Optimization of Poincaré sections for discriminating between stochastic and deterministic behaviour of dynamical systems

Krzysztof Michalak

Department of Information Technologies, Institute of Business Informatics, Wroclaw University of Economics, Wroclaw, Poland

Abstract

This paper studies the problem of finding optimal parameters for a Poincaré section used for determining the type of behaviour of a time series: a deterministic or stochastic one. To reach that goal optimization algorithms are coupled with the Poincaré & Higuchi (P&H) method, which calculates the Higuchi dimension using points obtained by performing a Poincaré section of a certain attractor. The P&H method generates distinctive patterns that can be used for determining if a given attractor is produced by a deterministic or a stochastic system, but this method is sensitive to the parameters of the Poincaré section. Patterns generated by the P&H method can be characterized using numerical measures which in turn can be used for finding such parameters for the Poincaré section for which the patterns produced by the P&H method are the most prominent. This paper studies several approaches to parametrization of the Poincaré section. Proposed approaches are tested on twelve time series, six produced by deterministic chaotic systems and six generated randomly. The obtained results show, that finding good parameters of the Poincaré section is important for determining the type of behaviour of a time series. Among the tested methods the evolutionary algorithm was able to find the best Poincaré sections for use with the P&H method.

Keywords: Poincaré & Higuchi method, Higuchi dimension, Poincaré section, optimization

1. Introduction

The analysis of the behaviour of nonlinear signals is an area of research that is interesting from both the theoretical and practical perspectives. One of the questions that arise when one studies the behaviour of a nonlinear system is whether the dynamics of the system is governed by a stochastic or a deterministic process. Unfortunately, the complexity of the behaviour of nonlinear systems makes it hard to distinguish the deterministic from the stochastic based on just the observation of the output of the system. A common approach to chaos detection is to calculate invariants such as the correlation dimension [1, 2], Lyapunov exponent [3, 4, 5, 6], entropy [7, 8, 9, 10, 11, 12] and others [13, 14, 15].

Recently, Golestani et al. [16] proposed a method named Poincaré & Higuchi (P&H) which combines the calculation of fractal dimension using the Higuchi method [17, 18, 19, 20] with the Poincaré section technique. The P&H method produces distinctive patterns which can be used to determine if the studied system is a deterministic or a stochastic one. It was applied in studies concerning iris patterns [21] and the analysis of simulated ecosystems [22]. A problem that arises when using the P&H method is that the output of the method depends on the placement of the Poincaré section. This paper attempts to mitigate this issue by employing optimization methods for finding an optimal placement of the Poincaré section for the P&H method.

Email address: krzysztof.michalak@ue.wroc.pl (Krzysztof Michalak)

The rest of this paper is structured as follows. In Sec. 2 the concept of a Poincaré section, the Higuchi dimension and the P&H method are described. Sec. 3 presents five different approaches to positioning the Poincaré section. This section also discusses optimization techniques used in some of these approaches. Sec. 4 presents the results produced by the methods studied in this paper on twelve time series, six produced by deterministic chaotic systems and six generated randomly. Sec. 5 concludes the paper.

2. The Poincaré section, the Higuchi dimension and the P&H method

This section describes the concepts used in the P&H method.

The Poincaré Section

When analyzing dynamical systems one is often interested in computing a Poincaré map T of a given dynamical system. To obtain a Poincaré map we select a subspace (for example a plane in the three-dimensional Euclidean space) and find points at which the orbit of a dynamical system intersects this subspace. The Poincaré map transforms the point at which the first intersection occurs to the point at which the second intersection occurs, then it transforms the second intersection point to the third and so on.

Formally, let Γ be a flow in $\mathbb{R}^d$. The Poincaré map T is obtained by finding an intersection of the flow Γ with a certain subspace Σ of the codimension 1. A method for obtaining the Poincaré map consists of the following steps.

1. A point $P_0 \in \mathbb{R}^d$ and a normal vector $\overrightarrow{n_0} \in \mathbb{R}^d$ are selected, which determine the subspace Σ. Σ is an $n-1$ dimensional subspace of R^d that satisfies $P_0 \in \Sigma$ and $\Sigma \perp \overrightarrow{n_0}$. For example, if $d = 3$ the flow Γ is three-dimensional and the subspace Σ is a plane in $\mathbb{R}^3$ orthogonal to a given vector $\overrightarrow{n_0}$ and containing the point P_0.
2. A set of points $S = \{P_1, P_2, P_3, \dots\}$ is obtained by intersecting the flow Γ with the subspace Σ with the condition that only those intersections are taken into account that occur when the flow Γ goes from one selected side of Σ to the other and not in the opposite direction.
3. The Poincaré map is a discrete function $T : S \to S$ such that $\forall k \geq 1 : T(P_k) = P_{k+1}$.

If Γ is represented as a set of points $x(1), x(2), x(3), \dots \in \mathbb{R}^d$ the Poincaré map of the flow Γ with Σ can be calculated using intersections of line segments $\overline{x(t)x(t+1)}$ with Σ:

$$S_i = \left\{\overline{x(t)x(t+1)} \cap \Sigma\right\} \tag{1}$$

for all such t that:

$$\overline{x(t)x(t+1)} \cap \Sigma \neq \emptyset \wedge \overrightarrow{x(t)x(t+1)} \circ \overrightarrow{n_0} > 0 \,.$$

This paper also studies another approach which, instead of intersections, uses a projection of points onto the subspace Σ with the additional condition that only the points located near Σ should be taken into consideration. The intersection S_s is thus generated as follows:

$$S_s = \left\{x(t) - [(x(t) - P_0) \circ \overrightarrow{n_0}] \cdot \overrightarrow{n_0}\right\} \tag{2}$$

for all such t that:

$$\left|(x(t) - P_0) \circ \overrightarrow{n_0}\right| < \mu \ ,$$

where:

μ - the maximum distance from Σ of points considered for the projection.

The first approach, which produces points according to the equation (1) is named **"intersect"** in the later part of this paper. Because the Poincaré section S_i is determined by the point $P_0 \in \mathbb{R}^d$ and the normal vector $\overrightarrow{n_0} \in \mathbb{R}^d$ the optimization of the Poincaré section is in this case equivalent to finding the best values for P_0 and $\overrightarrow{n_0}$. Therefore, the search space in the "intersect" approach is $\mathbb{R}^{2d}$.

The second approach, which produces points according to the equation (2) is named **"slice"** in the later part of this paper. This approach, apart from the point $P_0 \in \mathbb{R}^d$ and the normal vector $\overrightarrow{n_0} \in \mathbb{R}^d$, also requires a value μ which determines the thickness of the slice. Therefore, the search space in the "slice" approach is $\mathbb{R}^{2d+1}$.

The Higuchi dimension

Higuchi [17, 18] proposed a method of calculating the fractal dimension for a time series $x(1), x(2), x(3), \dots, x(N)$. This method constructs several new time series x_m^k by starting at $x(m)$, $m = 1, 2, 3, \dots, k$ and then selecting every k-th element in the original time series. For every value of k from 1 to a selected maximum time interval k_{max} we calculate the average length of the curves containing points spaced every k samples taken from the original time series. The slope of a linear function between $\ln(k)$ and the logarithm of the length of the curves obtained for each k is used to calculate the Higuchi dimension.

The Higuchi's method works by performing the following steps.

1. A set K of time intervals is chosen. In the original work by Higuchi [17] the values of $k = 1, 2, 3, 4$ and $k = \left\lfloor 2^{(j-1)/4} \right\rfloor$, $j = 11, 12, 13, \dots$ (where $\lfloor\cdot\rfloor$ denotes the floor function) were used. A maximum value for k was set to $k_{max} = 2^{11}$. Thus, the set K used by Higuchi was:
$$K = \left\{1, 2, 3, 4\right\} \cup \left\{\left\lfloor 2^{(j-1)/4} \right\rfloor\right\}_{j=11,12,13,\dots,45} . \tag{3}$$
In the original paper on the P&H method [16] the set $K = \{1, 2, 3, \dots, 20\}$ is used.
2. For $m = 1, 2, 3, \dots, k$ new time series x_m^k are constructed by taking elements from the original time series $x(t)$ in the following manner:
$$x_m^k = \{x(m), x(m+k), x(m+2k), \dots\} \,, \tag{4}$$
where:
$m = 1, 2, 3, \dots, k$.
Thus, we obtain k new time series from the original one, each constructed with a different starting point.
3. For each of the new k time series x_m^k, $m = 1, 2, 3, \dots, k$ the length $L_m(k)$ of the curve connecting the elements of x_m^k is calculated:
$$L_m(k) = \frac{\sum_{i=1} |x(m+ik) - x(m+(i-1)k)|(N-1)}{\left\lfloor \frac{N-m}{k} \right\rfloor k} / k \,. \tag{5}$$
4. The average value $L(k)$ is calculated from $L_m(k)$ for $m = 1, 2, 3, \dots, k$:
$$L(k) = \frac{\sum_{m=1}^{k} L_m(k)}{k} \,. \tag{6}$$
5. The fractal dimension is determined as such value D that $L(k) \propto k^{-D}$. Thus, when $\ln(L(k))$ is plotted against $\ln(k)$ the points should fall on a line with a slope $-D$.

The P&H method

The P&H method uses the Poincaré section and applies most of the steps involved in the calculation of the Higuchi dimension to the time series formed by points $P_i \in S$. There are two important modifications with respect to the procedure used by Higuchi. First, the P&H method uses the normalizing factor of:

$$\frac{N-1}{\left\lfloor \frac{N-m}{k} \right\rfloor k} \tag{7}$$

instead of:

$$\frac{N-1}{\left\lfloor \frac{N-m}{k} \right\rfloor kk} \,. \tag{8}$$

Thus, the equation for $L(k)$ becomes:

$$L_m(k) = \frac{\sum_{i=1} |x(m+ik) - x(m+(i-1)k)|(N-1)}{\left\lfloor \frac{N-m}{k} \right\rfloor k} \,. \tag{9}$$

Second, in the P&H method we do not actually calculate the fractal dimension, so no line fitting is performed for $\ln(L(k))$. Instead, a graph of $\ln(L(k))$ versus k is plotted. The characteristics of this graph can be used for determining if the time series was generated by a deterministic chaotic process or by a

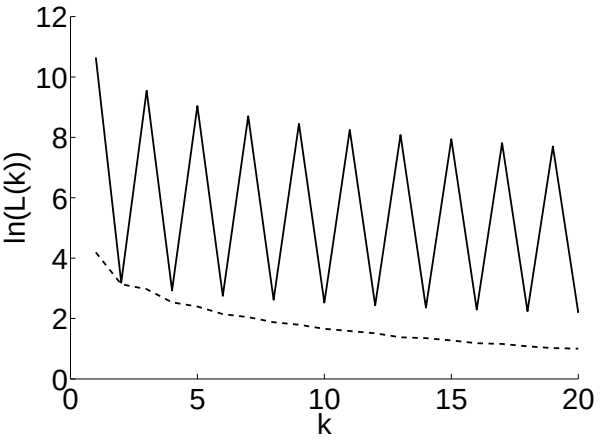


Figure 1: Two different graphs of $\ln(L(k))$ versus k obtained for the same time series generated by the logistic map for $r = 3.8284$ for two different Poincaré sections.

stochastic one. In the former case the line visible in the graph is jagged and has frequent increases and decreases. In the latter case a smooth decreasing curve can be observed. If the dynamical system produces points in $\mathbb{R}^d$, $d > 1$ the P&H method can be applied directly to these points. If a time series of scalar values is only available then this time series is embedded in some $\mathbb{R}^d$ using the Takens theorem [23].

Unfortunately, the P&H method seems to be sensitive to the selection of the surface Σ used for obtaining the Poincaré map. For certain locations of the Poincaré section it is possible to obtain a smooth decreasing curve even for a deterministic time series. In Fig. 1 two plots of $\ln(L(k))$ versus k are shown, both obtained for the same time series generated by the logistic map for $r = 3.8284$. The two graphs were obtained for two different placements of the surface Σ.

Clearly, observing the graph marked by a dashed line could lead to an incorrect conclusion that the studied time series is an output of a stochastic process while in reality it is an output of a deterministic, although chaotic, one. This paper studies several methods of selecting the surface Σ with the aim of obtaining the most prominent variations in the graphs produced by the P&H method. The goal is to provide an automatic procedure for determining if a given time series was produced by a stochastic process or by a deterministic one.

3. Optimization of the Poincaré section

This section presents five methods of choosing the parameters for the Poincaré section technique. The proposed methods are:

- **Perpendicular** in which the section subspace is always perpendicular to one of the axes of the coordinate system.
- **PCA$_{\mathbf{max}}$** in which the vector $\overrightarrow{n_0}$ points in the direction of the largest variance in data established using the Principal Component Analysis (PCA) method.
- **PCA$_{\mathbf{min}}$** in which the vector $\overrightarrow{n_0}$ points in the direction of the smallest variance in data established using the Principal Component Analysis (PCA) method.
- **Gradient** in which the location and orientation of the section is determined using an optimization method based on a gradient descent.
- **EvolutionaryAlgorithm(EA)** in which the location and orientation of the section is determined using an evolutionary algorithm.

The goal is to obtain the most prominent variations in the graphs produced by the P&H method. The advantage of the P&H method is that it produces graphs such as the ones presented in Fig. 1 which can be evaluated numerically. It is thus possible to approach the task of determining the parameters of a good Poincaré section as an optimization problem.

Let $\{y_k = \ln(L(k))\}_{k=1,2,3,\ldots,k_{max}}$ be the ordinates of points produced by the P&H method. To optimize the parameters of a Poincaré section we need a numerical criterion which we could minimize or maximize. One important characteristic of such criterion is that it should allow gradual improvement of the parameters by an optimization algorithm. For this reason a criterion which simply counts increases in the $\{y_k\}$ series (that is situations when $y_{k+1} > y_k$) can be used for classifying the behaviour of a given time series, but is not a good criterion for optimization, because it can only increase (or decrease) in integer increments. A better criterion should change in small increments when small changes are applied to the parameters of the Σ surface. This allows the optimization algorithms to iteratively improve the parameters of the Poincaré section. In preliminary tests several evaluation criteria were tested, and eventually an evaluation criterion was adopted which produced the most distinctive zig-zag patterns for deterministic dynamical systems.

Using the chosen criterion the graphs produced by the P&H method are evaluated in the following way. Let $y'_k = y_{k+1} - y_k$ and $y''_k = y'_{k+1} - y'_k$. Thus, the values y''_k are second-order differences calculated from y_k. In order to obtain high variability of y_k we try to maximize the squared variation of the values of y''_k:

$$\Delta(P_0, \overrightarrow{n_0}) = \sum_{i=1}^{k_{max}-3} (y''_{i+1} - y''_i)^2 \; . \tag{10}$$

Because $P_0, \overrightarrow{n_0} \in \mathbb{R}^d$ and $\Delta(P_0, \overrightarrow{n_0}) \in \mathbb{R}$ the task of determining the values of P_0 and $\overrightarrow{n_0}$ can be treated as an optimization problem of maximizing the value of the function $\Delta : \mathbb{R}^{2d} \to \mathbb{R}$. If the thickness of the slice μ is also considered the function becomes $\Delta : \mathbb{R}^{2d+1} \to \mathbb{R}$.

The five approaches of selecting a good surface Σ for the Poincaré section tested in this paper are as follows.

Perpendicular

An approach in which the surface Σ is perpendicular to one of the axes of the coordinate system which goes through the mean value $m \in \mathbb{R}^d$ of the observed points. Thus, $P_0 = m$ and $\overrightarrow{n_0}$ is a vector with all coordinates equal 0 except one set to 1. This approach is used, among others, in the poincare.exe utility

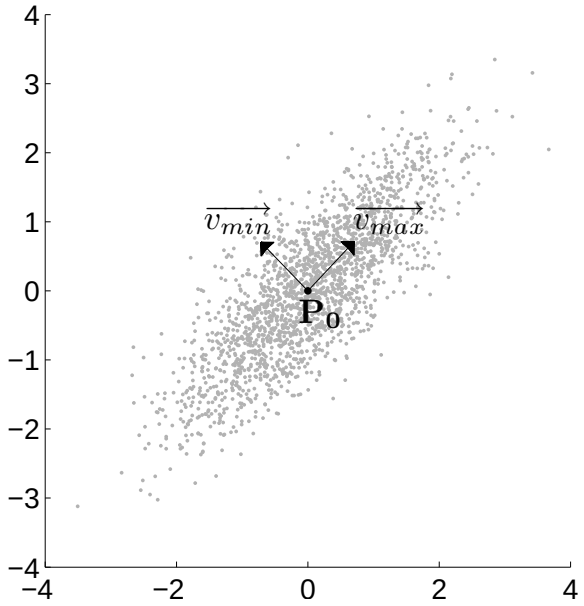


Figure 2: A basis obtained using the Principal Component Analysis (PCA) method for a two-dimensional set of points.

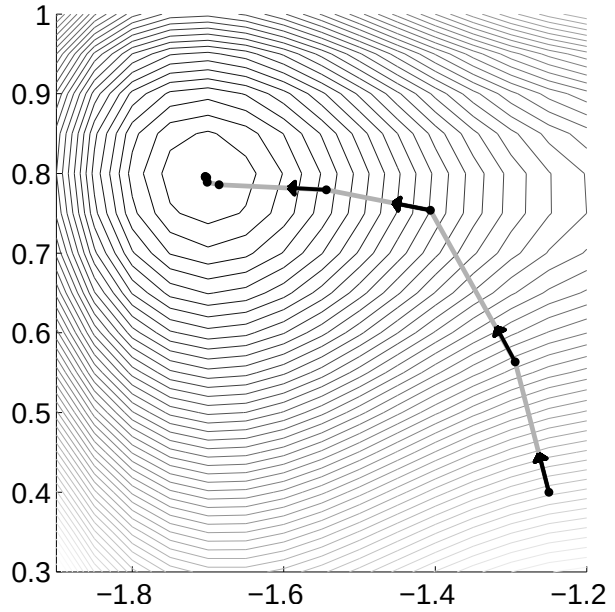


Figure 3: An example of the working of the gradient descent optimization method.

in the TISEAN package [24]. From the d possible choices for the $\overrightarrow{n_0}$ vector the one that maximizes $\Delta(P_0, \overrightarrow{n_0})$ is chosen.

$\mathbf{PCA_{max}}$ and $\mathbf{PCA_{min}}$

In both approaches the Principal Component Analysis method [25] is used. This method calculates an orthonormal basis $\{\overrightarrow{v_i}\}_{i=1,2,3,...,d}$ in $\mathbb{R}^d$. One of the vectors $\overrightarrow{v_1}$ in this basis points in the direction in which the variance of the analyzed sample is the greatest. The second is chosen so that it points in the direction which maximizes the variance in $R^{d-1} \perp \overrightarrow{v_1}$, and so on. The vectors $\overrightarrow{v_i}$ are the eigenvectors of the covariance matrix calculated from the points in the sample. Corresponding eigenvalues allow sorting of the vectors $\overrightarrow{v_i}$ in the order from the maximal to the minimal variance observed in the directions they point in. It is thus easy to select the vector $\overrightarrow{v_{max}}$ along which the variance is the largest and the vector $\overrightarrow{v_{min}}$ along which the variance is the smallest.

In the $\mathrm{PCA_{max}}$ approach the normal vector is set to $\overrightarrow{n_0} = \overrightarrow{v_{max}}$ and in the $\mathrm{PCA_{min}}$ the normal vector is set to $\overrightarrow{n_0} = \overrightarrow{v_{min}}$. Therefore, in the $\mathrm{PCA_{max}}$ method the variance in the direction transversal to Σ is maximized, but the variance within the section S is diminished. In the $\mathrm{PCA_{min}}$ method the variance in the direction transversal to Σ is minimized, but the variance within the section S is maximized. In both methods the point P_0 is set to the mean value $m \in \mathbb{R}^d$ of the observed points.

A basis obtained using PCA for a two-dimensional set of points is presented in Fig. 2. Obviously, in a two-dimensional case the $\overrightarrow{v_{max}}$ and $\overrightarrow{v_{min}}$ vectors are the only ones in the basis. In the figure they originate at the mean $m = [0, 0]$. Depending on which approach is selected ($\mathrm{PCA_{max}}$ or $\mathrm{PCA_{min}}$) the normal vector is set to either $\overrightarrow{n_0} = \overrightarrow{v_{max}}$ or $\overrightarrow{n_0} = \overrightarrow{v_{min}}$ respectively.

In all three methods: Perpendicular, $\mathrm{PCA_{max}}$ and $\mathrm{PCA_{min}}$ mentioned above the parameter μ for the "slice" technique was varied in the range from $\mu = 0.01$ to $\mu = 0.2$ with the step $\Delta\mu = 0.0001$. From all tested values of μ the one producing the highest value of the evaluation function was chosen.

Gradient

This method uses the gradient descent optimization method, which iteratively converges to an optimum using the function gradient to direct the search. Fig. 3 presents the idea of the gradient descent optimization. As shown in the figure, the optimization is performed iteratively starting from an initial point (in this paper selected randomly). In each step the algorithm calculates the gradient of the function (black arrow) at the currently known minimum point (black dot) and then searches for the next optimal point along the direction of the gradient.

The optimizer works on both P_0 and $\overrightarrow{n_0}$ treated jointly as a point in $\mathbb{R}^{2d}$. In the case of the "slice" method the search space is $\mathbb{R}^{2d+1}$ and the last coordinate is interpreted as the μ parameter. The optimization algorithm has to maximize the value of $\Delta(P_0, \overrightarrow{n_0})$. Of course, it is hard to guarantee the properties of the function $\Delta(P_0, \overrightarrow{n_0})$ typically required by gradient descent methods. Due to the discrete nature of the fractal dimension calculation procedure and a complex relationship between the values of P_0 and $\overrightarrow{n_0}$ and the obtained values of $\ln(L(k))$ the function $\Delta(P_0, \overrightarrow{n_0})$ cannot be assumed to have derivatives well defined at all points. Nevertheless, as seen in Sec. 4 the gradient descent method is able to produce quite good results. When optimizing complicated functions gradient descent methods can become stuck in local optima. Therefore in the Gradient approach used in this paper the optimization algorithm runs a certain number N_{runs} of times starting from randomly chosen points and the final result is the choice of P_0 and $\overrightarrow{n_0}$ which produces the maxi-

IN:
N_{gen} - the number of generations
N_{pop} - the size of the population
P_{cross} - crossover probability
P_{mut} - mutation probability

OUT:
$S^* = \emptyset$ - the best solution found by the algorithm

P = InitPopulation(N_{pop})
for $g = 1, \ldots, N_{gen}$ **do**
  Evaluate(P)
  S_0 = SelectBestSpecimen(P)
  S_1 = Mutate(S_0)
  P' = SelectMatingPool(P, $N_{pop} - 2$)
  P'' = Crossover(P', P_{cross})
  for $i = 1, \ldots, N_{pop} - 2$ **do**
    if Rand(U[0, 1]) < P_{mut} **then**
      Mutate($P''[i]$)
    end if
  end for
  $P = P'' \cup \{S_0\} \cup \{S_1\}$
  S^* = { SelectBestSpecimen($P \cup S^*$) }
end for

Figure 4: An overview of the evolutionary algorithm used in this paper.

mal observed value of $\Delta(P_0, \overrightarrow{n_0})$.

Evolutionary Algorithm (EA)

This method uses an evolutionary algorithm for optimizing the values of $\Delta(P_0, \overrightarrow{n_0})$. Population-based methods are able to effectively solve optimization problems involving multimodal functions and have the advantage that they do not assume specific properties of the goal function such as the existence of derivatives. These methods are also very effective for dealing with multiobjective optimization problems because they maintain an entire population of solutions that represent an approximation of a true Pareto front of the solved problem. A recent survey [26] states, that by January 2011 more than 5600 papers have been published on evolutionary multiobjective optimization alone, not including other applications of evolutionary methods and other population-based algorithms.

An evolutionary algorithm processes a population of solutions which undergo the processes of selection, crossover and mutation (cf. Fig. 4). The algorithm presented in this paper uses the mechanism of elitism, that is, it promotes the best specimen from the current generation and its mutated copy to the next generation without the need of winning in the selection process.

When the "intersect" technique is used, the genotype of the specimen used in this paper is simply a vector in $\mathbb{R}^{2d}$ which is split into P_0 and $\overrightarrow{n_0}$, which are used to determine the Poincaré section. In the case of the "slice" method one more real number that represents the μ parameter is added to the genotype (cf. Fig. 5).

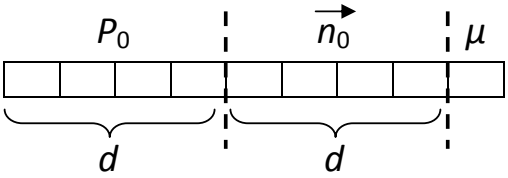


Figure 5: A genotype used in the evolutionary algorithm in the "slice" method.

Before the vector $\overrightarrow{n_0}$ is used in further calculations it is normalized by dividing all coordinates by $|\overrightarrow{n_0}|$. The evaluation of the specimen is the value of the function $\Delta(P_0, \overrightarrow{n_0})$ obtained using values stored in the genotype of the specimen.

The procedures used by the evolutionary algorithm are as follows.

InitPopulation - creates a given number of new solutions. The first d coordinates of a new solution are initialized to random values drawn from a normal distribution with the mean and variance equal to those estimated from the processed signal. The next d coordinates are initialized to random values drawn from a normal distribution $N(0, 1)$ which has the mean 0 and variance 1. In the case of the "slice" technique one more element in the genotype is required for the μ parameter. This element is initialized with a random variable drawn from the $N(0, 0.1)$ distribution.

Evaluate - for each specimen in the population this procedure calculates the value $\Delta(P_0, \overrightarrow{n_0})$ based on values stored in the genotype of the specimen.

SelectBestSpecimen - selects the specimen with the highest evaluation value from a given set of specimens.

SelectMatingPool - selects a given number of specimens into a mating pool with probability of selecting a specimen proportional to the evaluation of this specimen.

Crossover - produces a set of new specimens based on a given mating pool. For each consecutive pair of specimens from the mating pool a new pair of specimens is created with the probability P_{cross}, and with the probability $1 - P_{cross}$ both specimens are copied to the next generation without changes. In this paper the uniform crossover is used, which for every position in the genotype copies the value from one of the two parent specimens into one of the two offspring with $1/2$ probability. Thus, this operator copies approximately one half of the values from one parent and one half from the other parent into each new offspring.

Mutation - this operator selects one position in the genotype with the uniform probability and then adds a value drawn from a normal distribution with zero mean and the standard deviation equal 0.1 to the value of the selected coordinate in the genotype.

After N_{gen} generations are completed the genotype of the specimen with the highest evaluation found among all the specimens processed is returned as the best solution found by the algorithm.

The optimization step attempts to produce a Poincaré section that is as good as possible for obtaining distinctive patterns. The entire procedure for determining the type of behaviour for a given time series consists, therefore, of two steps.

1. A chosen algorithm optimizes the Poincaré section parameters (location and orientation of the subspace Σ) based on the optimization criterion (10).
2. A plot of $\ln(L(k))$ versus k is generated and evaluated.

While the evaluation in step 2. can be done visually, in this paper a numerical evaluation is performed by counting the number of increases (that is situations when $y_{k+1} > y_k$). As the results presented in Sec. 4 show, the number of such increases is, in most cases, much higher in the case of deterministic time series. Numerical approach has the advantage in that it is objective and allows easy incorporation of the proposed method as one of many data processing steps which may be useful for developing more sophisticated methods relying on the analysis of the type of behaviour of the time series.

4. Experiments and Results

The experiments were performed on twelve data sets: six generated by deterministic chaotic systems and six generated by stochastic processes. The six data sets used in the experiments generated by deterministic chaotic systems were: Bouali, Chen-Lee, Logistic, Lorenz, Rabinovich-Fabrikant and Rössler.

Bouali

A system introduced in 2012 by Bouali [27] as an example of an attractor topologically different from previously known ones. This model consists of three ordinary differential equations:

$$\begin{aligned} \frac{dx}{dt} &= x(a - y) + \alpha z \,, \\ \frac{dy}{dt} &= -y(b - x^2) \,, \\ \frac{dz}{dt} &= -x(c - sz) - \beta z \,. \end{aligned} \tag{11}$$

Parameters of this system were $a = 4$, $b = 1$, $c = 1.5$, $s = 1$, $\alpha = 0.3$ and $\beta = 0.05$.

Chen-Lee

Proposed in 2004 by Chen and Lee [28] in a paper concerning the behaviour of a rigid body influenced by certain feedback gains. This model consists of the equations:

$$\begin{aligned} \frac{dx}{dt} &= -yz + ax \,, \\ \frac{dy}{dt} &= xz + by \,, \\ \frac{dz}{dt} &= (1/3)xy + cz \,. \end{aligned} \tag{12}$$

Parameters of this system were $a = 5$, $b = -10$ and $c = -3.8$.

Logistic

This data set is a 1-dimensional time series generated by a discrete-time system introduced in 1962 by Pekka Juhana Myrberg [29]. It is given by an equation:

$$x_{n+1} = rx_n(1 - x_n) \,. \tag{13}$$

In the experiments the parameter value of $r = 3.8284$ was used. As with all 1D time series used in this paper this time series was embedded into $\mathbb{R}^3$ before applying the P&H method.

Lorenz

A system developed in 1963 by Edward Lorenz [30] as a simplified mathematical model for atmospheric convection. This model consists of three ordinary differential equations:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \,, \\ \frac{dy}{dt} &= x(\rho - z) - y \,, \\ \frac{dz}{dt} &= xy - \beta z \,. \end{aligned} \tag{14}$$

Parameters of this system were $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$.

Rabinovich-Fabrikant

A system introduced in 1979 by Rabinovich and Fabrikant [31] for modeling the dynamical behaviour caused by the modulation instability in a non-equilibrium dissipative medium. The R-F system is an example of a dynamical system with multiple attractors [32]. This model consists of three ordinary differential equations:

$$\begin{aligned} \frac{dx}{dt} &= y(z - 1 + x^2) + \gamma x \,, \\ \frac{dy}{dt} &= x(3z + 1 - x^2) + \gamma y \,, \\ \frac{dz}{dt} &= -2z(\alpha + xy) \,. \end{aligned} \tag{15}$$

Parameters of this system were $\alpha = 1.1$ and $\gamma = 0.87$.

Rössler

Probably the most popular example of the a simple folding proposed in 1976 by Rössler [33]. This model consists of three ordinary differential equations:

$$\begin{aligned} \frac{dx}{dt} &= -y - z \,, \\ \frac{dy}{dt} &= x + ay \,, \\ \frac{dz}{dt} &= b + z(x - c) \,. \end{aligned} \tag{16}$$

Parameters of this system were $a = 0.2$, $b = 0.2$ and $c = 5.7$.

In order to study the influence of noise on the possibility of detecting deterministic behaviour, random perturbations were added to the sets of points in R^3 produced by the above-mentioned systems. The parameter controlling the level of noise was set to $\eta = 0.00$, 0.02, 0.04, 0.06, 0.08 and 0.10. The procedure for adding noise to the original data sets was as follows. For each coordinate separately the standard deviation σ of points in the data set was calculated. Then, to each data point a random displacement was added with each coordinate drawn from a normal distribution $N(0, \eta \cdot \sigma)$ with the respective σ for each coordinate. Obviously, for $\eta = 0.00$ no noise was added, and for the remaining values an increasing level of noise was introduced to the data set.

The six data sets used in the experiments generated by stochastic processes were: Gaussian 1D, Gaussian 3D, Random Walk 1D, Random Walk 3D, Uniform 1D and Uniform 3D.

Gaussian 1D

This data set is a 1-dimensional time series with all the elements independently drawn from a 1-dimensional normal distribution with a mean 0 and a standard deviation 1. As with all 1D time series used in this paper this time series was embedded into $\mathbb{R}^3$ before applying the P&H method.

Gaussian 3D

This data set is a 3-dimensional time series with all the elements independently drawn from a 3-dimensional normal distribution with a mean $[0, 0, 0]$ and an identity covariance matrix $\mathbb{I}_3$.

Random Walk 1D

This data set contains a 1-dimensional time series with the first value equal to 0 and each consecutive element obtained by adding a value drawn from a 1-dimensional normal distribution with a mean 0 and a standard deviation 1 to the previous one. As with all 1D time series used in this paper this time series was embedded into $\mathbb{R}^3$ before applying the P&H method.

Random Walk 3D

This data set contains a 3-dimensional time series with the first value $[0, 0, 0]$ and each consecutive element obtained by adding a 3-dimensional vector drawn from a 3-dimensional normal distribution with a mean $[0, 0, 0]$ and an identity covariance matrix $\mathbb{I}_3$.

Uniform 1D

This data set is a 1-dimensional time series with all the elements independently drawn from a 1-dimensional uniform distribution on the interval $[0, 1]$. As with all 1D time series used in this paper this time series was embedded into $\mathbb{R}^3$ before applying the P&H method.

Uniform 3D

This data set is a 3-dimensional time series with all the elements independently drawn from a 3-dimensional uniform distribution on the cube $[0, 1]^3$.

Because the six data sets mentioned above are generated by random processes no additional noise was added to these data sets.

In the experiments the five methods of finding good Poincaré sections described earlier in this paper were tested on all data sets. For each method and data set 10 repetitions of the test were performed. The noise added to the data set was generated separately for each repetition. In the case of Gradient method the search is restarted from random points and the EA method involves random initialization of the population and probabilistic mechanisms for mating pool selection and mutation. Therefore, the effects of running these methods in the 10 repetitions were different.

Both the Gradient and the EA methods require certain parameters to be set. In the case of the Gradient method the number of iterations was $N_{iter} = 100$ and the number of restarts $N_{runs} = 10$. In the case of the EA method the population size was $N_{pop} = 1000$ and the number of generations $N_{gen} = 50$. The crossover probability was $P_{cross} = 0.9$ and the mutation probability $P_{mut} = 0.05$. It is worth noting that these probabilities are set to values typically used in the literature and they were not specifically tuned for the given problem instances. Similarly, other parameters of both optimization algorithms were set once for all tested data sets.

In the P&H method the fact that a given data set is an output of a deterministic process is indicated by the presence of sharp increases in the graph of $\ln(L(k))$. In the case of a stochastic process a smoothly decreasing curve is expected. To evaluate the solutions found by the methods studied in this paper the number of increases in the graph of $\ln(L(k))$ was counted. The count was increased by one for each k such that $y_{k+1} > y_k$ where $\{y_k = \ln(L(k))\}_{k=1,2,3,\ldots,k_{max}}$ were ordinates of points produced by the P&H method. Table 2 (in the Appendix) presents average values obtained in 10 runs for data sets generated by deterministic chaotic systems. Table 3 presents the averages for data sets generated by stochastic processes.

Figures 6 and 7 (in the Appendix) present, for the "intersect" and "slice" method respectively, how many times a given number of increases in the graphs of $\ln(L(k))$ was obtained by each optimization method in the tests for all data sets with a given noise level and for the stochastic data sets. The deterministic and stochastic behaviour is well distinguished if the number of increases obtained for stochastic time series (white bars) is predominantly low and for deterministic time series (gray bars) there are numerous instances in which large numbers of increases were obtained. Note, that these graphs treat all the data sets with a given noise level jointly, and similarly all the stochastic data sets. Therefore, while the graphs can be used to evaluate the general behaviour of the presented method the evaluation of the performance for individual data sets is only possible using Tables 2 and 3. The two best performing methods seem to be the Gradient and EA, both using the "slice" technique. In the case of the EA method no stochastic time series has produced the number of increases larger than 6 and no deterministic time series without noise has produced the number of increases smaller than 5. In a vast majority of cases the deterministic time series have produced 9 increases regardless of the noise level. There are a few cases in which deterministic time series without noise have produced as few as 5 increases. Based on the results presented in Table 2 it is clear that they all were obtained for the Lorenz data set - the only one for which the last column of Table 2 contains a value less than 9.0 for $\eta = 0.00$. In the case of the Gradient method the discrimination between the deterministic and the stochastic cases is good also, because no stochastic time series has produced the number of increases larger than 2. There are, however, quite a few cases in which deterministic time series without noise have produced 0 increases.

Figures 8 and 9 present the graphs generated by the P&H method. The figures were made for both stochastic data sets and the deterministic ones and they represent the best solution found in 10 runs in terms of the Δ function (cf. equation (10)). Because of the large number of figures in the case of deterministic data sets only the graphs for noise level $\eta = 0.10$ are presented. This is the highest level of noise applied in the tests,

so it makes it the hardest to properly determine the type of the time series behaviour. As can be seen in the graphs only for the Lorenz data set the zig-zag pattern is hard to detect.

Overall, the EA optimization method was able to find the best Poincaré sections among the tested optimization methods. This is not surprising, because population-based optimization methods are known to produce good results in the case of complicated functions. The Gradient method produced in some cases relatively good results considering the fact that in the case of fractal dimension calculation and subsequent application of the P&H method the resulting goal function cannot be expected to have desirable properties such as continuous derivatives. Of the two approaches based on the Principal Component Analysis only the PCA_{min} method produced in some cases a visible pattern, but overall seems not to be effective enough to be considered reliable. The simplest approach of placing the Σ subspace perpendicularly to one of the axes of the coordinate system was unable to produce meaningful patterns.

In the case of three data sets (Bouali, Logistic and Rössler) the average number of increases observed in 10 repetitions of tests in the plot of $\ln(L(k))$ obtained using the EA method was 9.0 (cf. Table 2). This means that 9 increases have been produced in every repetition of the test, because it is the maximum possible with $k_{max} = 20$. This result was achieved with all tested levels of noise. In the case of two further data sets (Chen-Lee and Rabinovich-Fabrikant) the same result was obtained when the noise was not present ($\eta = 0.00$). If the noise was added to the Chen-Lee and Rabinovich-Fabrikant data sets the number of detected increases was in some cases smaller. In the case of the Lorenz data set even without noise only the average value of 7.3 increases was obtained. For the noisy versions of the Chen-Lee, Lorenz and Rabinovich-Fabrikant data sets even lower averages were observed, reaching as low as 2.9. Because the highest observed average for a stochastic data set was 4.2 this means that on some occasions a noisy deterministic signal may be mistaken for a stochastic one. On the other hand the noise-free case should be easily distinguished from the stochastic one because the lowest average number of increases obtained for a noise-free deterministic data set was 7.3.

Apart from the quality of the results the running time of the algorithms was also measured. The average running times of all the algorithms obtained when running on a single core of an Intel Q6600 2.4 GHz CPU are presented in Table 1. In general, the better performing methods took longer to compute. The fastest were the methods based on the PCA and the method which tries to find a subspace Σ perpendicular to the axes of the coordinate system. As shown in the Tables 2 and 3 they produced, however, very poor results. The two best-performing methods, which are Gradient and EA, took the longest to compute, but also provided results much better than the first three ones.

Table 1: The average running times of all the algorithms

Algorithm	Average running time [s]	
	Intersect	Slice
Perpendicular	0.58	249.02
PCA_{max}	0.30	86.22
PCA_{min}	0.29	278.95
Gradient	537.58	880.77
EA	1830.32	3174.39

5. Conclusion

This paper studies the problem of the optimization of Poincaré sections for discriminating between stochastic and deterministic dynamical systems. The P&H method proposed by Golestani et al. [16] produces distinctive patterns, which can be used to distinguish deterministic chaotic systems and stochastic ones. However, this method is sensitive to the placement of the Poincaré section as depicted in Figure 1.

Results presented in this paper show, that optimization methods can be used for finding such Poincaré sections that produce the most prominent patterns in the plots generated by the P&H method. In particular, the evolutionary optimization approach is able to find good Poincaré sections. Using the optimized sections it is possible to distinguish between deterministic chaotic systems and stochastic ones. When noise is not present the distinction between chaotic and stochastic behaviour is very clear. The addition of noise makes the task harder, but nevertheless by optimizing the placement of the Poincaré section it should be possible to correctly determine the type of behaviour.

An interesting observation is that the P&H method can work more effectively if the section of the flow is obtained not by intersecting the orbits with the subspace Σ (the "intersect" method), but rather by projecting nearby data points onto it (the "slice" method).

Further work on this topic may include applying other optimization methods and choosing different goal functions for optimization. In the case of very noisy signals it might be necessary to establish a method, other than just counting the increases, for determining if a given $\ln(L(k))$ plot represents a deterministic or a stochastic system, For example, small increases in the $\ln(L(k))$ plot might be caused by random fluctuations in the noisy signal, so it could be worthwhile to develop methods for filtering out such fluctuations without removing meaningful variations. In the presence of noise the approach described in this paper could also be hybridized with preprocessing methods such as noise removal techniques.

Another interesting question is if the Poincaré sections optimized with respect to the performance of the P&H method could be used effectively for other methods of analyzing the behaviour of dynamical systems. To allow using the presented optimization methods in other studies, the program generating optimized Poincaré sections was made available at `http://krzysztof-michalak.pl/ps%5Fopt/index.html`.

References

[1] P. Grassberger, I. Procaccia, Characterization of strange attractors, Phys. Rev. Lett. 50 (1983) 346–349.

[2] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, 2nd Edition, Cambridge University Press, 2003.

[3] M. Sano, Y. Sawada, Measurement of the Lyapunov spectrum from a chaotic time series, Phys. Rev. Lett. 55 (1985) 1082–1085.
[4] A. Wolf, J. B. Swift, H. L. Swinney, J. A. Vastano, Determining Lyapunov exponents from a time series, Physica D 16 (1985) 285–317.
[5] M. T. Rosenstein, J. J. Collins, C. J. D. Luca, A practical method for calculating largest Lyapunov exponents from small data sets, Physica D 65 (1993) 117–134.
[6] J. B. Gao, J. Hu, W. W. Tung, Y. H. Cao, Distinguishing chaos from noise by scale-dependent Lyapunov exponent, Phys. Rev. E 74 (2006) 066204.
[7] J.-P. Eckmann, D. Ruelle, Ergodic theory of chaos and strange attractors, Reviews of Modern Physics 57 (1985) 617–656.
[8] S. M. Pincus, Approximate entropy as a measure of system complexity, Proceedings of the National Academy of Sciences 88 (6) (1991) 2297–2301.
[9] M. Q.-L. Wang Jun, Multiscale entropy based study of the pathological time series, Chinese Physics B 17 (12) (2008) 4424–4427.
[10] C. M., G. A. L., P. C.-K., Multiscale entropy analysis of physiologic time series, Phys. Rev. Lett. 89 (2002) 062102.
[11] C. M., G. A.L., P. C.-K., Multiscale entropy analysis of biological signals, Phys. Rev. E. 71 (2005) 021906.
[12] L. Jin, N. Xin-Bao, W. Wei, M. Xiao-Fei, Detecting dynamical complexity changes in time series using the base-scale entropy, Chinese Physics 14 (12) (2005) 2428–2432.
[13] M. Cencini, M. Falcioni, E. Olbrich, H. Kantz, A. Vulpiani, Chaos or noise: Difficulties of a distinction, Phys. Rev. E 62 (2000) 427–437.
[14] G. Boffetta, M. Cencini, M. Falcioni, A. Vulpiani, Predictability: a way to characterize complexity, Physics Reports 356 (2002) 367–474.
[15] S. Jian-Cheng, Z. Tai-Yi, L. Feng, Modelling of chaotic systems based on modified weighted recurrent least squares support vector machines, Chinese Physics 13 (12) (2004) 2045–2052.
[16] A. Golestani, M. R. Jahed Motlagh, K. Ahmadian, A. H. Omidvarnia, N. Mozayani, A new criterion to distinguish stochastic and deterministic time series with the poincare section and fractal dimension, Chaos: An Interdisciplinary Journal of Nonlinear Science 19 (1) (2009) 013137.
[17] T. Higuchi, Approach to an irregular time series on the basis of the fractal theory, Physica D: Nonlinear Phenomena 31 (2) (1988) 277–283.
[18] T. Higuchi, Relationship between the fractal dimension and the power law index for a time series: A numerical investigation, Physica D: Nonlinear Phenomena 46 (2) (1990) 254–264.
[19] L. Telesca, G. Colangelo, V. Lapenna, M. Macchiato, Monofractal and multifractal characterization of geoelectrical signals measured in southern italy, Chaos, Solitons & Fractals 18 (2) (2003) 385–399.
[20] W. Klonowski, E. Olejarczyk, R. Stepien, Sleep-EEG analysis using higuchi's fractal dimension, in: Proceedings of the International Symposium on Nonlinear Theory and Its Applications (NOLTA '05), 2005, pp. 222–225.
[21] A. Golestani, A. Ashouri, K. Ahmadian, M.-R. J. Motlagh, M.-A. Doostari, Irregularity analysis of iris patterns, in: IPCV'08, 2008, pp. 691–695.
[22] A. Golestani, R. Gras, Regularity analysis of an individual-based ecosystem simulation, Chaos: An Interdisciplinary Journal of Nonlinear Science 20 (4) (2010) 043120.
[23] F. Takens, Detecting strange attractors in turbulence, Lecture Notes in Mathematics 898 (1981) 366–381.
[24] R. Hegger, H. Kantz, T. Schreiber, E. Olbrich, Tisean 3.0.1 - nonlinear time series analysis, `http://www.mpipks-dresden.mpg.de/~tisean/Tisean_3.0.1/index.html`, online: accessed 2015.07.22 (2014).
[25] I. T. Jolliffe, Principal Component Analysis, 2nd Edition, Springer, 2002.
[26] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm and Evolutionary Computation 1 (1) (2011) 32–49.
[27] S. Bouali, A novel strange attractor with a stretched loop, Nonlinear Dynamics 70 (4) (2012) 2375–2381.
[28] H.-K. Chen, C.-I. Lee, Anti-control of chaos in rigid body motion, Chaos, Solitons & Fractals 21 (4) (2004) 957 – 965.
[29] P. J. Myrberg, Sur literation des polynomes reels quadratiques, Journal de Mathematiques pures et appliquees 9 (41) (1962) 339–351.
[30] E. N. Lorenz, Deterministic nonperiodic flow, Journal of Atmospheric Sciences 20 (1963) 130–148.
[31] M. I. Rabinovich, A. L. Fabrikant, Stochastic self-modulation of waves in nonequilibrium media, Soviet Journal of Experimental and Theoretical Physics 50 (1979) 311.
[32] X. Luo, M. Small, M.-F. Danca, G. Chen, On a dynamical system with multiple chaotic attractors.
[33] O. E. Rossler, An equation for continuous chaos, Physics Letters A 57 (5) (1976) 397–398.

Appendix: Tables and figures presenting the results

Table 2: The average number of increases observed in 10 repetitions of the tests in the plot of $\ln(L(k))$ obtained using the methods of optimizing the Poincaré section for deterministic time series

Data set	Intersect					Slice				
	Perp.	**PCA_{max}**	**PCA_{min}**	**Gradient**	**EA**	**Perp.**	**PCA_{max}**	**PCA_{min}**	**Gradient**	**EA**
Bouali										
$\eta = 0.00$	7.0	0.0	0.0	0.1	9.0	0.0	0.0	9.0	6.6	9.0
$\eta = 0.02$	2.5	0.9	0.2	1.5	8.9	0.8	0.4	6.0	7.8	9.0
$\eta = 0.04$	0.0	0.0	0.0	0.1	8.5	0.5	0.0	6.0	7.5	9.0
$\eta = 0.06$	0.2	2.5	0.0	0.1	5.9	0.3	0.0	6.0	6.9	9.0
$\eta = 0.08$	0.0	0.0	0.0	0.0	2.6	0.0	0.0	6.0	7.2	9.0
$\eta = 0.10$	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.0	7.2	9.0
Chen-Lee										
$\eta = 0.00$	6.0	0.0	0.0	8.4	9.0	0.0	0.0	9.0	0.0	9.0
$\eta = 0.02$	5.9	0.0	0.0	4.7	8.8	1.1	0.3	6.0	7.2	9.0
$\eta = 0.04$	5.6	0.0	0.0	0.7	7.9	0.7	0.0	0.0	7.5	9.0
$\eta = 0.06$	2.7	0.0	0.0	1.9	5.0	0.2	0.0	0.0	6.9	3.0
$\eta = 0.08$	0.3	0.1	0.0	0.5	1.9	0.0	0.0	6.0	6.9	9.0
$\eta = 0.10$	0.2	0.0	0.0	1.2	0.9	0.0	0.0	6.0	6.3	9.0
Logistic										
$\eta = 0.00$	0.0	0.0	0.0	2.2	9.0	0.0	0.0	9.0	8.4	9.0
$\eta = 0.02$	0.0	0.0	0.0	0.3	8.9	0.8	0.2	6.0	6.6	9.0
$\eta = 0.04$	0.0	0.0	0.0	0.8	8.5	0.3	0.0	6.0	7.5	9.0
$\eta = 0.06$	0.0	0.0	0.0	0.8	4.2	0.1	0.0	6.0	7.5	9.0
$\eta = 0.08$	0.0	0.0	0.0	0.5	2.4	0.0	0.0	6.0	7.2	9.0
$\eta = 0.10$	0.0	0.0	0.0	0.2	0.5	0.0	0.0	6.0	6.9	9.0
Lorenz										
$\eta = 0.00$	1.0	1.0	0.0	0.3	8.8	3.0	2.0	1.0	2.6	7.3
$\eta = 0.02$	0.0	0.4	0.0	0.3	8.9	2.3	1.7	0.9	3.1	6.1
$\eta = 0.04$	0.0	0.0	0.0	0.0	8.1	2.8	2.1	1.2	2.4	6.0
$\eta = 0.06$	0.0	0.0	0.0	0.0	5.5	2.8	1.7	0.9	1.2	6.1
$\eta = 0.08$	0.0	0.0	0.0	0.2	4.8	1.9	2.1	0.6	1.7	5.8
$\eta = 0.10$	0.1	0.0	0.0	0.1	1.1	2.9	1.6	0.9	1.7	5.4
Rab.-Fab.										
$\eta = 0.00$	1.0	0.0	0.0	9.0	9.0	0.0	0.0	9.0	6.6	9.0
$\eta = 0.02$	0.0	0.1	0.1	1.4	7.2	0.6	0.1	6.0	7.8	4.8
$\eta = 0.04$	0.0	0.2	0.0	0.3	4.7	0.0	0.0	6.0	7.5	2.9
$\eta = 0.06$	0.0	0.0	0.0	0.0	2.3	0.1	0.0	6.0	6.6	9.0
$\eta = 0.08$	0.0	0.0	0.0	0.3	2.4	0.0	0.0	6.0	6.6	9.0
$\eta = 0.10$	0.0	0.0	0.0	0.1	2.6	0.0	0.0	6.0	6.9	9.0
Rössler										
$\eta = 0.00$	0.0	0.0	0.0	9.0	9.0	0.0	0.0	9.0	7.5	9.0
$\eta = 0.02$	0.0	0.0	0.0	0.9	7.7	0.5	0.1	6.0	7.5	9.0
$\eta = 0.04$	0.0	0.0	0.0	1.4	6.8	0.9	0.0	6.0	8.1	9.0
$\eta = 0.06$	0.0	0.0	0.0	0.9	7.4	0.6	0.0	6.0	6.3	9.0
$\eta = 0.08$	0.0	0.0	0.0	0.2	2.6	0.1	0.0	6.0	6.9	9.0
$\eta = 0.10$	0.0	0.0	0.0	0.0	0.6	0.0	0.0	6.0	0.0	9.0

Table 3: The average number of increases observed in 10 repetitions of the tests in the plot of $\ln(L(k))$ obtained using the methods of optimizing the Poincaré section for stochastic time series

Data set	Intersect					Slice				
	Perp.	**PCA_{max}**	**PCA_{min}**	**Gradient**	**EA**	**Perp.**	**PCA_{max}**	**PCA_{min}**	**Gradient**	**EA**
Gaussian 1D	0.0	0.0	0.0	0.4	2.6	0.0	0.0	0.0	0.2	3.7
Random Walk 1D	0.7	0.4	0.0	1.5	3.6	1.1	0.7	0.0	1.1	4.1
Uniform 1D	0.0	0.0	0.0	0.1	2.8	0.0	0.0	0.0	0.2	2.7
Gaussian 3D	2.0	0.0	0.0	2.2	6.3	0.0	0.0	0.0	0.0	4.2
Random Walk 3D	0.0	0.0	0.0	0.0	5.1	0.0	0.0	0.0	0.0	0.0
Uniform 3D	0.0	0.0	0.0	0.1	3.9	0.0	0.0	0.0	0.1	3.7

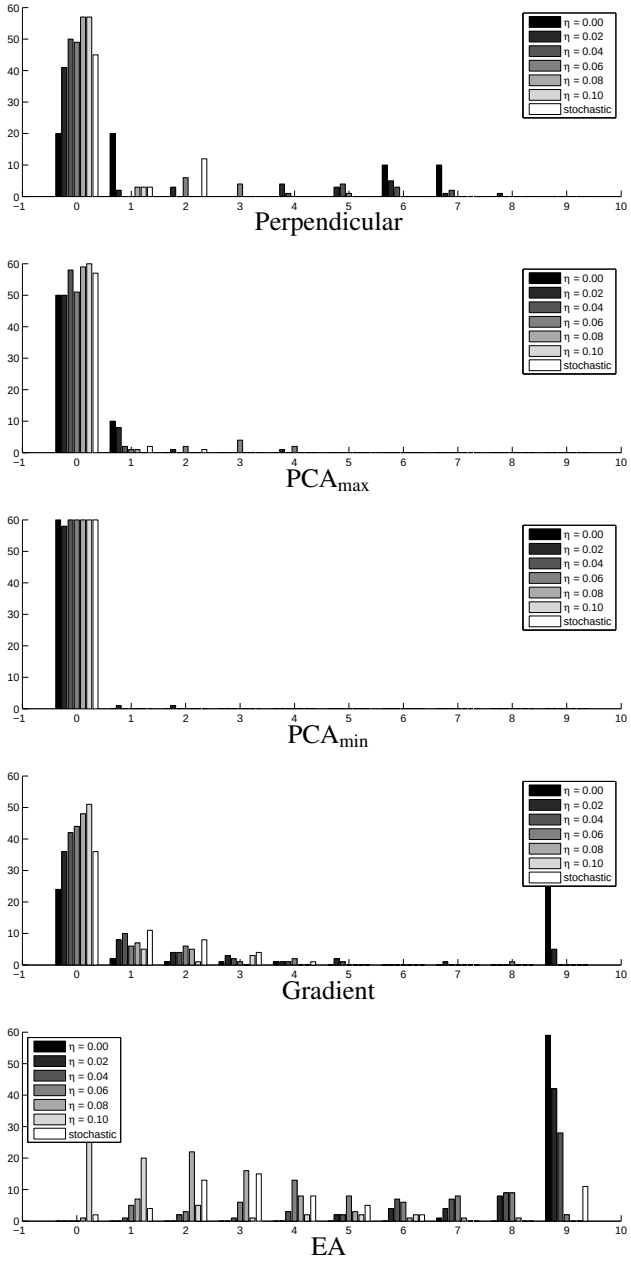


Figure 6: The number of increases obtained using the ”intersect” method for all the tested time series

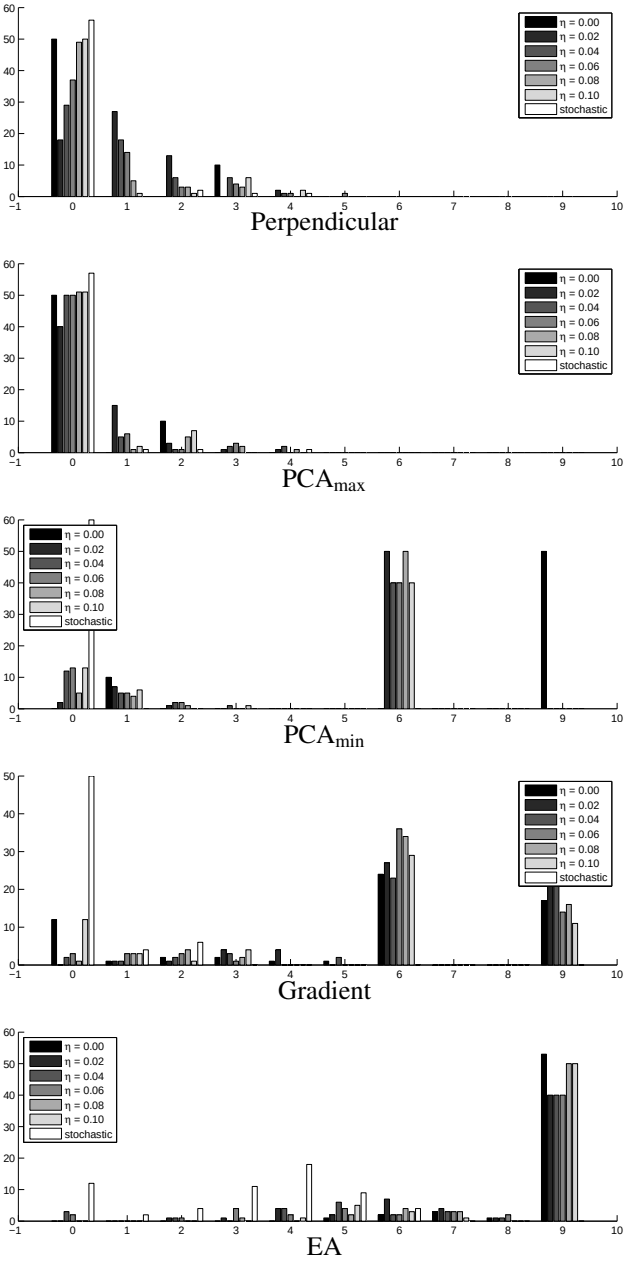


Figure 7: The number of increases obtained using the ”slice” method for all the tested time series

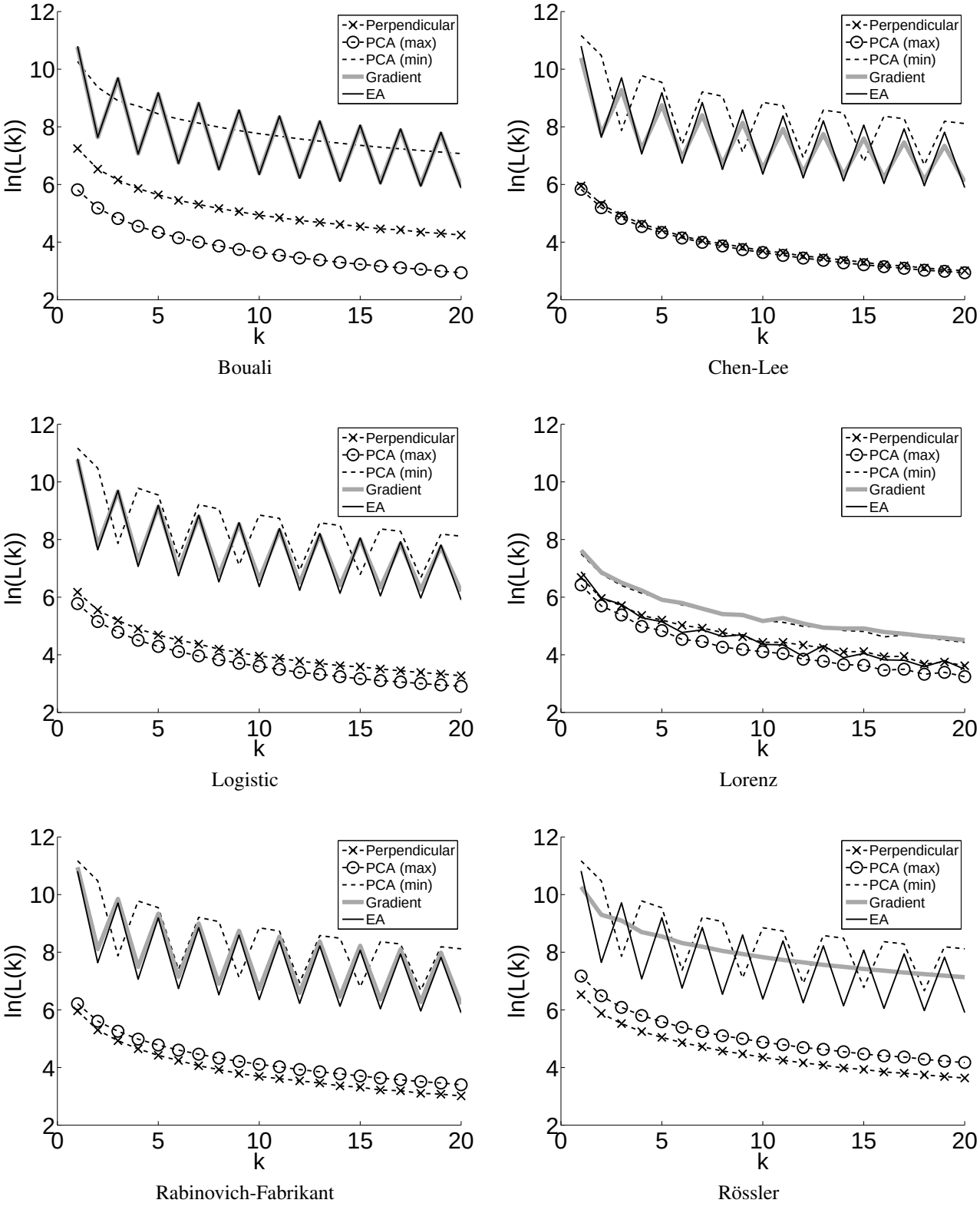
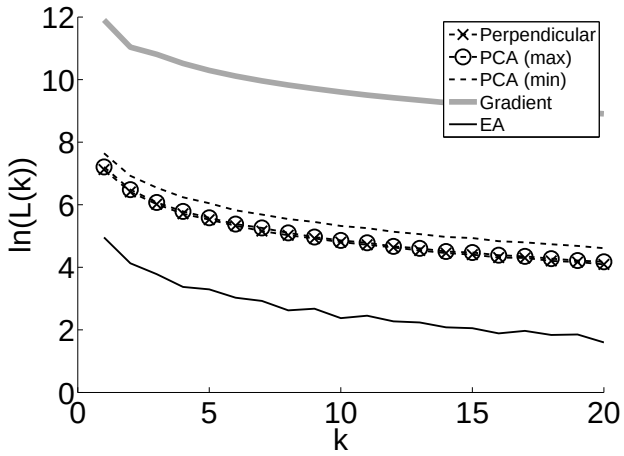
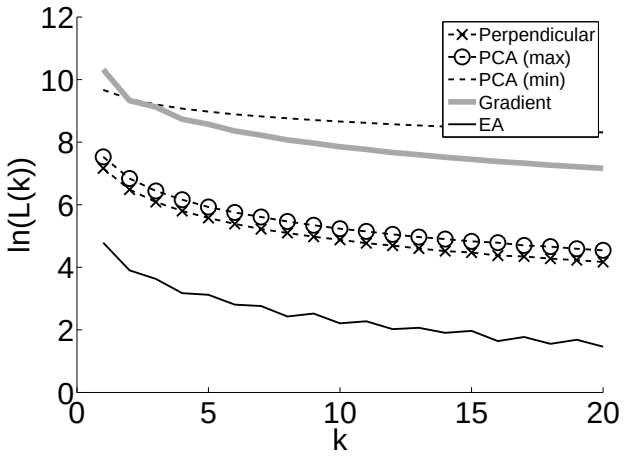


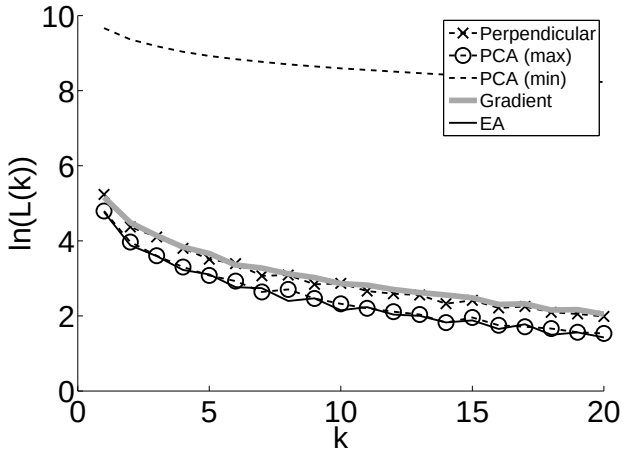
Figure 8: Values of $\ln(L(k))$ obtained using the "slice" method for deterministic time series with noise level $\eta = 0.10$



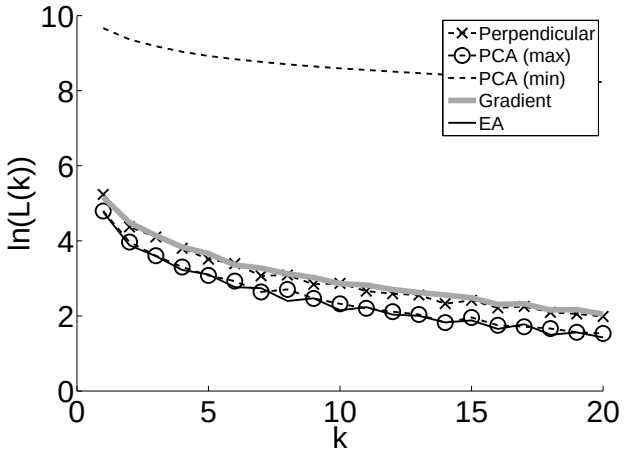
Gaussian 1D



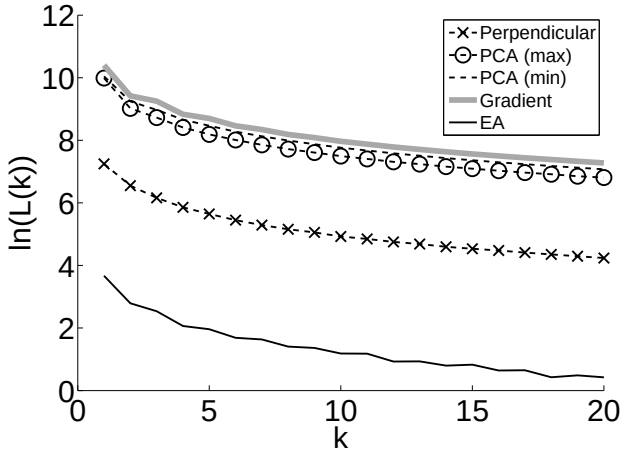
Gaussian 3D



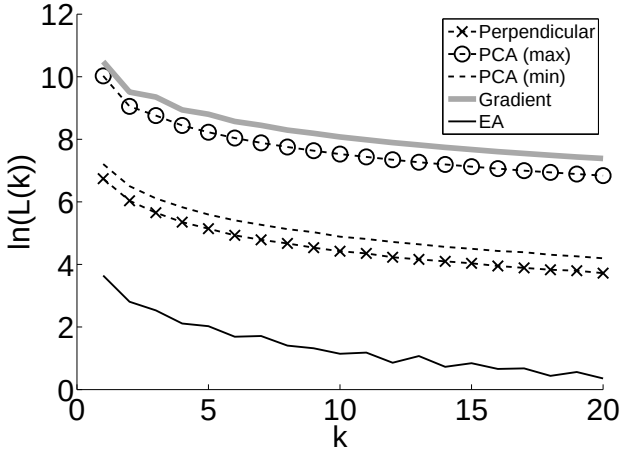
Random Walk 1D



Random Walk 3D



Uniform 1D



Uniform 3D

Figure 9: Values of $\ln(L(k))$ obtained using the "slice" method for stochastic time series