Revisiting the box counting algorithm for the correlation dimension analysis of hyperchaotic time series

K.P. Harikrishnan a,⇑, R. Misra b, G. Ambika c

a Department of Physics, The Cochin College, Cochin 682002, India
b Inter-University Center for Astronomy and Astrophysics, Pune 411007, India
c Indian Institute of Science Education and Research, Pune 411021, India

A R T I C L E  I N F O

Article history:
Received 26 December 2010
Received in revised form 30 April 2011
Accepted 4 May 2011
Available online 11 May 2011

Keywords:
Hyperchaos
Correlation dimension
Box counting algorithm

A B S T R A C T

We undertake the correlation dimension analysis of hyperchaotic time series using the box counting algorithm. We show that the conventional box counting scheme is inadequate for the accurate computation of correlation dimension ($D_2$) of a hyperchaotic attractor and propose a modified scheme which is automated and gives better convergence of $D_2$ with respect to the number of data points. The scheme is first tested using the time series from standard chaotic systems, pure noise and data added with noise. It is then applied on the time series from three standard hyperchaotic systems for computing $D_2$. Our analysis clearly reveals that a second scaling region appears at lower values of box size as the system makes a transition into the hyperchaotic phase. This, in turn, suggests that correlation dimension analysis can also give information regarding chaos-hyperchaos transition.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Since the introduction of the Lorenz system in 1963 [1], the developments in the field of deterministic chaos have been very rapid. The Lorenz attractor revealed all the complex and fundamental features of a typical low dimensional chaotic attractor and was instrumental in the development of many new mathematical and numerical techniques for the analysis of dynamical systems in general. Along with the theoretical advances, a large number of practical applications have also been developed for chaotic systems over the past four decades, on account of their complex dynamics and the self similar fractal structure of the underlying attractors.

But recently, with regard to practical applications, much of the attention has been shifted to systems producing hyperchaotic attractors. They are characterised by more than one positive Lyapunov exponents (LE) and are much more complex in terms of topological structure as well as dynamics compared to low dimensional chaotic attractors. Hence hyperchaotic attractors are preferred for many of the applications that require the complexity of the dynamics, as in network security [2] and data encryption [3,4] and also in many synchronisation studies using electro-optic devices [5,6]. In fact, hyperchaos was first reported by Rossler in 1979 [7], but became popular only in the last decade or so when its potential in secure communication was realised by various scientific and engineering communities.

Broadly, there are two types of nonlinear dynamical systems that can generate hyperchaotic attractors. One is a system of coupled autonomous differential equations (flow) or a system of coupled oscillators and the second is a nonlinear time delayed differential equation. In the former case, the dimensionality of the system should be at least four to display hyperchaotic behavior since there are at least two directions of stretching. A large number of such systems have been...
developed and analysed in the last few years [8–10]. The hyperchaotic flows are widely used in data encryption and secure messaging.

The most important examples of the second kind are the Mackey–Glass (M-G) system [11] and the Ikeda system [12]. Here the behavior of the system crucially depends on the time delay parameter, usually represented by \( \tau \). Depending on the value of \( \tau \), the system displays a range of periodic, chaotic as well as hyperchaotic dynamics. These systems are mainly developed from a biological perspective and more details regarding these systems are discussed in §4.

Hyperchaotic systems are mainly analysed by computing the LEs as a function of the control parameters. An important aspect in their quantitative analysis is the transition from chaos to hyperchaotic phase. Generally, this occurs when one of the control parameters passes through a critical value, when the second largest LE turns positive. When the equations governing the time evolution of the dynamical system are known, the transition from chaos to hyperchaos can be readily obtained through LE analysis. But, if the only information available on the system is a time series, such a method is more difficult to apply. In such cases, one is more inclined to use other measures, such as, \( D_2 \) or recurrence plots [13].

As the attractor becomes more complex with two directions of stretching in the hyperchaotic phase, the fractal dimension can also vary in a nontrivial manner. But the dimensional analysis of hyperchaotic systems have been very few in the literature. The main reason for this could be the practical difficulty for computing the dimension from a hyperchaotic time series, as discussed in detail in the next section. But in one of the attempts, [14], Kapitaniak et al. [15] have shown that \( D_2 \) displays more than one scaling region in the hyperchaotic phase for a system of unidirectionally coupled oscillators. The authors further show that the transition to hyperchaos is mediated by changes in the stability of an infinite number of unstable periodic orbits embedded in the chaotic attractor whose basin becomes riddled [16]. Thus, computing \( D_2 \) can give useful information regarding chaos-hyperchaos transition. Moreover, \( D_2 \) is an important quantifier that can represent the geometric complexity of a hyperchaotic attractor, which is vital for many practical applications of hyperchaos.

The computation of \( D_2 \) is usually done using a time series employing the delay embedding technique. Several methods have been proposed in the literature over the past two decades for computing \( D_2 \) from a time series, such as, the Grassberger–Procaccia (GP) algorithm [17], the box counting algorithm [18], the box assisted correlation algorithm by Theiler [19], the false nearest neighbour method [20], the gaussian kernal method by Diks [21] and the algorithm by Judd [22], to name a few. But none of the above methods have specifically addressed the issue of computing \( D_2 \) from a hyperchaotic time series, which is our main concern in this paper. Among the methods mentioned above, the GP algorithm and the box counting algorithm have gained more popularity due to their wide applicability. Hence we consider these two methods, to see which one is more suitable for the analysis of hyperchaotic time series.

We have recently proposed and implemented a non subjective approach to the GP algorithm [23] to compute \( D_2 \), that can be applied to synthetic as well as real world data involving noise. But the GP algorithm involves computation of distances from a reference point for calculating the correlation sum. As a consequence, it is much more expensive in terms of computer time and memory compared to the box counting algorithm, where the computer mainly does comparison and sorting. This is especially true if the number of data points in the time series is required to be large, as in the case of hyperchaotic time series. Moreover, since \( D_2 \) is typically large for these systems, the computation has to proceed to a much higher embedding dimension to ensure a reasonable saturation of \( D_2 \). The above factors give a definite advantage for the box counting algorithm over the GP algorithm in the analysis of hyperchaotic time series, which motivate us to undertake a systematic study of the box counting scheme.

We then find that the conventional box counting scheme is inadequate for the \( D_2 \) analysis of hyperchaotic time series, as shown in detail in the next section. Here we propose a modified scheme by redefining the expression for probability for computing \( D_2 \). By doing this, we are able to push the scaling region to much smaller values of box size compared to the conventional scheme. We show that this gives excellent results for hyperchaotic systems and we also bring out some interesting features of the hyperchaotic attractor using the modified scheme.

Our paper is organised as follows: In the next section, we point out the main drawbacks of the conventional box counting scheme and propose a modified scheme to compute \( D_2 \). In §3, the scheme is tested using several standard chaotic time series, random data and data added with noise. The scheme is then applied to the \( D_2 \) analysis of hyperchaotic systems in §4. Finally, the conclusions are drawn in §5.

2. Revisiting the box counting algorithm

2.1. The conventional scheme

The box counting algorithm for computing \( D_2 \) from a time series has been discussed by many authors [18,24–26] in the past. To apply the box counting scheme, one first constructs a delay embedding attractor from the time series in an embedding space of dimension \( M \) with a suitably chosen time delay \( \tau \). One then tries to cover the attractor using \( M \) dimensional cubes of side length \( r \). The correlation dimension \( D_2 \) is then defined by the expression

\[
D_2 \equiv \lim_{r \to 0} \frac{\log C(r)}{\log(r)},
\]

with \( C(r) \) given by

\[
C(r) = \sum_{i=1}^{N(r)} p_i^2(r).
\]
Here $N(r)$ is the number of non empty boxes and $p_i(r)$ is the probability that the trajectory passes through the $i$th box. But for a finite data stream, the limiting value is not available and hence a proper linear part in the log$C(r)$ versus log$r$ plot is identified as the scaling region whose slope is taken to be $D_2$. Conventionally one takes $p_i(r) = m_i/N_p$ as the weight of the $i$th box, where $N_p$ is the total number of points in the attractor and $m_i$ is the number of points falling in the $i$th box.

Eventhough the procedure appears to be simple computationally, two main drawbacks have been pointed out regarding the box counting algorithm in practice. Firstly, a chaotic attractor is, in general, highly nonuniform. Hence many boxes are needed to provide a good covering, especially for the parts of the attractor that are rarely visited. As a result, the convergence of $D_2$ with respect to the number of data points is very slow. Secondly, the box size $r$ needed to resolve the fractal structure of the attractor is usually much smaller than the practical length scale corresponding to the scaling region, leading to incorrect values of $D_2$. This is especially true in the case of hyperchaotic and high dimensional systems, as we show in §4.

Several improved box counting algorithms have been reported in the literature to overcome these difficulties. For example, Block et al. [27] has proposed an efficient numerical scheme that can reduce the computer time and memory. Similarly, Yamaguti [28] presented an improved box counting strategy which tried to minimise the convergence problem of $D_2$ with respect to the number of data points. But even after many improvements in the computational procedures and counting strategies, the problem regarding the scaling region still remain. For example, in the limit of the box size $r \to 0$, one expects, with finite number of points that, $m_i \to 1$ (for all $i$) and consequently $N(r) \to N_p$. Then, the normalised sum $N_p^2 C(r) \equiv \sum_{i=1}^{N_p} m_i^2 \to N_p$. This, in turn, forces the scaling region to be relatively on the higher side of $r$.

The shortcomings of the conventional scheme can be very clearly seen from the first two figures. In both these figures, the size of the embedded attractor has been normalised to unity so that $r$ varies from 0 to 1. Fig. 1 shows the log–log plot of (normalised) $C(r)$ versus $r$ for the standard Henon attractor using the conventional box counting scheme for different values of $M$ for number of data points $10^4$ and $10^5$. It is evident that $C(r)$ saturates to the number of data points ($N_p$) as $r \to 0$. The saturation effect is explicitly seen in our calculations because of the uniform deviate transformation (see below) of the time series by which, one is able to use much smaller values of box size $r$ to cover the attractor. The vertical dashed line in the figure roughly indicates the beginning of the scaling region in both cases. The scaling region gets extended to lower $r$ values as $N_p$ increases. Fig. 2 shows the same results for the standard Lorenz attractor. It is evident that in both cases, the scaling region corresponds to relatively higher values of $r$, resulting in a slow convergence of the computed $D_2$ values (see, for example, Fig. 5 below). We now show that a few changes in the computational procedure and a revised expression for computing the probability can greatly improve the results.

![Fig. 1. Log–log plot of the normalised sum $C(r)$ (see text) versus the box size $r$ used to cover the Henon attractor using the conventional box counting scheme for different values of the embedding dimension $M$, with the number of data points $10^4$ and $10^5$. Note that as $r$ decreases, $C(r)$ saturates to the number of data points used in the time series in both cases. Also, the scaling region extends to lower $r$ values with the increase in the number of data points, as indicated by the vertical dashed lines in the figure.](image-url)
As pointed above, one major drawback of the box counting algorithm is that with a finite time series, many of the boxes remain unoccupied. This is because, the series fills the attractor in a highly nonuniform manner and many parts of the attractor are rarely visited. This gives a slow convergence of the number of occupied boxes, making $D_2$ inaccurate.

We find that this problem can be minimised by transforming the time series into a uniform deviate in the unit interval $[0,1]$. By this, the number of vacant boxes becomes very small. Hence the first important step in our modified scheme is to transform the time series into a uniform deviate. To illustrate the effect of this transformation, we show in Fig. 3 the time delayed embedding of the Lorenz attractor with and without uniform deviate transformation of the time series. We will show in the next section that, by using the modified scheme, one gets a much better convergence of $D_2$ with respect to the number of data points.

The second major change in our approach compared to the conventional one is in the equation for computing the probability $p_i$ that the trajectory passes through the $i$th box. The emphasis on the box counting papers so far has been on minimising the computer time and memory [25,27] and on improved box counting strategies [28,29]. But as we have shown in Figs. 1 and 2, $C(r)$ saturates to the number of data points $N_p$ if we take $p_i$ in the conventional manner as the weight of the $i$th box. We show that the problem of saturation can be rectified by redefining $p_i$ in terms of the ensemble average of the number of points falling in the $i$th box, as discussed below in detail.

Consider an ensemble of chaotic attractors embedded in $M$ dimension, with each element in the ensemble consisting of $N_p$ number of points. Assume that each element of the ensemble is covered using $N(r)$ number of $M$ dimensional cubes of side length $r$. Let $p_i$ be the probability that the trajectory passes through the $i$th box for any randomly selected element of the ensemble. Then the probability that any $m_i$ out of $N_p$ points fall in the $i$th box is given by the Binomial distribution:

$$B_{m_i} = \frac{N_p!}{m_i!(N_p-m_i)!}p_i^{m_i}q_i^{N_p-m_i}$$

where $q_i = (1 - p_i)$. We have

$$\sum_{m_i=0}^{N_p} B_{m_i} = (p_i + q_i)^{N_p} = 1$$

Fig. 2. Same as the previous figure, but for the Lorenz attractor. Again, $C(r)$ saturates to the number of data points in the time series as $r$ decreases.
The ensemble average of the number of points in the $i^{th}$ box (or, the expectation value of getting $m_i$ points in the $i^{th}$ box) is given by

$$ \langle m_i \rangle = \sum_{m_i=0}^{N_p} m_i B_{m_i} $$

Substituting for $B_{m_i}$ from above

$$ \langle m_i \rangle = \sum_{m_i=0}^{N_p} m_i N_p^m q_i^{N_p-m_i} m_i! (N_p-m_i)! $$

which simplifies to

$$ \langle m_i \rangle = \sum_{m_i=1}^{N_p} \frac{N_p p_i^m q_i^{N_p-m_i}}{(m_i-1)! (N_p-m_i)!} $$

Making a change of variable from $m_i$ to $m'_i \equiv (m_i - 1)$, the above equation reduces to

$$ \langle m_i \rangle = N_p p_i \sum_{m'_i=0}^{N_p-1} \frac{(N_p-1) p_i^{m'_i} q_i^{N_p-1-m'_i}}{m'_i! (N_p-1-m'_i)!} $$

The summation in the RHS is simply the Binomial function for $(N_p - 1)$ trials which add up to 1. This gives

$$ p_i = \frac{\langle m_i \rangle}{N_p} $$

Thus, considering an ensemble of attractors, the probability that the trajectory passes through the $i^{th}$ box is determined by the ensemble average $\langle m_i \rangle$ of the number of points $m_i$ in the $i^{th}$ box.

Now, the equation for $D_2$ involves $p_i^2$. To calculate $p_i^2$, we consider

$$ \langle m_i^2 \rangle = \sum_{m_i=0}^{N_p} m_i^2 B_{m_i} $$

Fig. 3. The top panel shows the time delayed embedding of the Lorenz attractor using a time series consisting of 10000 data points. The bottom panel shows the Lorenz attractor embedded from the same time series after uniform deviate transformation.
To simplify, we first rewrite \( m^2 \) as \( m(m-1) + m \), which gives

\[
\langle m^2 \rangle = \sum_{m=0}^{N_p} m_i(m_i - 1)B_{m_i} + \langle m_i \rangle
\]

which reduces to

\[
\langle m^2 \rangle = \frac{\sum_{m=0}^{N_p} m_i(m_i - 1)N_p p_i^m q_i^{N_p - m_i}}{m!(N_p - m_i)!} + N_p p_i
\]

The first term in RHS (say, RHS (1)) can be simplified as above by making a change of variable from \( m_i \) to \( (m_i - 2) \). We have

\[
RHS(1) = \sum_{m_i=0}^{N_p-2} \frac{N_p p_i^m q_i^{N_p - m_i}}{(m_i - 2)!(N_p - m_i)!}
\]

Taking \( m_i = (m_i - 2) \), we can write

\[
RHS(1) = \sum_{m_i=0}^{N_p-2} \frac{N_p p_i^m q_i^{N_p - m_i}}{m_i!(N_p - m_i - 2)!}
\]

Simplifying, one gets

\[
RHS(1) = N_p(N_p - 1)p_i^2 \sum_{m_i=0}^{N_p-2} \frac{(N_p - 2)!p_i^m q_i^{N_p - 2 - m_i}}{m_i!(N_p - 2 - m_i)!}
\]

which reduces to

\[
RHS(1) = N_p(N_p - 1)p_i^2
\]

since the summation in RHS is Binomial function for \( (N_p - 2) \) trials which add up to 1. Thus we get

\[
\langle m^2 \rangle = N_p(N_p - 1)p_i^2 + N_p p_i
\]

In the limit \( N_p \gg 1 \),

\[
p_i^2 = \frac{\langle m^2 \rangle - N_p p_i}{N_p^2}
\]

Thus, it is clear that \( p_i^2 \neq \left( \frac{m_i}{N_p} \right)^2 \) as is taken conventionally. As a first approximation, if one takes \( \langle m_i \rangle = m_i \), the actual number of points in the ith box and \( \langle m^2 \rangle = m_i^2 \), one gets

\[
p_i^2 \approx \frac{m_i^2 - p_i N_p}{N_p}
\]

Then \( C(r) \) is given by

\[
\sum_{i=1}^{N(r)} p_i^2 = \frac{1}{N_p} \left[ \sum_{i} m_i^2 - N_p \right]
\]

Thus, in our modified scheme, the above equation is used to compute \( C(r) \). Note that in the limit that all elements of the ensemble of attractors are exactly identical (for example, generated from exactly the same initial conditions), \( m_i \rightarrow m \) and \( m_i^2 \rightarrow m^2 \) so that our modified expression reduces to the conventional one. In practice, one counts the number of points \( m_i \) in each box and calculate \( \sum m_i^2 \). Then, to get \( C(r) \), one has to subtract the total number of points in the attractor \( N_p \) from \( \sum m_i^2 \). We now show that this subtraction of \( N_p \) has a crucial effect, especially in the case of hyperchaotic attractors. But, before going into the hyperchaotic time series, we test our modified scheme using standard chaotic systems as well as white and colored noise with known behavior for \( D_2 \).

3. Testing the modified scheme

In order to test our modified scheme, we first recompute \( C(r) \) using Eq. (20) for the Henon and Lorenz attractors. The results are shown in Fig. 4 with \( N_p = 10^3 \) in both cases. Compare these results with that obtained in Figs. 1 and 2. It then becomes clear that, using the modified scheme, we are able to stretch the scaling region to much smaller length scales compared to the conventional scheme. This is important since one of the drawbacks of the box counting algorithm usually pointed out is that the length scale needed to resolve the fractal structure of the embedded attractor is generally much

...
smaller than the box size that can be employed to cover the attractor in a reasonable manner to get $D_2$. By stretching the scaling region, we are able to reduce this problem to some extent.

This is also reflected in the results obtained with our scheme in that, we get much better convergence of $D_2$ with respect to the number of data points as compared to the conventional scheme. For example, the results of computation of $D_2$ for the Lorenz attractor using the conventional as well as modified schemes for $N_p = 10^4$ and $10^5$ are shown in Fig. 5. Note that the convergence of $D_2$ is much better for the modified scheme. Moreover, with $10^5$ data points, we get the saturated value of $D_2$ as $2.01 \pm 0.04$, which is very close to the standard value.

In order to ensure the wide applicability of the scheme, we show in Fig. 6, the results of computing $D_2$ for four other standard low dimensional chaotic attractors. Reasonably accurate values of $D_2$ are obtained in all cases. We have also found that the scheme works well for high dimensional systems as well. As an example, we show in Fig. 7, the results of applying the scheme to pure white noise with $D_2 \to \infty$ and two colored noise data with high values of $D_2$. For colored noise, $D_2$ is given by the relation $D_2 = 2/(\alpha - 1)$, where $\alpha$ is the spectral index. We have shown the results for $\alpha = 1.5$ and $\alpha = 1.3$, with $D_2 \approx 4.0$ and $\approx 6.66$ respectively. In order to test the applicability of the scheme to real world data, we apply it to time series added with different amounts of noise. For this, we generate two data sets by adding 20% and 50% of white noise to Lorenz data. The results of applying our scheme to these two data sets are shown in Fig. 8. It is clear that the scheme is valid in general and gives better convergence of $D_2$ with respect to the number of data points.

Finally, the scheme can be automated by fixing the scaling region algorithmically. For this, we choose the scaling region by applying the conditions on $r_{\text{max}}$ and $r_{\text{min}}$ in the code, so that the code can directly compute $D_2$ as function of $M$. While $r_{\text{max}}$ is fixed directly by the condition $r_{\text{max}} = 0.1$, $r_{\text{min}}$ is fixed by applying a cut off on $C(r)$ in terms of the number of data points as $C(r) > 100/N_p$. The latter condition comes because, the scaling region can be extended to lower $r$ values as $N_p$ increases. We find that these conditions give optimum results for $D_2$ and good convergence with respect to $N_p$, for all the standard chaotic systems and noise data sets we have tested. We now turn to the analysis of hyperchaotic attractors.

4. Analysis of hyperchaotic time series

In this section, we apply the modified box counting scheme to compute $D_2$ from hyperchaotic time series. We show that the present scheme gives consistent results and is better suited for the correlation dimension analysis of hyperchaotic systems. These systems typically require much larger number of data points and higher embedding dimension to obtain satu-
ration of $D_2$. One should note that, in principle, $D_2$ for a hyperchaotic attractor is considered to be arbitrarily high so that, it is more difficult to get a proper saturation of $D_2$ as a function of $M$. Nevertheless, the trend in the variation of $D_2$ at higher $M$ values is taken here as an approximation to $D_2$. In this section, we analyse three hyperchaotic systems using our modified box counting algorithm.

The first one is a hyperchaotic flow represented by the following set of equations:

$$
\begin{align*}
\frac{dX}{dt} &= a(Y - X) + eYZ \\
\frac{dY}{dt} &= cX - dXZ + Y + W \\
\frac{dZ}{dt} &= XY - bZ \\
\frac{dW}{dt} &= -kY 
\end{align*}
$$

(21)

The system has been studied in detail by Chen et al. [8], and the authors have presented several interesting properties for the system including the presence of hyperchaotic attractors for certain ranges of parameter values. We choose the following set of parameters in that range and generate the hyperchaotic time series: $a = 35, b = 4.9, c = 25, d = 5, e = 35$ and $k = 100$. The result of applying our scheme to this time series with $10^5$ data points is shown in Fig. 9 (bottom panel). In the top panel, we show the results of applying the conventional scheme, for comparison. It is clear that the scaling region is fundamentally different in the two cases. Moreover, the variation of $C(r)$ is also basically different for normal chaotic and hyperchaotic systems, when our modified scheme is applied. While there is only a single linear part for normal chaotic systems, the attractor in the hyperchaotic phase clearly shows two scaling regions, indicated I and II in the figure, with a small intermediate region representing a cross over between the two slopes. Note that region I continues from the chaotic phase of the attractor, while region II appears only in the hyperchaotic phase at much lower $r$ values. In other words, the attractor shows multiple scaling in the hyperchaotic phase. It is also evident that only our modified scheme is able to reveal this new scaling region for the hyperchaotic attractor.

The presence of this multiple scaling region may be an indication that $D_2$ for a hyperchaotic attractor is scale dependent. In other words, $D_2$ depends on the choice of the scaling region as shown in Fig. 10. But theoretically, the $D_2$ that truly rep-
Fig. 6. $D_2$ versus $M$ for four standard low dimensional chaotic systems, computed using the modified box counting scheme. The number of data points used is 30000 in all cases.

Fig. 7. $D_2$ versus $M$ for white noise and two colored noise data with high value for dimension, computed using the modified scheme. The computations are done with $10^5$ data points in all cases.
Fig. 8. $D_2$ versus $M$ for time series from Lorenz attractor added with 20% and 50% of white noise. The computations are done with 30000 data points in both cases.

Fig. 9. The top panel shows the log–log plot of $C(r)$ versus $r$ for the hyperchaotic attractor generated from Eq. (21) (see text), by using the conventional box counting scheme. The bottom panel shows the same, but with the modified scheme. Number of data points in both cases is $10^5$. Two separate scaling regions (indicated I and II) are evident in the bottom panel for $C(r)$, with a small intermediate region in between where the slope changes. Also, the slope of the region I is same as that computed from the conventional scheme in the top panel.
Fig. 10. $D_2$ versus $M$ for the hyperchaotic attractor computed from the two scaling regions shown in the bottom panel of the previous figure.

Fig. 11. Log-log plot of the $C(r)$ versus $r$ for the time delayed M-G system (see text) for two different values of the time delay $\tau$, corresponding to the chaotic phase (top panel) and hyperchaotic phase (bottom panel). Note that the variation of $C(r)$ is different in the two cases. We also find that, as $\tau$ is increased, the number of data points should also be increased correspondingly for effectively computing $D_2$ at higher embedding dimension.
resents the geometric complexity of an attractor should be the one corresponding to the scaling region in the limit $r \to 0$. This implies that the slope of the newly created scaling region II is to be chosen as the proper $D_2$ of a hyperchaotic attractor. It also appears reasonable, as the $D_2$ corresponding to region II tends to saturate at a much higher value compared to that of chaotic attractors.

To substantiate the above results, we now consider two more hyperchaotic systems. These two systems are basically different from the above hyperchaotic flow as they are time delayed systems. Such systems make a transition from chaotic to hyperchaotic phase as the time delay $\tau$ increases beyond a critical value. The first one is the standard Mackey–Glass (M-G) system [11] given by the equation:

$$\frac{dx}{dt} = \beta x_t - \frac{x^n}{1 + x^2} - \gamma x$$

(22)

where $\beta, \gamma, n$ and $\tau$ are real numbers and $x_t$ represents the value of the variable $x$ at $(t - \tau)$. Depending on the values of the parameters, the equation displays a range of periodic, chaotic and hyperchaotic dynamics. The M-G system is a typical example of a feedback system where, the value of a control variable is sensed and appropriate changes are made to achieve a stable output. The system is important from a biological perspective, as physiological control systems represent prototypical feedback systems.

Here we choose $\beta = 2$, $\gamma = 1$, $n = 10$ and generate time series for various values of $\tau$ in the chaotic as well as hyperchaotic phase. In Fig. 11, we show the variation of $C(r)$ with $r$ for the time series from M-G system for $\tau = 1.8$ and $3.0$. We find that as $\tau$ increases, the number of data points are also to be increased correspondingly to get saturation of $D_2$ at higher embedding dimension. We use $N_p = 10^5$ for $\tau = 1.8$ and $N_p = 4 \times 10^5$ for $\tau = 3.0$. Note that the variation of $C(r)$ for the latter clearly indicates hyperchaotic behavior, while for $\tau = 1.8$, the system is in the chaotic phase. Once again, as the system enters the hyperchaotic phase, it develops two separate scaling regions. The variation of $D_2$ with $M$ for three values of $\tau$ is shown in Fig. 12, as computed by our modified scheme. The newly created scaling region corresponding to the lowest part of $C(r)$ versus $r$ is chosen for computing $D_2$. As expected, $D_2$ increases with $\tau$.

Our second example of the time delayed system is the Ikeda system [12] given by

$$\frac{dx}{dt} = -\alpha x + m \sin x(t - \tau)$$

(23)

where $a$ and $m$ are parameters and $\tau$ is the time delay. The Ikeda model was introduced to describe the dynamics of an optical bistable resonator and is well known for delay induced chaotic and hyperchaotic behavior [30,31]. We choose $a = 5$ and

![Fig. 12. Variation of $D_2$ with $M$ for the attractor generated from the M-G system for 3 different values of $\tau$. The lowest part of $C(r)$ versus $r$ (region II) is chosen as the scaling region in all cases for computing $D_2$. The number of data points used is $10^5$ for $\tau = 1.8$ and $4 \times 10^5$ for $\tau = 3.0$ and 5.0.](image-url)
m = 20 and generate time series for different values of $s$. We find that the variation of $C(r)$ with $r$ is identical to that of the M-G system for $s$ values in the hyperchaotic phase. In Fig. 13, we show $D_2$ as a function of $M$ for three different values of $s$ computed by our scheme. While the lowest one represents the chaotic phase, the other two $s$ values correspond to the hyperchaotic phase.

It is evident that our modified scheme gives consistent results for all hyperchaotic systems. The scheme also brings out some interesting features of hyperchaotic systems in comparison with the chaotic ones. For example, as mentioned earlier, the proper scaling region for a hyperchaotic attractor clearly shifts towards much lower $r$ values compared to a chaotic attractor. This implies that the box size or the length scale needed to resolve the fractal structure of a hyperchaotic attractor is much smaller than that for a chaotic attractor. Correspondingly, to automate the scheme for hyperchaotic systems, the condition on $r_{\text{max}}$ should be reduced suitably. Moreover, a newly created scaling region shifted towards lower $r$ values indicates a transition from chaos to hyperchaos and thus computing $D_2$ can give information regarding this transition, especially in time delayed systems, as $s$ is varied. More details regarding this will be presented elsewhere.

5. Conclusion

Hyperchaotic systems have gained importance in recent years due to a number of practical applications, such as, secure communication, network analysis and synchronisation studies. Geometric complexity of the hyperchaotic attractor is crucial in most of these applications. Important quantitative measures used for the analysis of hyperchaotic time series are LE and $D_2$. In this paper, we propose some modifications in the conventional box counting scheme to make it suitable for the correlation dimension analysis of hyperchaotic time series. Essentially, we consider the covering of an ensemble of attractors and redefine the probability in terms of the ensemble average of the number of points falling in the $i$th box.

One important advantage of the scheme is that it can be automated by algorithmically fixing the scaling region so that $D_2$ can be computed as a function of $M$ directly from the time series. Further, the scheme can be applied to all types of time series data and gives better convergence for $D_2$ in all cases with respect to number of data points. Apart from this, the present scheme brings in only minor improvements over the existing box counting schemes for the analysis of standard low dimensional chaotic systems, such as, the extension of the scaling region to lower $r$ values and better convergence of $D_2$.

The real advantage of the scheme is in the analysis of hyperchaotic time series. As we show explicitly, the existing box counting schemes are unable to provide accurate results in such cases. We show the utility of our scheme by applying it to one hyperchaotic flow and two standard time delayed hyperchaotic systems. In all cases, we are able to obtain reasonable estimates of $D_2$.  

Fig. 13. Same as the previous figure, but for the Ikeda system (see text) for 3 different values of $s$ as shown. For $s = 0.3$, the number of data points used is $10^5$, while for $s = 1.0$ and 2.0, it is $4 \times 10^5$. The former represents the chaotic phase, while the latter two values of $s$ correspond to hyperchaotic phase.
Finally, our analysis also brings out some interesting features of a hyperchaotic attractor. In particular, as the system makes a transition from chaotic to hyperchaotic phase, a new scaling region is created at much lower values of the box size in addition to the already existing scaling region in the chaotic phase. In other words, a hyperchaotic attractor appears to be geometrically more complex with multiple scaling, as compared to a normal chaotic attractor. Our scheme is able to clearly distinguish between the two phases through the computation of $D_2$ and hence is also able to give useful information regarding the chaos-hyperchaos transition as well, which is important in many ways. For example, hyperchaos-chaos transition occurs at the onset of synchronisation in coupled systems [16]. Such studies have so far been done using only LE and recurrence plots [13]. We thus hope that the results presented here may lead to a better understanding of the hyperchaotic attractors in general.

Acknowledgements

KPH and RM acknowledge the financial support from Dept. of Sci. and Tech., Govt. of India, through a Research Grant No. SR/S2/HEP-11/2008. KPH acknowledges the hospitality and the computing facilities in IUCAA, Pune.

References