continuously gather information not just about variable x, but also about variables y and z. In fact, if τ is chosen properly, the amount of information we thereby obtain about y and z is large enough to allow us to introduce the values of variable x at times $t + \tau$ and $t + 2\tau$ as substitutes for the original variables. Although this result seems shocking, there exists a rigorous mathematical proof of a theorem that confirms the validation of the above reasoning. The theorem is usually termed as the embedding theorem, and was formally proven by Takens [18]. It states that for a large enough embedding dimension m, the delay vectors

$$\mathbf{p}(t) = (x_t, x_{t+\tau}, x_{t+2\tau}, \ldots, x_{t+(m-1)\tau}) \tag{4}$$

yield a phase space with exactly the same invariant quantities as the original system. In Eq. (4), variables $x_t, x_{t+\tau}, x_{t+2\tau}, \ldots, x_{t+(m-1)\tau}$ denote the values of variable x at times $t, t+\tau, t+2\tau, \ldots, t+(m-1)\tau$ whereas τ is the so-called embedding delay. Note that the original theorem by Takens is formulated with respect to the embedding dimension m, and not with respect to the embedding delay τ as we have exemplified above. We could afford this minor discrepancy since the proper embedding dimension for the Lorenz system is known; it is the same as the dimensionality of the dynamical system. In general, however, this fact is not known and the correct formulation of the theorem, as proven by Takens, has to be used. A very nice intuitive demystification of the embedding theorems can be found in the book by Abarbanel [5], whereas a more mathematical approach can be found in the original articles by Takens [18] and Sauer et al. [19].

While the implementation of Eq. (4) should not pose a problem, the correct choice of proper embedding parameters τ and m is a somewhat different matter. The most direct approach would be to visually inspect phase portraits for various τ and m trying to identify the one that looks best. The word "best", however, might in this case be very subjective. Examine the phase portraits shown in Fig. 1. For a student with a great imagination, the reconstruction presented in Fig. 1c might seem as the "best", while a more conservative character would perhaps prefer the one shown in Fig. 1a. In fact, the phase space portrait in Fig. 1b is the best choice amongst the presented ones. But how can we tell? The problem with the phase portrait presented in Fig. 1a is that it looks a bit compressed. Some might say that the attractor does not have well-evolved folding regions. This manifests so that the nicely expressed arcs at both ends of the attractor come crushing together in the middle in a seemingly violent manner, making the structure of the attractor indistinguishable at small scales. On the other hand, the phase portrait in Fig. 1c is a bit to complex. In this case, the trajectory folds and wraps around very frequently. Although this yields and appealing picture, it also introduces a seemingly stochastic component, especially in the middle of the phase space, which is usually not a good appearance for a proper embedding. Clearly, the phase portrait in Fig. 1b is the best. The presented attractor has nice evolved folding regions, but still looks compact and deterministic.

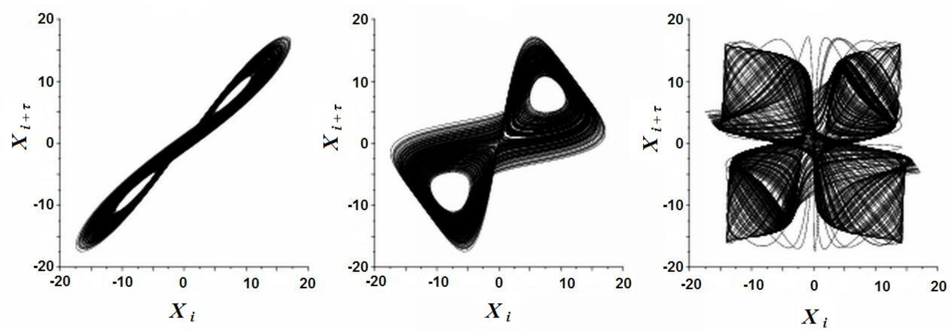

Fig. 1. Reconstructed phase space obtained with different embedding delays and dimensions. a) $\tau = 3, m = 2$. b) $\tau = 17, m = 3$. c) $\tau = 100, m = 4$.

Before we continue with the discussion of the results presented in Fig. 1, let us mention that they are easily reproducible with our program *embed.exe* that can be downloaded from our web page [20]. The program graphically displays 2D projections of four different embeddings simultaneously to enable a better comparison of results for various τ and m. Some general instructions for the usage of our tool kit are given in the Appendix, whereas a more detailed description can be found at the download site.

Despite the fact that it is advisable, even necessary, to visually inspect the data before performing any calculations, and you might feel satisfied with the above reasoning for determining proper embedding parameters, and it might even seem that the given instructions will do the job also in other cases, be aware of the following. First, pay attention to the values of embedding delays used for the phase space portraits in Fig. 1. Notice that the delays differ almost by orders of magnitude. In this case, it is easy to distinguish between a proper and a less proper embedding. In reality, however, the task of finding the proper embedding delay becomes increasingly difficult as the value approaches the optimum. Second, be aware of the fact that although it is said that Fig. 1a was obtained by setting $m = 2$, Fig. 1b by setting $m = 3$ and Fig. 1c by setting $m = 4$, you would not notice the difference if all pictures were obtained with $m = 2$ or $m = 100$, since the first two embedding coordinates depend only on the embedding delay [see Eq. (4)]. Certainly, if you chose an embedding delay greater than two, you could go ahead and examine also other possible phase space projections. However, this would really make the task of finding the proper embedding mind boggling and difficult. Besides these general warnings, there is also the issue of time consume that has to be addressed. Imagine you want to analyse a time series that originates from a rather unknown system, which is usually the case. Then you would not know if the underlying dynamics that produced the time series had two or twenty degrees of freedom. It is easy to verify that the time required to check all possibilities that might yield a proper embedding with respect to various τ and m is very long. This being said, let it be a good motivation to study the mutual information method and the false nearest neighbour method, which enable us to efficiently determine proper values of the embedding delay τ and embedding dimension m.

We start with the mutual information method. A suitable embedding delay τ has to fulfil two criteria. First, τ has to be large enough so that the information we get from measuring the value of variable x variable at time $t + \tau$ is relevant and significantly different from the information we already have by knowing the value of the measured variable at time t. Only then it will be possible to gather enough information about all other variables that influence the value of the measured variable to successfully reconstruct the whole phase space with a reasonable choice of m. Note here that generally a shorter embedding delay can always be compensated with a larger embedding dimension. This is also the reason why the original embedding theorem is formulated with respect to m, and says basically nothing about τ. Second, τ should not be larger than the typical time in which the system looses memory of its initial state. If τ would be chosen larger, the reconstructed phase space would look more or less random since it would consist of uncorrelated points, as in Fig. 1c. The latter condition is particularly important for chaotic systems which are intrinsically unpredictable and, hence, loose memory of the initial state as time progresses. This second demand has led to suggestions that a proper embedding delay could be estimated from the autocorrelation function defined as

$$a(\tau) = \frac{1}{T+1} \sum_{t=0}^{T} x_t x_{t+\tau} \,, \tag{5}$$

where the optimal τ would be determined by the time the autocorrelation function first decreases below zero or decays to 1/e. For nearly regular time series, this is a good thumb rule, whereas for chaotic time series, it might lead to spurious results since it based solely on linear statistic and doesn't take into account nonlinear correlations.

The cure for this deficiency was introduced by Fraser and Swinney [8], who proposed to use the first minimum of the mutual information between x_t and $x_{t+\tau}$ as the optimal embedding delay. The mutual information between x_t and $x_{t+\tau}$ quantifies the amount of information we have about the state $x_{t+\tau}$ presuming we know the state x_t. Given a time series of the form $\{x_0, x_1, x_2, \dots, x_t, \dots, x_T\}$, one first has to find the maximum (x_{max}) and the minimum (x_{min}) of the sequence. The absolute value of their difference $|x_{max} - x_{min}|$ then has to be partitioned into j equally sized intervals, where j should be a large enough integer number. Finally, one calculates the expression

$$I(\tau) = \sum_{h=1}^{j} \sum_{k=1}^{j} P_{h,k}(\tau) \ln P_{h,k}(\tau) - 2 \sum_{h=1}^{j} P_h \ln P_h \,, \tag{6}$$

where P_h and P_k denote the probabilities that the variable assumes a value inside the h-th and k-th bin, respectively, and $P_{h,k}(\tau)$ is the joint probability that x_t is in bin h and $x_{t+\tau}$ is in bin k. As long as the partitioning of the whole interval occupied by the data is fine enough, i.e. j is chosen large enough, the value of the mutual information does not explicitly depend on the bin size. While it has often been shown that the first minimum of $I(\tau)$ really yields the optimal embedding delay, the proof of this has a more intuitive, or shall we rather say empiric, background. It is often said that at the embedding delay where $I(\tau)$ has the first local minimum, $x_{t+\tau}$ adds the largest amount of information to the information we already have from knowing x_t, without completely losing the correlation between them. Perhaps a more convincing evidence of this being true can be found in the very nice article by Shaw [21], who is, according to Fraser and Swinney, the idea holder of this reasoning. However, a formal mathematical proof is lacking. Kantz and Schreiber [6] also report that in fact there is no theoretical reason why there should even exist a minimum of the mutual information. Nevertheless, this should not undermine your trustworthiness in the presented method, since it has often proved reliable and well suited for the appointed task. At most, you should be careful and not take the method completely for granted if further applications with the obtained embedding delay yield somewhat doubtful results.

Finally, let us examine the results obtained with the autocorrelation function as well as the mutual information method. Both results are presented in Fig. 2. The supposed optimal embedding delay obtained with the autocorrelation function equals $\tau = 34$, whereas the mutual information method yields $\tau = 17$ time steps. The two values differ by a factor of two, whereas the optimal one is the latter, as can be seen in Fig. 1b. Since the phase space for $\tau = 34$ is not presented, you should perhaps consider it as an exercise to plot the attractor and find the differences between the two embeddings. The presented results in Fig. 2 can be easily reproduced with our program *mutual.exe*, which can be downloaded from our web page [20]. The program has only two crucial parameters, which are the number of bins j (in our case 50) and the maximal embedding delay τ (in our case 500). All calculated results are displayed graphically. Some further instructions for the usage of our tool kit are given in the Appendix, whereas a more detailed description can be found at the download site.

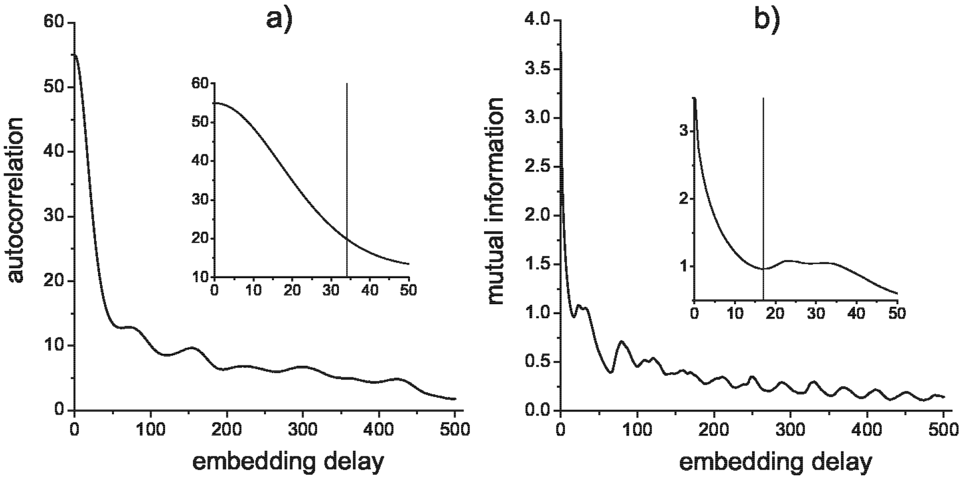

Fig. 2. Determination of the proper embedding delay. a) The autocorrelation decays to $1/e$ at $\tau = 34$. b) The mutual information has the first minimum at $\tau = 17$.

Let us now turn to establishing a proper embedding dimension m for the examined time series by studying the false nearest neighbour method. The false nearest neighbour method was introduced by Kennel et al. [9] as an efficient tool for determining the *minimal* required embedding dimension m in order to fully resolve the complex structure of the attractor. Again note that the embedding theorem by Takens [18] guarantees a proper embedding for all large enough m, i.e. that is also for those that are larger than the *minimal* required embedding dimension. In this sense, the method can be seen as an optimisation procedure yielding just the minimal required m. The method relies on the assumption that an attractor of a deterministic system folds and unfolds smoothly with no sudden irregularities in its structure, like previously described in Figs. 1a and c. By exploiting this assumption, we must come to the conclusion that two points that are close in the reconstructed embedding space have to stay sufficiently close also during forward iteration. If this criterion is met, then under some sufficiently short forward iteration, originally proposed to equal the embedding delay, the distance between two points $\mathbf{p}(i)$ and $\mathbf{p}(t)$ of the reconstructed attractor, which are initially only a small ϵ apart, cannot grow further as $R_{tr}\epsilon$, where R_{tr} is a given constant (see below). However, if a t-th point has a close neighbour that doesn't fulfil this criterion, then this t-th point is marked as having a false nearest neighbour. We have to minimize the fraction of points having a false nearest neighbour by choosing a sufficiently large m. As already elucidated above, if m is chosen too small, it will be impossible to gather enough information about all other variables that influence the value of the measured variable to successfully reconstruct the whole phase space. From the geometrical point of view, this means that two points of the attractor might solely appear to be close, whereas under forward iteration, they are mapped randomly due to projection effects. The random mapping occurs because the whole attractor is projected onto a hyperplane that has a smaller dimensionality than the actual phase space and so the distances between points become distorted.

In order to calculate the fraction of false nearest neighbours, the following algorithm is used. Given a point $\mathbf{p}(t)$ in the m-dimensional embedding space, one first has to find a neighbour $\mathbf{p}(i)$, so that $||\mathbf{p}(i) - \mathbf{p}(t)|| < \epsilon$, where $||\dots||$ is the square norm and ϵ is a small constant, usually not larger than the standard deviation of data [see Eq. (9) below]. We then calculate the normalized distance R_i between the $(m+1)^{st}$ embedding coordinate of points $\mathbf{p}(t)$ and $\mathbf{p}(i)$ according to the equation

$$R_i = \frac{|x_{i+m\tau} - x_{t+m\tau}|}{||\mathbf{p}(i) - \mathbf{p}(t)||}\,. \tag{7}$$

If R_i is larger than a given threshold R_{tr}, then $\mathbf{p}(t)$ is marked as having a false nearest neighbour. Equation (7) has to be calculated for the whole time series and for various $m = 1, 2, \dots$ until the fraction of points for which $R_i > R_{tr}$ is negligible. According to Kennel et al. [9], $R_{tr} = 10$ has proven to be a good choice for most data sets, but a formal mathematical proof for this conclusion is not known. Therefore, if for $R_{tr} = 10$ the method yields non-convincing results, it is advisable to repeat the calculations for various R_{tr}. Before we finally turn to the obtained results, let us mention that by introducing some additional criteria to the described procedure, the false nearest neighbour method can also be used to determine the presence of determinism in a time series. However, since there exist some conceptually simpler methods to be introduced below, we just advise the reader to the relevant article by Hegger and Kantz [22], and use the method here solely for determining the minimal required embedding dimension.

The results obtained with the false nearest neighbour method are presented in Fig. 3. It can be well observed that the fraction of false nearest neighbours (fnn) convincingly drops to zero for $m = 3$. Thereby, this result is in excellent agreement with the dimensionality of the dynamical system (Lorenz equations) that produced the time series. Again, this result can be easily reproduced with our program *fnn.exe*, which can be downloaded from our web page [20]. Parameter values

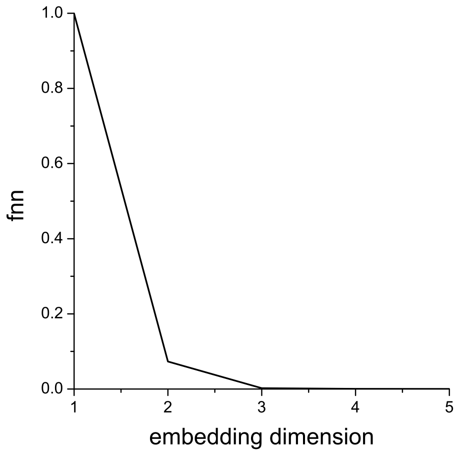

Fig. 3. Determination of the minimal required embedding dimension. The fraction of false nearest neighbours (fnn) drops convincingly to zero at $m = 3$.

that have to be provided are the minimal and the maximal embedding dimension for which the fraction of false nearest neighbours is to be determined ($m = 1, \ldots 5$), the previously obtained embedding delay ($\tau = 17$), the initial ϵ, a factor for increasing the initial ϵ (the latter is increased until ϵ reaches the standard deviation of data), the threshold R_{tr}, and the percent of data that is allowed to be wasted. The last parameter is useful when data is sparse so that there exists a possibility that not all phase space points will have a neighbour inside the maximally allowed ϵ. Besides being displayed graphically, the results are also stored in the file *fnn.dat*, which consists of two ASCII columns. The first column is the embedding dimension and the second column is the pertaining fraction of false nearest neighbours. Additionally, the program displays the standard deviation of data to allow an easier estimation of the initial ϵ (in our case set to 0.05) as well as the factor for increasing the initial ϵ (1.41 in our case).

By now we have equipped ourselves with the knowledge that is required to successfully reconstruct the embedding space from an observed variable. This is a very important task since basically all methods of nonlinear time series analysis require this step to be accomplished successfully in order to yield meaningful results. No exceptions thereby are also the stationarity and the determinism tests introduced below. Recall that stationarity and determinism are important properties of a time series that to some extent have to be present in order to guarantee relevancy of invariant quantities, such as the maximal Lyapunov exponent, that can be extracted from the data.

Let us now describe a stationarity test, which was originally proposed by Schreiber [23], in order to determine if the studied time series originated from a stationary process. In the Introduction, we demanded that every time series must be obtained from a system whose parameters are constant during measurements. Let us think about how a parameter that changes during the experiment could affect the outcome of the measurements. The simplest yet completely uninteresting possibility is that parameter variations are so small that they don't affect the measurement. A more interesting and far more likely case would be that different parts of the time series have different characteristics such as the mean

$$\langle x \rangle_P = \frac{1}{P+1} \sum_{t=0}^{P} x_t \,, \tag{8}$$

or the standard data deviation

$$\sigma_P = \sqrt{\frac{1}{P} \sum_{t=0}^{P} (x_t - \langle x \rangle_P)^2} \,. \tag{9}$$

Note that Eqs. (8) and (9) are formulated with respect to P, which is the number of data points inside a particular time series segment. Sometimes quantities $\langle x \rangle_P$ and σ_P are also refereed to as the *running* mean and standard deviation, respectively, since they are not calculated on the whole data set at once, but pertain only to a particular segment of the time series (for example to the first, the second...1000 data points). A time series is considered to originate from a stationary process if statistical fluctuations of $\langle x \rangle_P$ and σ_P are negligible for various non-overlapping data segments. However, this simple approach for testing the stationarity of data suffers from the same drawbacks as the autocorrelation function, since it is also based solely on linear statistic. Therefore, we have to come up with a suitable nonlinear statistic that would enable us to compare properties in one part of the time series with properties in other parts, thus providing an appropriate tool for testing the stationarity of data.

To this purpose, Schreiber [23] proposed the usage of the so-called cross-prediction error statistic. In order to better understand the elaborate sounding statistic, let us consider the meaning of words "prediction error" and "cross-prediction" separately. First, let us turn to the prediction error. At this point, we first come across the word prediction. While the word is pretty self-explanatory, let us nevertheless elucidate its meaning in the context of nonlinear-time series analysis. A prediction is usually a statement about things that will happen in the future based on the knowledge we have about events that happened in the past. In the language of nonlinear time series analysis, this means that we consider some data points up to time t in order to predict the value of an unknown data point at time $t + \Delta t$, where Δt is a small time interval usually in the order of magnitude of couple of time steps. However, we can still make the same prediction even if we already know the value of x at the time $t + \Delta t$ ($x_{t+\Delta t}$). In particular, this test is done if we want to evaluate the successfulness of our prediction algorithm by calculating the prediction error. If we denote the predicted value of $x_{t+\Delta t}$ by $\tilde{x}_{t+\Delta t}$, then we can calculate the average prediction error according to the equation

$$\delta = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (\tilde{x}_{t+\Delta t} - x_{t+\Delta t})^2} \,, \tag{10}$$

where N is the number of trials we have made, i.e. for how many *different* points x_t we have predicted the value $\tilde{x}_{t+\Delta t}$ and compared it with the true value $x_{t+\Delta t}$. Note that the counter k in Eq. (10) does not directly pertain to the time of variable x, but solely counts the number of trials for different x_t. Since we now know how to estimate the error of a prediction, it remains of interest to describe how to actually make one.

The most common and simple prediction of an event one can make is to consider similar events that have happened in the past, and from that knowledge deduce the most likely event that will happen in the future. For example, if you are not living in a desert, you have probably experienced several times that thick dark clouds usually bring along rain. So if today you see thick dark clouds in the sky you immediately assume, based on the knowledge you have from the past, that it will probably rain in the near future. In order to integrate this basic idea into the concept of nonlinear time series analysis, we first have to think about how to determine events that we call similar. The answer is perhaps far more simple than you imagine. If we consider a point $\mathbf{p}(t)$ in the reconstructed embedding space as an event, than similar events will simply be points that are close to this particular point. By close, we mean closer than some ϵ, which is in the order of magnitude of the data resolution, and certainly not larger than the standard data deviation. How can we then estimate a future value of x_t? The answer is simply to calculate the average value of all $x_{i+\Delta t}$, which pertain to all points $\mathbf{p}(i)$ that are less than ϵ apart from $\mathbf{p}(t)$. If we denote by $|\Theta_\epsilon|$ the number of points $\mathbf{p}(i)$ that satisfy the relation $||\mathbf{p}(i) - \mathbf{p}(t)|| < \epsilon$, then the prediction $\tilde{x}_{t+\Delta t}$ of x_t can finally be calculated using the equation

$$\tilde{x}_{t+\Delta t} = \frac{1}{|\Theta_\epsilon|} \sum_{k=1}^{|\Theta_\epsilon|} x_{i+\Delta t} \,, \tag{11}$$

where the counter k, as in Eq. (10), doesn't pertain to the time of variable x but solely counts the number of all found neighbours. The above-described prediction algorithm is described in the book by Kantz and Schreiber [6], while the original idea allegedly belongs to Pikovksy [24]. In general, there exists a minimal number of neighbours $|\Theta_\epsilon|$ that are necessary to make a relevant prediction. If, for example, $|\Theta_\epsilon|$ for a given point $\mathbf{p}(t)$ is less than 10, then it is safe to say that the prediction will be inaccurate, and thus the value $\tilde{x}_{t+\Delta t}$ should not be considered in Eq. (10) when calculating the average prediction error. If there exist many data points that don't have enough neighbours inside ϵ that is smaller than the standard data deviation, then you cannot use this statistic on your data set. Alternatively, if not enough neighbours for a particular point are found, you can simply calculate the average value of the variable in the data set according to Eq. (8), and use it as the best possible prediction for such points. With these remarks we conclude the description of how to make a prediction and evaluate the pertaining error, and devote ourselves to explaining the second part of our statistic that we plan to use for stationarity testing, namely the "cross-prediction".

First, let us briefly recall what we intend to do. We plan to use the cross-prediction error in order to reveal possible differences between various non-overlapping segments of the time series, to test if the data originate from a stationary process. With this in mind, the main idea to which the word "cross-prediction" pertains should be clear. It is simply to use one segment of the data (say the first 1000 points) to make predictions in another segment of the data (for example the second 1000 points). In terms of variables we have introduced above, this means that for every point $\mathbf{p}(t)$ in a particular data segment, we have to find close neighbours $\mathbf{p}(i)$ in another non-overlapping data segment in order to predict the future value of x_t. Thereby, we check if the dynamics that produced the second data segment is similar to the one that yielded the first data segment. Obviously, this will be the case only if the whole data set originates from the same dynamics. In terms of stationarity, this means that parameters during measurements, i.e. while obtaining the first, second, third... data segment, remained unaltered. Furthermore, our statistic rigorously checks also the second notion of stationarity, which is sufficient sampling of data. This condition for stationarity is simply fulfilled if a data segment can provide enough neighbours for a point in another data segment to make a proper prediction.

At this point, we have all in place to summarize the algorithm for testing the stationarity of a data set as proposed by Schreiber [23]. First, the time series has to be partitioned into equally sized non-overlapping segments of sufficient length. Then, for each point of the segment j, we perform predictions according to Eq. (11), however, by searching for neighbours in a distinct segment i. Subsequently, we evaluate the accuracy of obtained predictions by calculating the average prediction error δ according to Eq. (10), where N now counts all points that are inside segment j. At this stage, it is convenient to replace the symbol δ by the symbol δ_{ji}, to indicate on which data segments predictions were performed (j), and which data set provided the neighbours (i). Finally, this procedure has to be repeated for all combinations of j and i. If a specific combination of j and i yields an error δ_{ji} that is significantly larger than the average, then the dynamics that produced the segment j is obviously not the same as the one that produced i. The second possibility is that the observed phenomenon was not sufficiently sampled while obtaining i, thus providing an insufficient amount of neighbours to make a good prediction for points in the segment j. In both cases, the resulting exceptionally high prediction error δ_{ji} is a clear indicator that the stationarity requirements in the examined time series are not fulfilled. Especially in cases where i is substantially different (substantially in this case depends on the number of data segments) from j, the cross-prediction error is expected to be maximal, since in this case the largest temporal separation between x_t and the neighbours constitutes the largest probability that the dynamics during this time has changed. The final remark concerns the special case when $i = j$. In such cases, the cross-prediction error is simply a prediction error that is expected to be minimal since x_t and the neighbours pertain to the same data segment, and thus the possibility of an altered dynamics is small.

Results obtained with the described algorithm are shown in Fig. 4. We have calculated the cross-prediction error for two time series, each occupying 40000 points that were split into 40 segments of 1000 points. In both cases we used the embedding parameters obtained with the mutual information method ($\tau = 17$) and the false nearest neighbour method ($m = 3$). The results presented in Fig. 4a pertain to the original time series ($\sigma = 10, r = 25, b = 8/3$) used also in all previous calculations, whereas the results presented in Fig. 4b were obtained by using a time series that was generated with the Lorenz equations with a variable parameter r. In particular, every 1000 integration steps r was set to $r+1$. Thereby, we simulated a non-stationary process to enable a better evaluation of the method. As expected, the cross-prediction error is in both cases minimal when $i = j$ (diagonal). However, while for the original time series δ_{ji} remains equally low basically for all cross-predictions, the error increase for the non-stationary time series is clearly evident at off-diagonal entries. In particular, the first half of the time series ($i < 20$) is a terrible source of useful neighbours for the second half of the time series ($j > 20$), whereas the error increase is also evident, but in a more moderate manner. The presented results clearly confirm that the original time series originates from a stationary process, and thus qualifies for further analysis. As in previous cases, this result can be easily reproduced with our program *stationarity.exe*, which can be downloaded from our web page [20]. Crucial parameter values that have to be provided are the previously obtained embedding delay and dimension, the number of points in one segment (in our case 1000), the initial ϵ (usually set to 1/4 of standard data deviation), a factor for increasing the initial ϵ (1.41 in our case), and the number of time steps for prediction (by default 1). The obtained colour map is displayed graphically, so no additional programs are required.

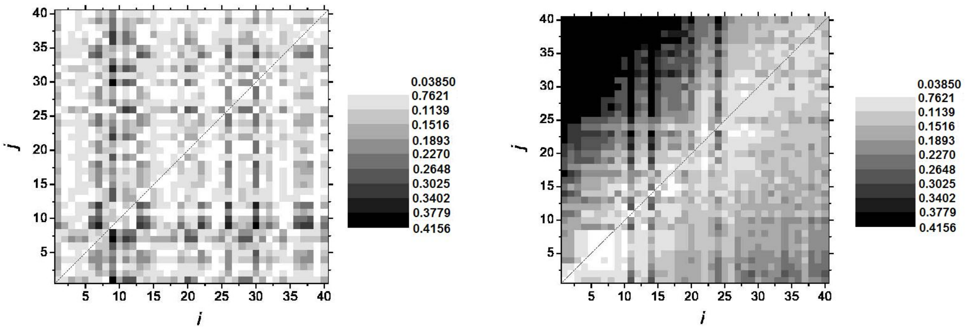

Fig. 4. Stationarity test. a) *Cross-prediction errors obtained for the original time series.* b) *Cross-prediction errors obtained for the time series calculated with variable parameters during recording.*

Before we can definitely qualify our time series as suitable for further analyses, there is still one last test we have to make. That is the determinism test [25 – 27]. What properties has a deterministic time series in comparison to a stochastic one that would allow us to make the distinction? To answer this question consider the fact that a deterministic time series always originates from a deterministic process, which in turn can always be described by a set of more or less complex first-order ordinary differential equations. The relevant consequence of this fact, which follows from the mathematical theory of ordinary differential equations, is that solutions of such systems exist and are unique. Uniqueness of solutions is the property on which we will build up our determinism test. If a system is described by a set of ordinary differential equations, its vector field can be drawn easily. The length as well as the rotation of each vector in every point of the phase space is uniquely determined with the differential equations. However, in case we are faced solely with a time series, the differential equations are obviously not available. Consequently, the uniqueness of solutions of the system that produced the time series cannot be tested directly. What we need is a method that would allow us to construct the vector field of the system directly from the time series, and subsequently test if it assures uniqueness of solutions in the phase space. This awareness led Kaplan and Glass [25] to a beautiful and effective determinism test, which we are going to describe next.

The first step towards a successful realization of the test is, as so often, to construct a good embedding space from the observed variable. Thereby, we obtain the path of the trajectory in the phase space. To construct an approximate vector field of the system, the phase space has to be coarse grained into equally sized boxes with the same dimension as the embedding space. To each box that is occupied by the trajectory, a vector is assigned, which will finally be our approximation for the vector field. The vector pertaining to a particular box is obtained as follows. Each pass i of the trajectory through the k-th box generates a unit (the fact that it is *unit* is crucial) vector, which we will denote as $\mathbf{e}_i$, whose direction is determined by the phase space point where the trajectory first enters the box and the phase space point where the trajectory leaves the box. In fact, this is the average direction of the trajectory through the box during a particular pass. The approximation for the vector field $\mathbf{V}_k$ in the k-th box of the phase space is now simply the average vector of all passes obtained according to the equation

$$\mathbf{V}_k = \frac{1}{P_k} \sum_{i=1}^{P_k} \mathbf{e}_i \,, \tag{12}$$

where P_k is the number of all passes through the k-th box. Completing this task for all occupied boxes gives us a *directional* approximation for the vector field of the system. Note that the word directional is stressed since the described method provides no information about how fast the trajectory moves through particular boxes. Therefore, we cannot say anything about the absolute lengths of the obtained vectors. The absolute magnitude of the vector field is, however, not important for the determinism test. What is important is the fact that, if the time series originated from a deterministic system, and the coarse grained partitioning is fine enough, the obtained vector field should consist solely of vectors that have unit length (remember that each $\mathbf{e}_i$ is a unit vector). This follows directly from the fact that we demand uniqueness of solutions in the phase space. If solutions in the phase space are to be unique, then the unit vectors inside a particular box must all point in the same direction. In other words, the trajectories inside each box may not cross, since that would violate the uniqueness condition at each crossing. Note that each crossing decreases the size of the average vector $\mathbf{V}_k$. For example, if the crossing of two trajectories inside the k-th box would occur at right angle, then the size of $\mathbf{V}_k$ would be, according to Pythagoras, $\sqrt{2}/2 \approx 0.707 < 1$. Hence, if the system is deterministic, i.e. no trajectory crossings inside a particular box occur, each average resultant vector obtained according to Eq. (12) will be exactly of unit length. Accordingly, the average length of all resultant vectors $\mathbf{V}_k$ will be exactly 1, while for a system with a stochastic component this value will be substantially smaller than 1. In the original paper [25], a definite measure for determinism (κ) was proposed to be a weighted average of $\mathbf{V}_k$ with respect to the average displacement per step, R_k^m, of a random walk, which can be calculated according to the equation

$$\kappa = \frac{1}{A} \sum_{k=1}^{A} \frac{(\mathbf{V}_k)^2 - (R_k^m)^2}{1 - (R_k^m)^2} \,, \qquad ((13)$$

where A is the total number of occupied boxes, and R_k^m is obtained according to the equation

$$R_k^m = c_m P_k^{-1/2} \,, \qquad (14)$$

where c_m is a constant that depends on the embedding dimension and equals $\pi^{1/2}/2$, $4/(6\pi)^{1/2}$, $3\pi^{1/2}/32^{1/2}$ for $m = 2, 3, 4$, respectively. As already mentioned, the determinism factor for a deterministic system is $\kappa = 1$, while due to the weighted average with respect to R_k^m, $\kappa = 0$ for a random walk.

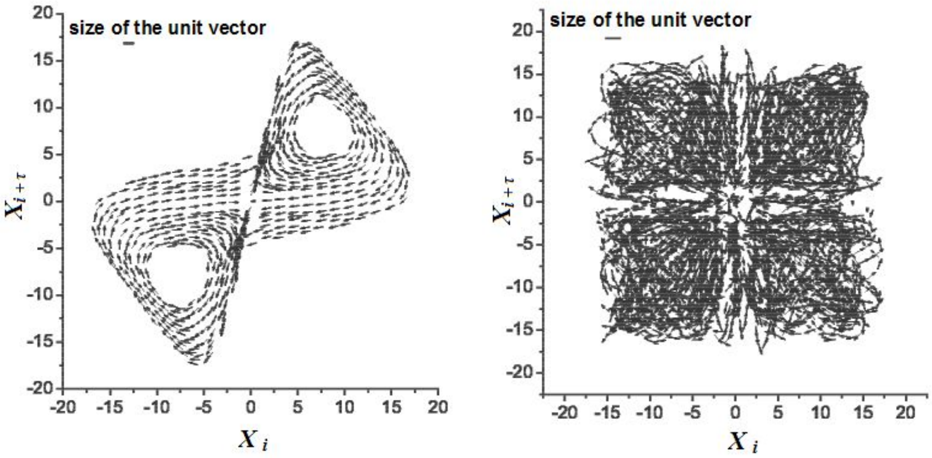

Fig. 5. Determinism test. a) The approximated vector field for the embedding space reconstructed with $\tau = 17$ and $m = 3$. The pertaining determinism factor is $\kappa = 0.99$. b) The approximated vector field for the embedding space reconstructed with $\tau = 500$ and $m = 3$. The pertaining determinism factor is $\kappa = 0.31$.

Results obtained with the described method are presented in Fig. 5. We have performed the determinism test on the original time series for two different embedding delays to better evaluate the method, and in particular, to show how a large embedding delay destroys determinism by introducing uncorrelated points as close neighbours in the phase space. For both calculations, the three-dimensional embedding space was coarse grained into a $24 \times 24 \times 24$ grid. In Fig. 5a, the approximated vector field for the embedding space reconstructed with the optimal embedding delay, obtained with the mutual information method ($\tau = 17$), is presented. The pertaining determinism factor is $\kappa = 0.99$. It can be well observed that basically all vectors are of unit length, and indeed it would be difficult to distinguish between the approximated and the actual vector field if the latter existed (for the embedding space). You may also compare the vector field with the phase space presented in Fig. 1b, since they are both obtained with exactly the same embedding parameters. In Fig. 5b, however, the situation is significantly different. The approximated vector field pertains to the embedding space obtained with $\tau = 500$. For that delay, the autocorrelation function as well as the mutual information are nearly zero (see Fig. 2), which means that values of the time series that are separated by such a temporal gap are completely uncorrelated. Thus, the embedding at this delay is quite similar to a reconstructed phase space one would obtain with a completely stochastic time series. Accordingly, only few vectors are of unit length, and perhaps the vector field is indeed best described with the words used by the authors of the original article: "it is a spaghetti mess" [25]. The pertaining determinism factor is, not surprisingly, $\kappa = 0.31$ (very low). As always, the presented results can be reproduced in an easy manner with our program *determinism.exe*, which can be downloaded from our web page [20].

At last, we have successfully established that the studied time series originates from a stationary deterministic process. Prior to that, we have explained how to obtain optimal embedding parameters, and so it is now safe to say that we are equipped with enough skills to take on nearly any challenge of nonlinear time series analysis. Of course there is still a lot to learn, and tedious work lies ahead of us trying to understand and implement new, perhaps more sophisticated, methods. Nevertheless, so far we are on the right way, and a good start is always important. At the end, let us calculate the maximal Lyapunov exponent of the examined data set to confirm the presence of deterministic chaos in the underlying system.

Lyapunov exponents [28 – 30] determine the rate of divergence or convergence of initially nearby trajectories in phase space. This is true for the phase space obtained from differential equations, as well as for the reconstructed phase space obtained from a single variable. Generally, an m-dimensional phase space is characterized by m different Lyapunov exponents, which we will denote as Λ_i, where $i = 1, 2, \ldots, m$. They can be ordered from the largest to the smallest to form the Lyapunov exponent spectrum $\Lambda_1, \Lambda_2, \ldots, \Lambda_m$. It is a well-established fact that if one or more of these exponents is larger than zero, the system is chaotic [28]. If this is the case, two arbitrary close trajectories of the system will diverge apart exponentially, eventually ending up in completely different phase space areas as time progresses. This, so-called, extreme sensitivity to changes in initial conditions is the hallmark of chaos. It is this extreme sensitivity on the initial state that makes chaotic systems *inherently* unpredictable. If, for example, we measure the temperature only with 0.1% inaccuracy, this little measurement error will eventually lead to a completely wrong prediction, even if we would have a complete set of differential equations that would absolutely describe temporal temperature variations on Earth. This is certainly an astonishing fact. At this point, however, let us concentrate on the fact that the maximal Lyapunov exponent Λ_1 uniquely determines whether the system is chaotic or not. For our purposes it will, therefore, be sufficient to concentrate only on the calculation of the maximal Lyapunov exponent. There exist several effective and robust algorithms [31,32] that have been developed for this task. However, we decided to trade some of this efficacy and robustness for a conceptually simpler and direct approach. Therefore, we will describe the algorithm developed by Wolf et al. [33] that implements the theory in a very direct fashion.

As already noted, the Lyapunov exponents determine the rate of divergence or convergence of initially nearby trajectories in phase space. So what we need to do in order to calculate the maximal Lyapunov exponent is the following. For a point of the embedding space $\mathbf{p}(t)$ find a near neighbour $\mathbf{p}(i)$, which satisfies the relation $||\mathbf{p}(i) - \mathbf{p}(t)|| \leq \epsilon$. Then iterate both points forward in time for a fixed evolution time ν, which should be a couple of times larger than the embedding delay τ, but not much larger than $m\tau$. If the system is chaotic, the distance after the evolved time $||\mathbf{p}(i+\nu) - \mathbf{p}(t+\nu)|| = \epsilon_\nu$, will be larger than the initial ϵ, while in case of regular behaviour $\epsilon \approx \epsilon_\nu$. After each evolution ν, a so-called replacement step is attempted in which we look for a new point $\mathbf{p}(j)$ in the embedding space, whose distance to the evolved point $\mathbf{p}(t+\nu)$ should be small (ϵ), under the constraint that the angular separation between the vectors constituted by the points $\mathbf{p}(t+\nu)$ and $\mathbf{p}(i+\nu)$, and $\mathbf{p}(t+\nu)$ and $\mathbf{p}(j)$ is small. This procedure is repeated until the initial point of the trajectory reaches the last one. Finally, the maximal Lyapunov exponent can be calculated according to the equation

$$\Lambda_1 = \frac{1}{M\nu} \sum_{i=1}^{M} \ln \frac{\epsilon_\nu}{\epsilon} \,, \tag{15}$$

where M is the total number of replacement steps. The successive replacement steps are crucial for a correct estimation of the maximal Lyapunov exponent. If the embedding space is properly constructed and densely populated with points, the algorithm performs extremely well. However, if data is sparse, it might happen that we have to except a rather large initial distance or angular separation in a replacement step. If this is the case, it cannot be argued that the obtained maximal Lyapunov exponent is always precisely accurate. Nevertheless, if the algorithm is implemented with a variable acceptable initial distance and angular separation, then the obtained result is always accurate enough at least for a "yes/no chaos" assessment.

The result obtained with the above-described algorithm for the optimal embedding parameters is presented in Fig. 6. It can be well observed that the maximal Lyapunov exponent converges extremely well to $\Lambda_1 = 0.88$, which is in good agreement with $\Lambda_1 = 0.82$ that can be calculated with the help of differential equations. More importantly, however, the obtained positive maximal Lyapunov exponent is a clear indicator that the studied time series originated from a chaotic system. Together with the results obtained from the stationarity and determinism test, it is safe to say that deterministic chaos is inherently present in the studied time series, and thus that the underlying system [1] is, for the studied parameter values, deterministically chaotic.

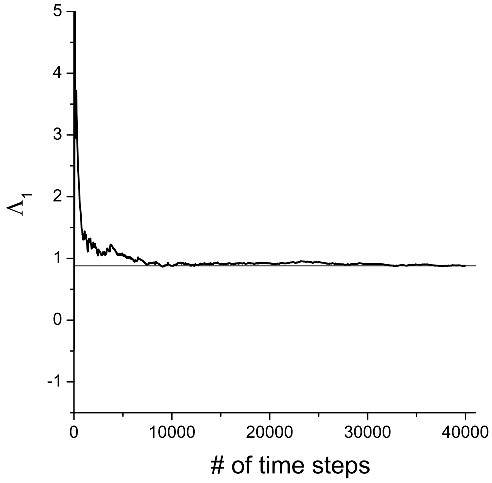

Fig. 6. Calculation of the maximal Lyapunov exponent for the embedding space reconstructed with $\tau = 17$ and $m = 3$. The maximal Lyapunov exponent converges well to $\Lambda_1 = 0.88$.

3. *Discussion*

In the present paper, we describe essential nonlinear time series analysis methods that are required to establish the presence of chaos in an observed time series. We emphasize that chaos in a time series cannot be positively established, without first checking whether the data set originated from a stationary and deterministic process. Only if both these criteria are met, the obtained invariant quantities, such as the maximal Lyapunov exponent, truly characterize the system as we believe they do. For example, a positive maximal Lyapunov exponent can then be considered as a convincing evidence for the presence of chaos in the studied time series. If, however, stationarity and determinism are not tested for, it cannot be claimed that the exponential divergence of nearby trajectories originates from the system dynamics, for it might be solely a consequence of stochastic inputs or variable parameters during data acquisition.

The knowledge one obtains when mastering the above-described methods presents a good starting point for performing further analysis, such as calculation of dimensions [10 − 12] or noise reduction [13 − 14]. In fact, these tasks would be appropriate subjects for a sequel paper. Currently, these topics are covered in existing monographs on nonlinear time series analysis [5 − 7], and in the original papers. An excellent source of information for these methods is also the TISEAN project [34]. Together with the pertaining paper [35] and the book by the same authors [6], the TISEAN project presents a very valuable source of information as well as useful programs for virtually all topics of nonlinear time series analysis. Therefore, particularly for students that would like to dig deeper into the nonlinear time series analysis, we recommend to exploit the benefits offered by the project.

It is our sincere hope that the reader will find interest and inspiration in the nonlinear time series analysis. Therefore, we have invested a lot of effort in making the presented results as easily reproducible as possible [20]. Our goal was to make the methods accessible to undergraduate students, to whom this article may represent the first contact with the presented theory. The paper is also devoted to teachers who would like to integrate nonlinear time series analysis methods into the physics curriculum.

Appendix

Since it is our sincere hope that the interested reader will find great joy in nonlinear time series analysis, we have developed user-friendly programs that allow a quick and easy reproduction of all presented results in this paper. The whole package, that can be downloaded from our web page [20], consists of six programs (*embedd.exe, mutual.exe, fnn.exe, stationarity.exe, determinism.exe, and lyapmax.exe*) and a sample input file (*ini.dat*). All programs have a graphical interface and display results in forms of drawings or colour maps. In order to function, they require a Windows environment and an input file named *ini.dat* in the working directory. After download, the content of the *package.zip* file should be extracted into an arbitrary (preferably empty) directory. Thereafter, the programs are ready to run via a double-click on the appropriate icon. Initially, a parameter window will appear, which allows you to insert proper parameter values (by default they are set equally to those used in this paper). Once this step is completed, just press the OK button to execute the program. A progress bar will appear, which lets you know how fast the program is running, and when it will eventually finish. After completion results are displayed graphically in a maximized window. To avoid exceptionally high memory allocation and running times, all programs are currently limited to operate maximally on 250000 data points with 10 degrees of freedom. Upon request, we can provide programs that can handle also larger data sets. Finally, we strongly advise the reader to read the manual pertaining to the programs on our web page, where more detailed instructions are given.

References

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

[2] H. G. Schuster, *Deterministic chaos*, VCH, Weinheim (1989).

[3] E. Ott, *Chaos in dynamical systems*, Cambridge University Press, Cambridge (1993).

[4] S. H. Strogatz, *Nonlinear dynamics and chaos*, Addison-Wesley, Massachusetts (1994).

[5] H. D. I. Abarbanel, *Analysis of observed chaotic data*, Springer, New York (1996).

[6] H. Kantz and T. Schreiber, *Nonlinear time series analysis*, Cambridge University Press, Cambridge (1997).

[7] J. C. Sprott, *Chaos and time-series analysis*, Oxford University Press, Oxford (2003).

[8] A. M. Fraser and H. L. Swinney, Phys. Rev. A **33** (1986) 1134.

[9] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, Phys. Rev. A **45** (1992) 3403.

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

[11] J. Theiler, Phys. Rev. A **34** (1986) 2427.

[12] H. Kantz and T. Schreiber, Chaos **5** (1994) 143.

[13] Section II in *Time series prediction: Forecasting the future and understanding the past*, ed. A. S. Weigend and N. A. Gershenfeld, Santa Fe Institute Studies in the Sciences of Complexity **15** (1994) 173.

[14] T. Schreiber, Phys. Rev. E **47** (1993) 2401.

[15] P. Grassberger, R. Hegger, H. Kantz, C. Schaffrath, and T. Schreiber, Chaos **3** (1993) 127.

[16] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, Physica D **58** (1992) 77.

[17] T. Schreiber and A. Schmitz, Physica D **142** (2000) 346.

[18] F. Takens, in *Detecting strange attractor in turbulence*, ed. D. A. Rand and L. S. Young, Lect. Notes Math. **898** (1981) 366.

[19] T. Sauer, J. A. Yorke, and M. Casdagli, J. Stat. Phys. **65** (1991) 579.

[20] All results presented in this paper can be easily reproduced with our set of programs, which can be downloaded from the web page: http://fizika.tk

[21] R. Shaw, Z. Naturforsch. **36a** (1981) 80.

[22] R. Hegger and H. Kantz, Phys. Rev. E **60** (1999) 4970.

[23] T. Schreiber, Phys. Rev. Lett. **78** (1997) 843.

[24] A. Pikovsky, Sov. J. Commun. Technol. Electron. **31** (1986) 911.

[25] D. T. Kaplan and L. Glass, Phys. Rev. Lett. **68** (1992) 427.

[26] R. Wayland, D. Bromley, D. Pickett, and A. Passamante, Phys. Rev. Lett. **70** (1993) 580.

[27] L. W. Salvino and R. Cawley, Phys. Rev. Lett. **73** (1994) 1091.

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

[29] S. S. Machado, R. W. Rollins, D. T. Jacobs, and J. L. Hartman, Am. J. Phys. **58** (1990) 321.

[30] J. C. Earnshaw and D. Haughey, Am. J. Phys. **61** (1993) 401.

[31] M. T. Rosenstein, J. J. Collins, and C. J. De Luca, Physica D **65** (1993) 117.

[32] H. Kantz, Phys. Lett. A **185** (1994) 77.

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

[34] The official web page of the TISEAN project is www.mpipks-dresden.mpg.de/~tisean.

[35] R. Hegger and H. Kantz, Chaos **9** (1999) 413.

UVOĐENJE ANALIZE NELINEARNIH VREMENSKIH NIZOVA U DODIPLOMSKI STUDIJ

Ovaj smo članak napisali za dodiplomske studente i nastavnike koji se žele upoznati s osnovnim metodama analize nelinernih vremenskih nizova. Postupno proučavamo jednostavan primjer takvog niza i dajemo programe za lako ponavljanje izloženih ishoda računa. Taj je primjer umjetan vremenski niz stvoren Lorenzovim sustavom jednadžbi. Podrobno objašnjavamo metode uzajamnih informacija i krivog najbližeg susjeda, koje se primjenjuju za najbolje nalaženje nakupinskih točaka. Zatim se ispituju stacionarnost i determinizam vremenskih nizova, koji su važna svojstva za ispravno tumačenje nepromjenljivih veličina koje se mogu izvesti iz skupa podataka. Na kraju računamo najveći Ljapunovljev eksponent koji je najvažnija nepromjenljiva veličina za razlikovanje pravilnog i kaotičnog ponašanja niza. Slijedom ovih koraka utvrđujemo uvjerljivo da je Lorenzov sustav kaotičan, izravno iz izvedenog niza, bez upotrebe diferencijalnih jednadžbi. U radu se poklanja velika pažnja naputcima i izravnoj primjeni metoda kako bi i dodiplomski studenti mogli ponoviti izložene račune i tako se potaknuli da bolje upoznaju prikazanu teoriju.